

Alibaba Cloud

対象存储
ユーティリティ

Document Version: 20200819

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).
5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
 Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
 Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.
 Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: If the weight is set to 0, the server no longer receives new requests.
 Note	A note indicates supplemental instructions, best practices, tips, and other content.	 Note: You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click Settings> Network> Set network type .
Bold	Bold formatting is used for buttons, menus, page names, and other UI elements.	Click OK .
Courier font	Courier font is used for commands	Run the <code>cd /d C:/window</code> command to enter the Windows system folder.
<i>Italic</i>	Italic formatting is used for parameters and variables.	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] or [a b]	This format is used for an optional value, where only one item can be selected.	<code>ipconfig [-all -t]</code>
{ } or {a b}	This format is used for a required value, where only one item can be selected.	<code>switch {active stand}</code>

Table of Contents

1.OSS ツールリスト	07
2.ossbrowser	10
2.1. クイックスタート	10
2.2. アクセス許可の管理	14
3.ossutil	19
3.1. 概要	19
3.2. ダウンロードとインストール	20
3.3. 一般的なコマンド	22
3.3.1. appendfromfile	23
3.3.2. bucket-encryption	24
3.3.3. bucket-policy	26
3.3.4. bucket-tagging	30
3.3.5. cat	32
3.3.6. config	33
3.3.7. cors	37
3.3.8. cors-options	38
3.3.9. cp	40
3.3.10. cp	40
3.3.11. create-symlink	52
3.3.12. du	54
3.3.13. getallpartsize	55
3.3.14. hash	56
3.3.15. help	58
3.3.16. lifecycle	58
3.3.17. listpart	61
3.3.18. logging	62

3.3.19. ls	63
3.3.20. mb	69
3.3.21. mkdir	71
3.3.22. object-tagging	72
3.3.23. probe	74
3.3.24. read-symlink	77
3.3.25. set-meta	78
3.3.26. stat	82
3.4. サポートされているすべてのオプションの表示	83
3.5. バケット関連のコマンド	85
3.6. オブジェクト関連のコマンド	98
3.7. マルチパート関連コマンド	102
4.ossimport	105
4.1. アーキテクチャと構成	105
4.2. スタンドアロン展開	120
4.3. 分散配置	124
4.4. データ移行	128
4.5. よくある質問	132
5.RAM Policy Editor	140
6.ossftp	142
6.1. OSS FTP のクイックインストール	142
6.2. Discuz で OSS インスタンスにリモート添付ファイルを格納する...	145
6.3. PHPWind でリモート添付ファイルを OSS インスタンスに格納す...	145
6.4. WordPress でリモート添付ファイルを OSS インスタンスに格納...	146
6.5. ファイル共有のために RAM を統合する方法	147
6.6. よくある質問	152
7.ossfs	153
7.1. クイックインストール	153

7.2. よくある質問	155
8. osscmd	162
8.1. クイックインストール	162
8.2. 例	163
8.3. バケットコマンド	168
8.4. オブジェクトコマンド	172
8.5. マルチパートコマンド	177

1.OSS ツールリスト

OSSには、コンソールの他にも OSS をより効率的に使用するためのツールが用意されています。

ツール	説明
ossbrowser	OSS 用のグラフィカルオブジェクト管理ツールです。 • グラフィカルインターフェイスを備えた、使いやすいツールです。 • Windows エクスプローラーと同様の機能を提供します。 • オブジェクトを直接確認できます。 • フォルダ (ディレクトリ) のアップロードとダウンロードをサポートします。 • 同時アップロードと再開可能アップロードをサポートします。 • グラフィカルインターフェイス上で、ポリシーに基づいた RAM ユーザーの承認をサポートします。 • Windows、Linux、および Mac をサポートします。 制限事項: • ossbrowser はグラフィカルツールであるため、伝送速度と性能の面では ossutil が優れています。 • アップロードまたはコピー可能なオブジェクトのサイズは5 GB 未満に限られます。
ossutil	オブジェクトとバケットの管理に使用するコマンドラインツールです。 • オブジェクトとバケットの管理に使用する、操作性に優れ、便利でシンプルなコマンドが豊富に用意されています。 • 同時アップロードと再開可能アップロードをサポートします。 • フォルダ (ディレクトリ) のアップロードとダウンロードをサポートします。 • 日常的に使用するバケット管理コマンドをサポートします。
osscmd	オブジェクトとバケットの管理に使用するコマンドラインツールです。 • オブジェクトとバケットを管理するために必要なすべてのコマンドが用意されています。 • Windows と Linux をサポートします。 制限事項: • Python 2.5 から 2.7 に対応し、Python 3.x には非対応です。 • 低コストアクセスソリューション (IA) およびアーカイブストレージクラス、リージョン間レプリケーション、バックツーオリジンのミラーリングなど、OSS の新機能はサポート対象外です。 ❓ 説明 osscmd は、将来 ossutil に置き換えられる予定です。ossutil でサポートされていないバケット管理機能が必要な場合を除き、ossutil の使用を推奨します。

ツール	説明
ossfs	ローカルファイルシステムにバケットをマウントするために使用するツールです。 ossfs を使用すると、OSS バケットを Linux のローカルファイルシステムにマウントできます。バケットのマウント後、ローカルファイルシステムで操作を実行し、バケット内のオブジェクトにアクセスまたはオブジェクトを共有できます。 • ファイルの読み書き、ディレクトリ、リンク操作、権限、UID / GID、拡張属性など、POSIX ファイルシステムのほとんどの機能をサポートします。 • マルチパートアップロード機能を使用して、大きなサイズのファイルをアップロードできます。 • データの整合性を確保するため、MD5 検証をサポートしています。 制限事項: • アーカイブストレージクラスのバケットはマウントできません。 • アップロードされたファイルを編集すると、ファイルが再度アップロードされます。 • <code>list directory</code> などのメタデータ関連の操作については、OSS サーバーにリモートでアクセスする必要があるため、速度が低下します。 • オブジェクト名またはフォルダ名を変更した際にエラーが発生する場合があります。操作が失敗した場合、データの整合性が失われる場合があります。 • 読み取りと書き込み操作が頻繁に同時実行されるケースには適していません。 • OSS バケットが複数のクライアントにマウントされている場合は、データの整合性を確保する必要があります。たとえば、1つのオブジェクトが複数のクライアントによって同時に書き込まれるのを防止するため、オブジェクトの使用をスケジューリングする必要があります。 • ハードリンクはサポートされません。
ossftp	OSS オブジェクトの管理に使用するツールです。 • ossftp を使用すると、FileZilla、WinSCP、FlashFXP などの FTP クライアントを使用して OSS の操作を実行できます。 • Ossftp は、FTP リクエストを受け取り、ファイルおよびフォルダの操作を OSS オブジェクトおよびバケットの操作にマップする FTP サーバーとして機能します。 • Python2.7 以降に対応します。 • Windows、Linux、および Macをサポートします。
ossimport	OSS とデータを同期するために使用するツールです。 • ossimport を使用すると、サードパーティのデータソースファイルを OSS に同期できます。 • 分散デプロイモードをサポートします。複数のサーバーを使用して、データを一括で移行できます。 • 大量データの移行をサポートします。 • Windows と Linux をサポートします。 • Java 1.7 以降に対応しています。

ツール	説明
ビジュアル署名ツール	サードパーティ製のビジュアル署名ツールを使用できます。 • OSS データの署名付き URL を生成するために使用するツールです。 • OSS データの署名時に発生したエラーをデバッグできます。OSS データの署名時にエラーが発生した場合、ツールで生成した署名と照合してエラーの原因を特定できます。 • ブラウザー版では、Chrome、Firefox、Safari をサポートします。
RAM Policy Editor	承認ポリシーを自動的に生成するためのツールです。 • 要件に基づいて、承認ポリシーを自動的に生成します。生成されたポリシーは、RAM コンソールのカスタムポリシーで追加します。 • Chrome、Firefox、および Safari をサポートします。 🔍 説明 承認ポリシーを自動的に生成する必要がある場合は、このツールの使用を推奨します。
oss-emulator	軽量の OSS サービスシミュレータです。 • OSS と同じ API を提供し、OSS アプリケーションのデバッグとテストに適用されます。 • Ruby 2.2.8 以降に対応しています。 • Windows と Linux をサポートします。

2.ossbrowser

2.1. クイックスタート

ossbrowser は、Alibaba Cloud が開発した、グラフィカル管理ツールです。機能は Windows エクスプローラーと似ています。ossbrowser を使用すると、オブジェクトを簡単に表示、アップロード、ダウンロード、および管理できます。

? 説明

- 移動またはコピーできるオブジェクトのサイズは、5 GB 未満です。5 GB を超えるオブジェクトについては、[ossutil](#) を使用してください。
- ossbrowser は Linux、Mac、および Windows (Windows 7 以降) をサポートしています。Windows XP および Windows Server での使用は推奨されていません。

インストール

1. ossbrowser をダウンロードして、インストールします。

オペレーティングシステム	ダウンロードリンク
Windows x32	Windows x32
Windows x64	Windows x64
MAC	MAC
Linux x64	Linux x64

? 説明 その他のダウンロード URL については、[Github](#) をご参照ください。

2. ossbrowser を起動します。□

以下のパラメーターを設定し、ossbrowser にログインします。

- **Endpoint:** ログインするリージョン (エンドポイント) を選択します。
 - **Default:** デフォルトのエンドポイントで ossbrowser にログインします。
 - **Customize:** ossbrowser へのログインに使用するエンドポイントを入力します。"http" または "https" で始まる URL を入力して、HTTP または HTTPS メソッドを使用して ossbrowser にログインします (例: <https://oss-cn-beijing.aliyuncs.com>)。エンドポイントの詳細については、「[リージョンおよびエンドポイント](#)」をご参照ください。
 - **cname:** OSS リソースにバインドされたカスタムドメイン名 (CNAME) で OSS ブラウザにログインできます。CNAME のバインドについての詳細は、「[カスタムドメイン名のバインド](#)」をご参照ください。
- **AccessKeyId/AccessKeySecret:** アカウントのアクセスキー (AK) を入力します。データのセキュリティを確保するため、RAM ユーザーのアクセスキーを使用して ossbrowser にログインすることを推奨します。アクセスキーの詳細については、「[AccessKey の作成](#)」をご参照ください。
- **Preset OSS Path:**

- Administrator RAM users with administration permissions on all buckets: 設定は不要です。
- Operator RAM users: 設定が必要です。アクセスする OSS バケット、またはサブディレクトリのパスを入力します (RAM ユーザーは OSS バケットまたはサブディレクトリへのアクセス権限が必要です)。パスの形式は、oss:// bucket name/sub-directory name/ です。
- Region: OSS のリソースが属するリージョンを選択します。
- Remember: アクセスキーを保存する場合に選択します。ossbrowser へのログイン時に、アクセスキーを繰り返し入力せずに、[AK Histories] をクリックして保存済みの アクセスキーを選択するだけでログインできます。共有コンピューターを使用している場合は、このオプションを選択しないでください。

使用方法

OSSBrowser で、OSS リソースを容易に管理できます。

- バケットの管理
 - バケットの作成
 - a. ossbrowser のメイン画面で、[Create Bucket]をクリックします。
 - b. バケットに関する次の情報を設定します。
 - Name: バケット名は長さ 63 文字まで、かつ一意である必要があります。
 - Region: バケットが属するリージョンを選択します。
 - ACL: バケットの ACL を選択します。ACL についての詳細については、「[ACL](#)」をご参照ください。
 - Type: バケットのデフォルトストレージクラスを選択します。ストレージクラスの詳細については、「[ストレージクラスの概要](#)」をご参照ください。
 - c. [OK] をクリックします。
 - バケットの削除

削除するバケットを選択し、[More]>[Remove] をクリックします。オブジェクトまたはパーツがバケットに格納されている場合、バケットを削除することはできません。
- オブジェクト/ディレクトリの管理
 - ディレクトリの作成
 - a. ossbrowser のメイン画面で、フォルダを作成するバケットをクリックします。
 - b. [ディレクトリ] をクリックします。

c. ディレクトリ名を入力し、[OK] をクリックします。

② 説明

- ディレクトリ名に顔文字は使用できません。また、UTF-8 準拠の文字を使用してください。
- 一度に作成できるディレクトリは 1 階層だけです。たとえば、*abc* のように 1 階層のディレクトリは作成できます。しかし、複数階層のディレクトリ *abc/123* は作成できません。
- サブディレクトリ名を「..」とすることはできません。
- ディレクトリ名の長さは、1 から 254 文字です。

○ ファイル/ディレクトリのアップロード

指定したバケットまたはディレクトリで、[Files]/[Folder]をクリックし、アップロードするファイルまたはフォルダを選択します。同時に複数のファイルまたはフォルダをアップロードできます。

○ オブジェクト/ディレクトリのダウンロード

指定したバケットまたはディレクトリで、ダウンロードするオブジェクトまたはディレクトリを選択し、[Download] をクリックします。同時に複数のオブジェクトまたはフォルダをダウンロードできます。

○ オブジェクト/ディレクトリのコピー

- a. 指定したバケットまたはディレクトリで、[Copy] をクリックします。
- b. データをコピーするバケットまたはディレクトリを入力し、[Paste] をクリックします。コピーされたオブジェクトのコピー元のアドレスとコピー先のアドレスが同じである場合、元のオブジェクトは上書きされます。上書きされたオブジェクトのストレージクラスが IA またはアーカイブで、保存されていた期間が規定の日数に達していなかった場合は、早期削除の料金が発生します。詳しくは、「[OSSの課金について](#)」をご参照ください。

○ オブジェクト/ディレクトリの移動

- a. 指定したバケットまたはディレクトリで、移動するオブジェクトまたはディレクトリを選択し、[More] > [Move] をクリックします。
- b. データを移動するバケットまたはディレクトリを入力して、[Paste] をクリックします。

 **注意** オブジェクトまたはディレクトリを移動すると、オブジェクトまたはディレクトリが元のアドレスからコピー先のアドレスにコピーされ、元のアドレスにあったオブジェクトまたはディレクトリは削除されます。移動したオブジェクトのストレージクラスが IA またはアーカイブで、保存されていた期間が規定の日数に達していなかった場合は、早期削除の料金が発生します。

- オブジェクト/ディレクトリの名の変更

指定したバケットまたはディレクトリで名前を変更するオブジェクトまたはディレクトリを選択し、[More]>[Rename]をクリックして新しい名前を入力します。

 注意

- 名前を変更できるのは、1 GB 未満のオブジェクトに限られます。
- オブジェクトまたはディレクトリ名を変更すると、オブジェクトまたはディレクトリがコピーされ、名前が変更され、保存されます。元のオブジェクトまたはディレクトリは削除されます。名前を変更したオブジェクトのストレージクラスが IA またはアーカイブで、保存されていた期間が規定の日数に達していなかった場合は、早期削除の料金が発生します。

- オブジェクト/ディレクトリの削除

削除するオブジェクトまたはディレクトリを選択して、[More]>[Remove]をクリックします。削除したオブジェクトのストレージクラスが IA またはアーカイブで、保存されていた期間が規定の日数に達していなかった場合は、早期削除の料金が発生します。

- オブジェクトの URL 生成

- a. 指定したオブジェクトを選択して、[More]>[Address]をクリックします。
- b. URL の有効期間を入力し、Generate をクリックします。
- c. [Copy] または [Mail it] をクリックして、オブジェクトにアクセスするユーザーに URL を送信します。QR コードをスキャンしてオブジェクトにアクセスすることもできます。

- オブジェクトのプレビュー表示

オブジェクトをダブルクリックすると、プレビューを表示できます。ossbrowser では、txt および PDF 形式の画像やオブジェクトをプレビュー表示できます。

- パーツの管理

指定したバケットを選択して、[Multipart]をクリックします。不要なパーツを削除できます。

その他の操作

- アップロード/ダウンロードパフォーマンスの最適化

Settings をクリックして、次のパラメーターを設定できます。

- **Upload tasks concurrent number:** 同時に実行できるアップロードタスク数の上限を指定します。アップロードタスクの数が設定値を超えた場合、追加のタスクは現在のタスクが完了するまで保留されます。帯域幅に応じてパラメーターを適切に設定することで、アップロード速度が向上する場合があります。
- **Download tasks concurrent number:** 同時に実行できるダウンロードタスク数の上限を指定します。ダウンロードタスクの数が設定値を超えた場合、追加のタスクは現在のタスクが完了するまで保留されます。帯域幅に応じてパラメーターを適切に設定することで、ダウンロード速度が向上する場合があります。
- **overtime:** タスクのタイムアウト時間を指定します。
- **uploadpart size:** マルチパートアップロードタスクでのパーツのサイズを指定します。アップロードするファイルのサイズが非常に大きい場合や、ネットワークの状態が不安定な場合、オブジェクトを分割してアップロードするためのパーツサイズを設定できます。
- **retry times:** アップロードタスクまたはダウンロードタスクで再試行が可能な時間を指定します。

- 電子メールの設定

[Settings] をクリックして電子メール アカウントを設定します。ossbrowser の電子メールに関連するすべての操作は、アカウント単位で実行されます。

- ログ出力設定

- デバッグモードの有効化

アップロード、ダウンロード、およびその他の操作によって生成されるログを表示する場合、次の 2 つの方法でデバッグモードを有効化できます。

- 1: [Settings] をクリックし、続いて [Open debug] をクリックします。□
 - 2: 左上隅にある oss browser のロゴを 10 回連続してクリックします。□

- ログの有効化

ログ出力機能は、[Settings] ダイアログボックスで有効化できます。

- Local log を選択してローカルログ出力機能を有効化し、エラーログを収集します。ossbrowser によって収集されたログは、デフォルトで次のパスに格納されます。
 - Linux: ~/.config/oss-browser/log.log
 - Mac: ~/Library/Logs/oss-browser/log.log
 - Windows: %USERPROFILE%\AppData\Roaming\oss-browser\log.log
 - local info file log を選択して、通常のローカルファイル情報を収集します。

2.2. アクセス許可の管理

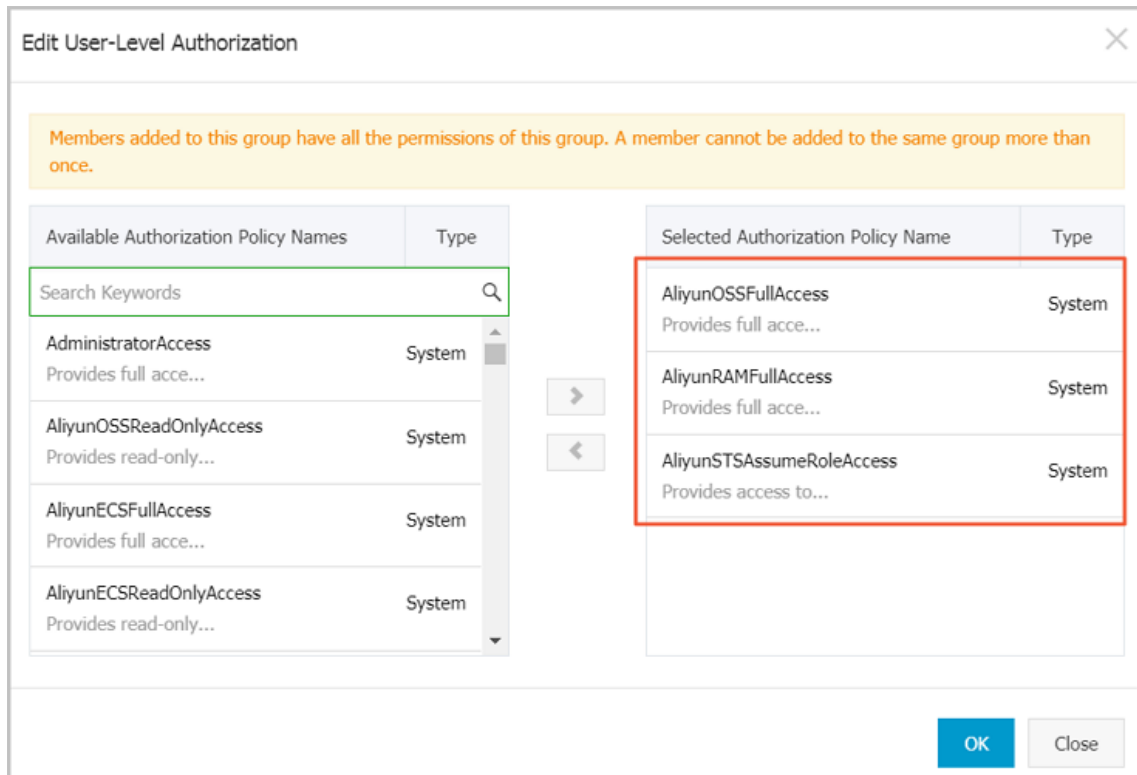
RAM ユーザーとして ossbrowser にログイン

データのセキュリティを確保するため、RAM ユーザーの AccessKey (AK) を使用して ossbrowser にログインすることを推奨します。ossbrowser にログインするには、次の手順に従います。

1. RAM ユーザーと AK を作成します。詳しくは、「[RAM ユーザーの作成](#)」をご参照ください。

RAM ユーザーは、権限に基づき、2 つのタイプに分類できます。

- 管理者 RAM ユーザー: 管理権限を持つ RAM ユーザーを示します。たとえば、すべてのバケットを管理し、他の RAM ユーザーを承認できる RAM ユーザーは管理者 RAM ユーザーです。次の図に示すように、Alibaba Cloud アカウントで RAM コンソールにログインして管理者 RAM ユーザーを作成し、管理者 RAM ユーザーにアクセス許可を付与することができます。



- オペレーター RAM ユーザー: バケットまたはディレクトリに対する読み取り専用のアクセス許可の
みを持つ RAM ユーザーを示します。管理者は、RAM ユーザーを承認するための簡易ポリシーを用
いることで、アクセス許可を付与できます。

② 説明 RAM ユーザーに低レベルのアクセス許可を付与できます。詳しくは、「アクセス制
御」をご参照ください。

2. 以下のパラメーターを設定し、ossbrowser にログインします。

- Endpoint: 既定値を使用します。
- AccessKeyId および AccessKeySecret: RAM ユーザーの AK を入力します。
- Preset OSS Path:
 - すべてのバケットに対する管理権限を持つ管理者 RAM ユーザー: 設定は不要です。
 - オペレーター RAM ユーザー: 設定が必要です。アクセスする OSS バケット、またはサブディレ
クトリのパスを入力します (RAM ユーザーは OSS バケットまたはサブディレクトリへのアクセ
ス許可が必要です)。パスの形式は次のとおりです。oss:// bucket name/sub-directory
name/
- 記憶する: AK を保存するため選択します。ossbrowser にログインしたときに、AK を繰り返し入
力する代わりに、[AK の履歴]をクリックして保存済みの AK を選択するだけで済むようになりま
す。共有コンピューターを使用している場合は、このオプションを選択しないでください。

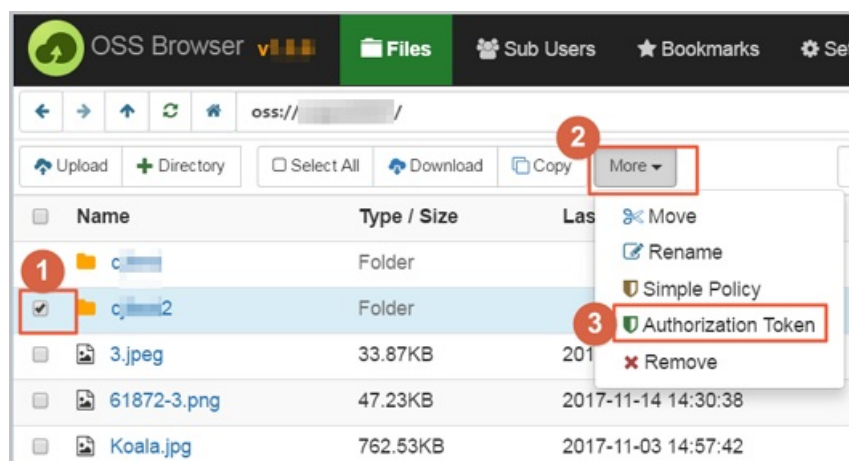
一時的な認証コードで、ossbrowser にログイン

一時的な認証コードを用いて OSSBrowser にログインすることができます。一時的な認証コードを承認ユーザーに提供することで、認証コードが期限切れになるまでの間、バケットのディレクトリーに一時的にアクセスできるようにすることができます。有効期限が切れると、一時的な認証コードは自動的に無効になります。

1. 一時的な認証コードを生成

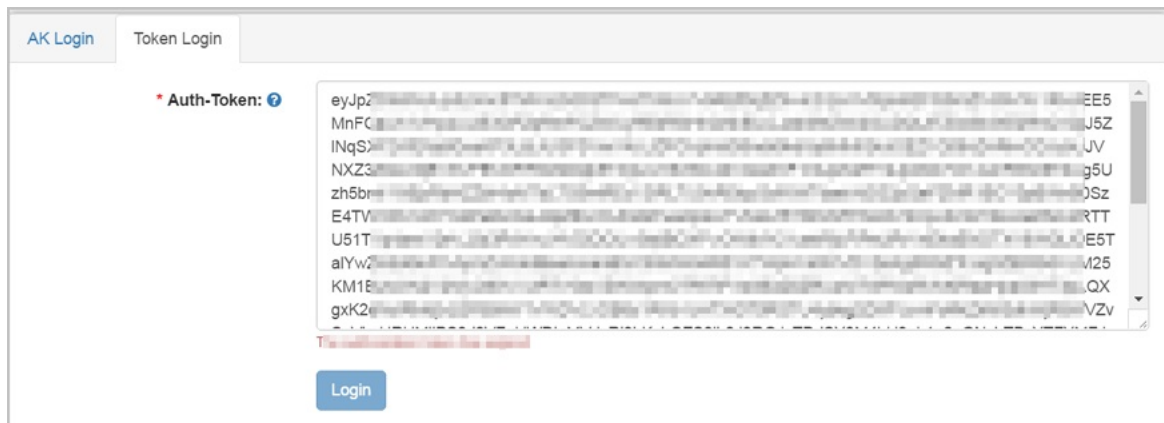
管理者 RAM ユーザーの AK を使用し、ossbrowser にログインします。以下の図に示すように、承認ユーザーは一時的にアクセスするオブジェクトまたはディレクトリを選択し、一時的な許可コード

を生成します。



2. 認証コードを使用して ossbrowser にログイン

次の図に示すように、一時的に承認されたユーザーは、認証コードを使用することで期限が切れるまでの間、ossbrowser にログインできます。

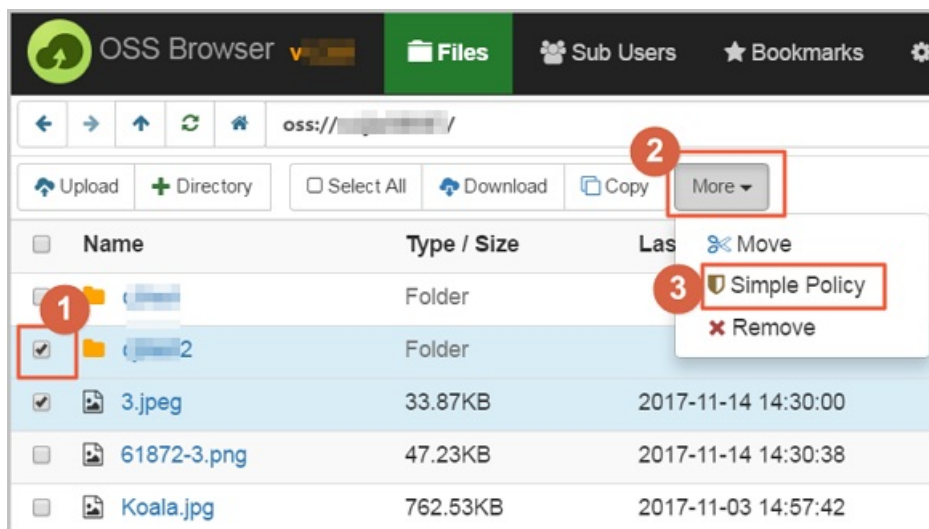


簡易ポリシーで権限を付与

管理者の RAM ユーザーとして ossbrowser にログインした後、簡易ポリシーでアクセス許可を付与してオペレーター RAM ユーザーを作成するか、オペレーター RAM ユーザーにバケットまたはディレクトリ上での読み取り専用または読み取り / 書き込み権限を付与することができます。

説明 Alibaba Cloud ossbrowser は、Alibaba Cloud RAM サービスに基づいたアクセスの制御機能である簡易ポリシー承認を提供しています。また、Alibaba Cloud の公式 Web サイトから、RAM コンソールにログインし、RAM ユーザーをより正確に管理することもできます。

1. 次の図に示すように、承認ユーザーが一時的にアクセスするオブジェクトまたはディレクトリを 1 つ以上選択し、[簡易ポリシー] をクリックします。



2. [ポリシー承認の簡易化] ダイアログボックスで、[特権] を選択します。
3. このダイアログボックスで、既存のオペレーター RAM ユーザーにアクセス許可を付与したり、新しいオペレーター RAM ユーザーを作成したりすることもできます。

 **注意** 簡易ポリシー承認を使用するには、RAM 構成権限を持つ RAM ユーザーの AK より、ossbrowser にログインする必要があります。たとえば RAM 構成権限を持つ管理者 RAM ユーザーの AK を使用します。



3.ossutil

3.1. 概要

ossutil は、OSS のデータを管理するために Windows、Linux、macOS で使用するコマンドラインツールです。このツールでは、オブジェクトとバケットを管理するための便利でシンプルなコマンドが幅広く提供されています。

ossutil を使用して、次の操作を実行できます。

- バケット管理 (バケットの作成、一覧表示、削除など)。
- オブジェクト管理 (オブジェクトのアップロード、ダウンロード、一覧表示、コピー、削除など)。
- パート管理 (パートの一覧表示、削除など)。

ossutil のインストール

ossutil をダウンロードし、インストールする方法の詳細は、「[ダウンロードとインストール](#)」をご参照ください。

一般的なコマンド

次の表に、ossutil でサポートされているコマンドを示します。

コマンド	説明
appendfromfile	ローカルファイルを OSS の追加可能オブジェクトにアップロードします。
bucket-encryption	バケットの暗号化設定を追加、変更、クエリ、または削除します。
bucket-policy	バケットのバケットポリシー設定を追加、変更、クエリ、または削除します。
bucket-tagging	バケットのタグ設定を追加、変更、クエリ、または削除します。
cat	オブジェクトのコンテンツを ossutil にエクスポートします。
config	OSS アクセス情報を保存する設定ファイルを生成します。
cors	バケットのクロスオリジンリソース共有 (CORS) 設定を追加、変更、クエリ、または削除します。
cors-options	バケットが、指定されたクロスオリジンリクエストを許可するかどうかをテストします。
cp	オブジェクトをアップロード、ダウンロード、またはコピーします。
create-symlink	シンボリックリンク (ソフトリンクとも呼ばれます) を作成します。
du	指定されたバケット、オブジェクト、またはフォルダーのストレージサイズを取得します。
getallpartsize	バケット内のすべてのパートの合計サイズと、アップロードされていない各パートのサイズを取得します。
hash	ローカルファイルの CRC64 値または MD5 値を計算します。

コマンド	説明
help	コマンドのヘルプ情報を取得します。特定のコマンドの情報を取得するには、 help コマンドを使用することを推奨します。
lifecycle	バケットのライフサイクル設定を追加、変更、クエリ、または削除します。
listpart	指定されたオブジェクトでアップロードされていないパートをリストします。
logging	バケットのログ設定を追加、変更、クエリ、または削除します。
ls	バケット、オブジェクト、またはパートをリストします。
mb	バケットを作成します。
mkdir	バケット内にディレクトリを作成します。
object-tagging	オブジェクトのタグ設定を追加、変更、クエリ、または削除します。
probe	OSS へのアクセスをモニタリングし、ネットワーク障害や不適切なパラメーター設定によってアップロードおよびダウンロードプロセス中に発生した問題をトラブルシューティングします。
read-symlink	シンボリックリンクオブジェクトの説明を読み取ります。
referer	バケットのホットリンク保護設定を追加、変更、クエリ、または削除します。
restore	単一のオブジェクトを凍結状態から読み取り可能状態に復元します。
request-payment	バケットの pay-by-requester 設定を設定またはクエリします。
rm	バケット、オブジェクト、またはパートを削除します。
set-acl	バケットまたはオブジェクトの ACL を設定します。
set-meta	アップロードされたオブジェクトのメタデータを設定します。
sign	第三者ユーザーがバケット内のオブジェクトにアクセスするための署名付き URL を生成します。
stat	指定されたバケットまたはオブジェクトの説明を取得します。
update	ossutil バージョンを更新します。
website	バケットの静的 Web サイトホスティングと オリジンフェッチ設定を追加、変更、クエリ、または削除します。

3.2. ダウンロードとインストール

このトピックでは、ossutil をダウンロードしてインストールする方法について説明します。

バージョンとランタイム環境

- 現在のバージョン：1.6.6

- ソースコード: **ossutil**
- ランタイム環境
 - Windows/Linux/macOS
 - サポートされているアーキテクチャ: x86 (32 ビットと 64 ビット)

ダウンロード URL

- **Linux x86 32 ビット**
- **Linux x86 64 ビット**

 **注意** 上記の URL を wget コマンドにコピーして ossutil をダウンロードする場合、`?spm=xx` セクションを URL から削除してください。

- **Windows x86 32 ビット**
- **Windows x86 64 ビット**
- **macOS x86 32 ビット**
- **macOS x86 64 ビット**

クイックインストール

オペレーティングシステムに応じてパッケージをダウンロードし、対応するバイナリファイルを実行します。

- Linux に ossutil をインストールします (例として 64 ビット Linux システムを使用)

- i. ossutil インストールパッケージをダウンロードします。

```
wget http://gosspublic.alicdn.com/ossutil/1.6.6/ossutil64
```

- ii. ファイル実行権限を変更します。

```
chmod 755 ossutil64
```

- iii. インタラクティブ処理に従って、設定ファイルを生成します。

```
./ossutil64 config
```

This command generates a configuration file. Enter the path of the configuration file. The default path is `/home/user/.ossutilconfig`. If you press Enter without specifying a different destination, the file will be generated in the default path. If you want to generate the file in another path, set the `--config-file` option to the path.

If the path of the configuration file is not entered, the default configuration file `/home/user/.ossutilconfig` is used.

A parameter is ignored if you press Enter without configuring it. To learn more about parameters, run the `help config` command.

Enter the endpoint: `http://oss-cn-shenzhen.aliyuncs.com`

Enter the AccessKey ID: your AccessKey ID

Enter the AccessKey secret: your AccessKey secret

Enter the STS token:

- **Bucket-Endpoint**: 指定したバケットのエンドポイントを指定します。Bucket-Endpoint オプションを設定した場合、バケットに対して操作を実行するとき、指定したエンドポイントが検索されます。指定したエンドポイントが存在する場合、そのエンドポイントを使用してバケットが管理されます。指定したエンドポイントが存在しない場合、Credentials オプションで指定したエンドポイントを使用してバケットが管理されます。
- **Bucket-Cname**: 指定したバケットのオリジンドメインを指定します。Bucket-Cname オプションを設定した場合、バケットに対して操作を実行するとき、指定したオリジンドメインが検索されます。指定したオリジンドメインが存在する場合、Bucket-Endpoint オプションと Credentials オプションで指定したエンドポイントはオリジンドメインに置き換えられます。オリジンドメインの詳細は、「[CNAME の設定](#)」をご参照ください。
- **AkService**: デフォルトでは、このオプションは追加されません。ECS インスタンスにバインドされた RAM ロールを使用して OSS で操作を実行する必要がある場合、このオプションが必要です。このオプションを設定するとき、ECS インスタンスにバインドされた RAM ロールに `EcsRamRoleTesting` を設定する必要があります。AkService オプションを設定した後、`accessKeyID`、`accessKeySecret`、`stsToken` オプションを設定する必要はありません。これらのオプションを設定した場合、AkService オプションではなく、これらのオプションの設定が有効になります。RAM ロールを ECS インスタンスにバインドする方法の詳細は、「[コンソールでの RAM ロールの使用](#)」をご参照ください。RAM ロールを ECS インスタンスにバインドする方法の詳細は、「[コンソールでのインスタンス RAM ロールの使用](#)」をご参照ください。

❓ 説明 設定ファイルの詳細は、「[config](#)」をご参照ください。

- Windows に ossutil をインストールします (例として 64 ビット Windows システムを使用)
 - i. ossutil インストールパッケージをダウンロードします。
 - ii. 指定したフォルダーにパッケージを展開し、`ossutil.bat` ファイルをダブルクリックします。
 - iii. 設定ファイルを生成します。パラメーターの詳細は、前述の Linux セクションに記載されている設定パラメーターをご参照ください。

```
D:\ossutil>ossutil64.exe config
```

- macOS に ossutil をインストールします (例として 64 ビット macOS システムを使用)
 - i. ossutil インストールパッケージをダウンロードします。

```
curl -o ossutilmac64 http://gosspublic.alicdn.com/ossutil/1.6.6/ossutilmac64
```

- ii. ファイル実行権限を変更します。

```
chmod 755 ossutilmac64
```

- iii. 設定ファイルを生成します。パラメーターの詳細は、前述の Linux セクションに記載されている設定パラメーターをご参照ください。

```
./ossutilmac64 config
```

3.3. 一般的なコマンド

3.3.1. appendfromfile

appendfromfile コマンドは、ローカルファイルのコンテンツを OSS の追加可能オブジェクトにアップロードする場合に使用します。

 **説明** 追加アップロードの詳細は、「[追加アップロード](#)」をご参照ください。

コマンド構文

```
./ossutil appendfromfile local_file_name oss://bucket/object [--meta=meta-value] [--payer requester]
```

追加可能オブジェクトが存在しない場合、--meta オプションを使用して、オブジェクトのメタデータを設定することができます (--meta "X-Oss-Meta-Author:test" など)。--meta オプションを使用して、既存のオブジェクトにコンテンツを追加することはできません。

注意

- 追加アップロードを使用してアップロードされたオブジェクトの名前を指定する必要があります。
- 追加アップロードを使用して作成されたオブジェクトにのみコンテンツを追加できます。
- set-meta** コマンドを使用して、オブジェクトのメタデータを変更できます。

例

追加アップロードを使用してオブジェクトをアップロードします。

```
./ossutil appendfromfile /file/test.txt oss://bucket1/test.txt
total append 64(100.00%) byte,speed is 0.00(KB/s)
local file size is 64,the object new size is 64,average speed is 0.34(KB/s)
```

一般的なオプション

次の表に、appendfromfile コマンドに追加できるオプションを示します。

オプション	説明
--meta	指定したオブジェクトのメタデータを設定します。--meta オプションを使用して、既存のオブジェクトにコンテンツを追加することはできません。
--maxupspeed	最大アップロード速度を指定します。単位: KByte/s。デフォルト値: 0 (無制限)。
--encoding-type	オブジェクト名のエンコードタイプを指定します。このオプションを指定する場合、値は url でなければなりません。このオプションが指定されていない場合、オブジェクト名はエンコードされません。バケット名は URL エンコードできません。

オプション	説明
--loglevel	ログレベルを指定します。デフォルト値は null で、ログファイルが生成されないことを示します。有効な値: - info: プロンプトログを生成します。 - debug: HTTP リクエストとレスポンスを含む詳細なログを生成します。
--payer	リクエストの支払者を指定します。このオプションをリクエスターに設定して、pay-by-requester モードを有効にすることができます。
--proxy-host	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 プロキシがサポートされています。URL の例: http://120.79.**.**:3128、socks5://120.79.**.**:1080 など。
--proxy-user	プロキシサーバーのユーザー名を指定します。デフォルト値は null です。
--proxy-pwd	プロキシサーバーのパスワードを指定します。デフォルト値は null です。

② 説明 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.3.2. bucket-encryption

bucket-encryption コマンドは、バケットの暗号化設定を追加、変更、クエリ、または削除する場合に使用します。

② 説明 バケットの暗号化の詳細は、「[サーバー側暗号化](#)」をご参照ください。

コマンド構文

- バケットの暗号化設定を追加または変更する

```
./ossutil bucket-encryption --method put oss://bucket --sse-algorithm algorithmName [--kms-masterkey-id keyid]
```

- --sse-algorithm: バケットの暗号化方式を指定します。有効値: KMS と AES256。
- --kms-masterkey-id: 暗号化に使用する CMK ID を指定します。--sse-algorithm が AES256 に設定されている場合、-kms-masterkey-id オプションを追加しないでください。--sse-algorithm パラメーターが KMS に設定されている場合、必要に応じてこのオプションを追加します。

② 説明 特定の CMK ID を使用して暗号化する機能は、パブリックプレビュー中です。この機能を有効にするには、[チケットを起票し、サポートセンターへお問い合わせください](#)。

バケットに暗号化が設定されていない場合、このコマンドを実行して、バケットに暗号化方式を追加できます。バケットに暗号化が設定されている場合、このコマンドを実行して既存の暗号化設定を上書きできます。

- バケットの暗号化設定を取得する


```
./ossutil bucket-encryption --method get oss://bucket
```

- バケットの暗号化設定を削除する

```
./ossutil bucket-encryption --method delete oss://bucket
```

例

- バケットの暗号化設定を追加または変更する

```
./ossutil bucket-encryption --method put oss://bucket --sse-algorithm KMS --kms-masterkey-id 9468da86-3509-4f8d-a61e-6eab1eac****
```

- バケットの暗号化設定を取得する

```
./ossutil bucket-encryption --method get oss://bucket
```

- バケットの暗号化設定を削除する

```
./ossutil bucket-encryption --method delete oss://bucket
```

一般的なオプション

次の表に、**bucket-encryption** コマンドに追加できるオプションを示します。

オプション	説明
--sse-algorithm	サーバー側の暗号化アルゴリズムを指定します。有効値: KMS と AES256。
--kms-masterkey-id	暗号化に使用する CMK ID を指定します。 <code>--sse-algorithm</code> が AES256 に設定されている場合、 <code>--kms-masterkey-id</code> オプションを追加しないでください。 <code>--sse-algorithm</code> パラメーターが KMS に設定されている場合、必要に応じてこのオプションを追加します。
--method	HTTP リクエストメソッドを指定します。有効値: ● put: バケットの暗号化設定を追加または変更します。 ● get: バケットの暗号化設定を取得します。 ● delete: バケットの暗号化設定を削除します。
--loglevel	ログレベルを指定します。デフォルト値は null で、ログファイルが生成されないことを示します。有効値: ● info: プロンプトログを生成します。 ● debug: 対応する HTTP リクエストとレスポンス情報を含む詳細なログを生成します。
--proxy-host	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 がサポートされています。URL の例: <code>http://120.79.***:3128</code> や <code>socks5://120.79.***:1080</code> など。

オプション	説明
--proxy-user	プロキシサーバーのユーザー名を指定します。デフォルト値は null です。
--proxy-pwd	プロキシサーバーのパスワードを指定します。デフォルト値は null です。

② 説明 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.3.3. bucket-policy

bucket-policy コマンドは、バケットのバケットポリシー設定を追加、変更、クエリ、または削除する場合に使用します。

② 説明 バケットポリシーの詳細は、「[バケットポリシー](#)」をご参照ください。

コマンド構文

- バケットポリシー設定を追加または変更する

```
./ossutil bucket-policy --method put oss://bucket local_json_file [options]
```

ossutil は *local_json_file* 設定ファイルを読み取ります。バケットにバケットポリシー設定がない場合、ossutil はこの設定ファイルに設定を書き込みます。バケットにバケットポリシー設定がある場合、新しい設定により既存の設定が上書きされます。

② 説明 *local_json_file* 設定ファイルは、次のような JSON 形式です。

```
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ram:ListObjects"
      ],
      "Principal": [
        "1234567"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {}
    }
  ]
}
```

- バケットポリシー設定を取得する

```
./ossutil bucket-policy --method get oss://bucket [local_json_file] [options]
```

local_json_file パラメーターには、設定ファイルの名前を指定します。このパラメーターが指定されている場合、バケットポリシー設定はローカルファイルとして保存されます。このパラメーターが指定されていない場合、ossutil はバケットポリシー設定を表示します。

- バケットポリシー設定を削除する

```
./ossutil bucket-policy --method delete oss://bucket [options]
```

例

- バケットポリシーを追加して、指定した IP アドレスを持つ匿名ユーザーがバケット内のすべてのリソースにアクセスできるようにする

```
./ossutil bucket-policy --method put oss://bucket1 /file/policy.json
```

policy.json 設定ファイルの内容は次のとおりです。

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "oss:*"
      ],
      "Effect": "Allow",
      "Principal": [
        "*"
      ],
      "Resource": [
        "acs:oss:*:174649585760****:bucket1",
        "acs:oss:*:174649585760****:bucket1/*"
      ],
      "Condition": {
        "IpAddress": {
          "acs:SourceIp": [
            "10.10.10.10"
          ]
        }
      }
    }
  ]
}
```

- バケットポリシー設定を取得する

```
./ossutil.exe bucket-policy --method get oss://bucket1
```

```
{
  "Version": "1",
  "Statement": [{
    {
      "Action": [
        "oss:*"
      ],
      "Effect": "Allow",
      "Principal": [
        "*"
      ],
      "Resource": [
        "acs:oss*:174649585760****:bucket1",
        "acs:oss*:174649585760****:bucket1/*"
      ],
      "Condition": {
        "IpAddress": {
          "acs:SourceIp": [
            "10.10.10.10"
          ]
        }
      }
    }
  ]
}
```

- バケットポリシー設定を削除する

```
./ossutil.exe bucket-policy --method delete oss://bucket1
```

一般的なオプション

次の表に、**bucket-policy** コマンドに追加できるオプションを示します。

オプション	説明
--method	HTTP リクエストメソッドを指定します。有効値: ● put: バケットポリシー設定を追加または変更します。 ● get: バケットポリシー設定を取得します。 ● delete: バケットポリシー設定を削除します。

オプション	説明
--loglevel	ログレベルを指定します。デフォルト値は null で、ログファイルが生成されないことを示します。有効値: • info: プロンプトログを生成します。 • debug: 対応する HTTP リクエストとレスポンス情報を含む詳細なログを生成します。
--proxy-host	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 がサポートされています。URL の例: https://120.79.**.*:3128、socks5://120.79.**.*:1080 など。
--proxy-user	プロキシサーバーのユーザー名を指定します。デフォルト値は null です。
--proxy-pwd	プロキシサーバーのパスワードを指定します。デフォルト値は null です。

② 説明 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.3.4. bucket-tagging

bucket-tagging コマンドは、バケットのタグ設定を追加、変更、クエリ、または削除する場合に使用します。

② 説明 詳細は、「[バケットのタグ](#)」をご参照ください。

コマンド構文

- バケットのタグ設定を追加または変更する

```
./ossutil bucket-tagging --method put oss://bucket tagkey1#tagvalue1 tagkey2#tagvalue2
```

バケットにタグが設定されていない場合、このコマンドを実行してバケットにタグを追加できます。バケットにタグが設定されている場合、このコマンドを実行して既存のタグを上書きできます。

? 説明

- バケットのタグを設定できるのは、バケット所有者と許可されたユーザーのみです。それ以外の場合、403 Forbidden がエラーコード AccessDenied と共に返されます。
- タグのキーと値は、シャープ記号 (#) で区切ります。
- バケットには最大 20 個のタグを設定できます。複数のタグはスペースで区切ります。
- タグのキーと値には、文字、数字、スペース、および次の特殊文字を含めることができます。

+ - = . _ : /

- タグのキーは必須です。タグのキーは最大 64 バイトで、`http://`、`https://`、または `Aliyun` で始めることはできません。
- タグの値はオプションです。タグの値は最大 128 バイトです。
- タグのキーと値は UTF-8 でエンコードする必要があります。

● バケットのタグ設定をクエリする

```
./ossutil bucket-tagging --method get oss://bucket
```

● バケットのタグ設定を削除する

```
./ossutil bucket-tagging --method delete oss://bucket
```

例

● バケットのタグ設定を追加する

```
./ossutil bucket-tagging --method put oss://bucket1 tag1#test1 tag2#test2
```

● バケットのタグ設定をクエリする

```
./ossutil bucket-tagging --method get oss://bucket1
```

● バケットのタグ設定を削除する

```
./ossutil bucket-tagging --method delete oss://bucket1
```

一般的なオプション

次の表に、`bucket-tagging` コマンドに追加できるオプションを示します。

オプション	説明
<code>--method</code>	HTTP リクエストメソッドを指定します。有効値: ● <code>put</code>: バケットのタグ設定を追加または変更します。 ● <code>get</code>: バケットのタグ設定を取得します。 ● <code>delete</code>: バケットのタグ設定を削除します。

オプション	説明
--loglevel	ログレベルを指定します。デフォルト値は null で、ログファイルが生成されないことを示します。有効値: • info: プロンプトログを生成します。 • debug: 対応する HTTP リクエストとレスポンス情報を含む詳細なログを生成します。
--proxy-host	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 がサポートされています。URL の例: http://120.79. **. **:3128、socks5://120.79. **. **:1080 など。
--proxy-user	プロキシサーバーのユーザー名を指定します。デフォルト値は null です。
--proxy-pwd	プロキシサーバーのパスワードを指定します。デフォルト値は null です。

② 説明 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.3.5. cat

cat コマンドは、特定のオブジェクトのコンテンツを ossutil にエクスポートする場合に使用します。

コマンド構文

```
./ossutil cat oss://bucket/object [--payer requester]
```

② 説明 TXT ファイルのコンテンツを表示する場合にのみ、このコマンドを使用することを推奨します。

例

特定のオブジェクトのコンテンツを表示する

```
./ossutil cat oss://bucket1/test.txt
language=ch
endpoint=oss-cn-hangzhou.aliyuncs.com

0.273027(s) elapsed
```

- バージョン管理が有効になっているバケットにあるオブジェクトの特定バージョンのコンテンツを表示する


```
./ossutil cat oss://bucket1/test.txt --version-id CAEQARiBgID8rumR2hYiIGUyOTAYZGY2MzU5MjQ5ZjIhYzQzZjNlYTAyZDE3MDRk
language=ch
endpoint=oss-cn-hangzhou.aliyuncs.com

0.375125(s) elapsed
```

`--version-id` オプションを使用するには、`ls --all-versions` コマンドを実行して、オブジェクトのバージョン ID を取得する必要があります。

❓ 説明 `--version-id` オプションは、バージョン管理が有効になっているバケット内のオブジェクトにのみ使用できます。バケットのバージョン管理を有効にするコマンドの詳細は、「[bucket-versioning](#)」をご参照ください。

一般的なオプション

次の表に、`cat` コマンドに追加できるオプションを示します。

オプション	説明
<code>--encoding-type</code>	オブジェクト名のエンコードタイプを指定します。このオプションが指定されている場合、この値を URL にする必要があります。このオプションが指定されていない場合、オブジェクト名はエンコードされません。バケット名は URL エンコードできません。
<code>--loglevel</code>	ログレベルを指定します。デフォルト値は <code>null</code> で、ログファイルが生成されないことを示します。有効値: <code>info</code>: プロンプトログを生成します。 <code>debug</code>: 対応する HTTP リクエストとレスポンス情報を含む詳細なログを生成します。
<code>--version-id</code>	オブジェクトのバージョン ID を指定します。オブジェクトが配置されているバケットでは、バージョン管理が有効になっている必要があります。
<code>--payer</code>	リクエストの支払者を指定します。pay-by-requester モードを有効にするには、このオプションを requester に設定します。
<code>--proxy-host</code>	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 がサポートされています。URL の例: <code>http://120.79. **. **:3128</code> 、 <code>socks5://120.79. **. **:1080</code> など。
<code>--proxy-user</code>	プロキシサーバーのユーザー名を指定します。デフォルト値は <code>null</code> です。
<code>--proxy-pwd</code>	プロキシサーバーのパスワードを指定します。デフォルト値は <code>null</code> です。

❓ 説明 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.3.6. config

config コマンドは、OSS アクセス情報を保存するための設定ファイルを作成する場合に使用します。-c オプションを他のコマンドに追加して、OSS アクセス情報を提供することができます。

コマンド構文

```
./ossutil config [-e endpoint] [-i id] [-k key] [-t token] [-L language] [--output-dir outdir] [-c file]
```

❓ 説明 このコマンドは、インタラクティブモードと非インタラクティブモードの両方で使用できます。セキュリティを確保するため、インタラクティブモードでコマンドを実行することを推奨します。

例

- インタラクティブモードで設定ファイルを生成する

```
./ossutil config
```

This command generates a configuration file. Enter the path of the configuration file. The default path is /home/user/.ossutilconfig. If you press Enter without specifying a different destination, the file is generated in the default path. If you want to generate the file in another path, set the --config-file option to the path.

If no configuration file path is entered, the default configuration file path /home/user/.ossutilconfig is used.

The following parameters are ignored if you press Enter without configuring them. For more information about the parameters, run the help config command.

Enter the endpoint: http://oss-cn-shenzhen.aliyuncs.com

Enter the AccessKey ID: yourAccessKeyID

Enter the AccessKey secret: yourAccessKeySecret

Enter the STS token:

- endpoint: バケットが属するリージョンのドメイン名を指定します。詳細は、「[リージョンとエンドポイント](#)」をご参照ください。
 - accessKeyID: AccessKey ID を表示する方法の詳細は、「[AccessKey の作成](#)」をご参照ください。
 - accessKeySecret: AccessKey Secret を表示する方法の詳細は、「[AccessKey の作成](#)」をご参照ください。
 - stsToken: このオプションは、一時的な STS トークンを使用して OSS バケットにアクセスする場合にのみ必要です。それ以外の場合、このパラメーターを未指定のままにすることができます。STS トークンを生成する方法の詳細は、「[一時的なアクセス資格情報](#)」をご参照ください。
- 非インタラクティブモードで設定ファイルを生成する

```
./ossutil config -e oss-cn-beijing.aliyuncs.com -i LTAIbZcdVCmQ**** -k D26oqKBudxDRBg8Wuh2EWDBrM0**** -L CH -c /myconfig
```

config コマンドに --language と --config-file 以外のオプションを指定して実行した場合、非インタラクティブモードが使用されます。すべての設定項目は、オプションを使用して指定します。

設定ファイルの形式

生成された設定ファイル内の OSS アクセス情報を変更できます。設定ファイルの形式は次のとおりです。

```
[Credentials]
language = CH
endpoint = oss.aliyuncs.com
accessKeyId = your_key_id
accessKeySecret = your_key_secret
stsToken = your_sts_token
outputDir = your_output_dir
[Bucket-Endpoint]
bucket1 = endpoint1
bucket2 = endpoint2
...
[Bucket-Cname]
bucket1 = cname1
bucket2 = cname2
...
[AkService]
ecsAk=http://100.100.100.200/latest/meta-data/Ram/security-credentials/EcsRamRoleTesting
```

- **Bucket-Endpoint**: 各バケットのエンドポイントを指定します。Bucket-Endpoint オプションを設定した場合、バケットに対して操作を実行するとき、指定したエンドポイントが検索されます。指定したエンドポイントが存在する場合、そのエンドポイントを使用してバケットが管理されます。指定したエンドポイントが存在しない場合、Credentials オプションで指定したエンドポイントを使用してバケットが管理されます。
- **Bucket-Cname**: 各バケットのオリジンドメインを指定します。Bucket-Cname オプションを設定した場合、バケットに対して操作を実行するとき、指定したオリジンドメインが検索されます。指定したオリジンドメインが存在する場合、Bucket-Endpoint オプションと Credentials オプションで指定したエンドポイントはオリジンドメインに置き換えられます。オリジンドメインの詳細は、「**CNAME の設定**」をご参照ください。
- **AkService**: デフォルトでは、このオプションは追加されません。ECS インスタンスにバインドされた RAM ロールを使用して OSS に対する操作を実行する必要がある場合、このオプションが必要です。このオプションを設定するとき、ECS インスタンスにバインドされた RAM ロールに EcsRamRoleTesting を設定する必要があります。AkService オプションを設定した後、accessKeyId、accessKeySecret、stsToken オプションを設定する必要はありません。これらのオプションを設定した場合、AkService オプションではなく、これらのオプションの設定が有効になります。RAM ロールを ECS インスタンスにバインドする方法の詳細は、「**コンソールでの RAM ロールの使用**」をご参照ください。RAM ロールを ECS インスタンスにバインドする方法の詳細は、「**コンソールでのインスタンス RAM ロールの使用**」をご参照ください。

 注意

- ossutil の新バージョンでは、インタラクティブモードで Bucket-Endpoint オプションと Bucket-Cname オプションを指定する必要はありません。ただし、2つのオプションは設定ファイルで引き続き有効です。設定ファイルの各バケットにエンドポイントかオリジンドメインを指定できます。
- ossutil は、複数のリージョンのエンドポイントを指定できます。すべてのエンドポイント設定の優先順位は次のとおりです。コマンドの `--endpoint` オプションで指定されたエンドポイント > Bucket-Cname で指定されたエンドポイント > Bucket-Endpoint で指定されたエンドポイント > Credentials で指定されたエンドポイント。コマンド実行時に `--endpoint` (短縮形 `-e`) オプションを指定した場合、`--endpoint` オプションの値が優先されます。このオプションが指定されていない場合、設定ファイルの Bucket-Cname、Bucket-Endpoint、および Credentials で指定されたエンドポイントが検索されます。最も優先度の高いエンドポイントが使用されます。

一般的なオプション

次の表に、指定した設定項目を生成するために `config` コマンドに追加できるオプションを示します。

オプション	説明
<code>-e</code> 、 <code>--endpoint</code>	設定ファイルの [Credentials] セクションのエンドポイントを指定します。
<code>-i</code> 、 <code>--access-key-id</code>	設定ファイルの [Credentials] セクションの AccessKey ID を指定します。
<code>-k</code> 、 <code>--access-key-secret</code>	設定ファイルの [Credentials] セクションの AccessKey Secret を指定します。
<code>-t</code> 、 <code>--sts-token</code>	オプション。このオプションには、設定ファイルの [Credentials] セクションの STS トークンを指定します。
<code>--output-dir</code>	出力オブジェクトが存在するディレクトリを指定します。出力オブジェクトには、 <code>cp</code> コマンドを使用して複数のオブジェクトをコピーするときのエラーが原因で生成されたレポートオブジェクトが含まれます。デフォルト値は、現在のディレクトリの <code>ossutil_output</code> ディレクトリです。
<code>-L</code> 、 <code>--language</code>	ossutil の使用言語を指定します。有効値: CH と EN。デフォルト値: CH。このオプションを CH に設定する場合、システムで UTF-8 エンコードがサポートされていることを確認してください。
<code>--loglevel</code>	ログレベルを指定します。デフォルト値は null で、ログファイルが生成されないことを示します。有効値: - info: プロンプトログを生成します。 - debug: 対応する HTTP リクエストとレスポンス情報を含む詳細なログを生成します。

 説明 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.3.7. cors

cors コマンドは、バケットの CORS 設定を追加、変更、クエリ、または削除する場合に使用します。

❓ 説明 CORS の詳細は、「**CORS ルールの設定**」をご参照ください。

コマンド構文

- 特定のバケットの CORS 設定を追加または変更する

```
./ossutil cors --method put oss://bucket local_xml_file
```

ossutil は *local_xml_file* 設定ファイルを読み取ります。バケットに CORS ルールが設定されていない場合、ossutil は対応する CORS ルールをこの設定ファイルに追加します。バケットに CORS ルールが設定されている場合、新しいルールは既存のルールを上書きします。

❓ 説明 *local_xml_file* 設定ファイルは、次のような XML 形式です。

```
<? xml version="1.0" encoding="UTF-8"? >
<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>www.aliyun.com</AllowedOrigin>
    <AllowedMethod>PUT</AllowedMethod>
    <MaxAgeSeconds>10000</MaxAgeSeconds>
  </CORSRule>
</CORSConfiguration>
```

- バケットの CORS 設定を取得する

```
./ossutil cors --method get oss://bucket [local_xml_file]
```

local_xml_file パラメーターには、設定ファイルの名前を指定します。このパラメーターが指定されている場合、ossutil は取得した CORS 設定をローカルファイルとして保存します。このパラメーターが指定されていない場合、ossutil は取得した CORS 設定を表示します。

- バケットの CORS 設定を削除する

```
./ossutil cors --method delete oss://bucket
```

例

- バケットの CORS 設定を追加する

```
./ossutil cors --method put oss://bucket1 /file/cors.xml
```

- バケットの CORS 設定を取得する

```
./ossutil cors --method get oss://bucket1 /file/cors.xml
```

- バケットの CORS 設定を削除する

```
./ossutil cors --method delete oss://bucket1
```

一般的なオプション

次の表に、cors コマンドに追加できるオプションを示します。

オプション	説明
--method	HTTP リクエストメソッドを指定します。有効値： ● put: CORS 設定を追加または変更します。 ● get: CORS 設定を取得します。 ● delete: CORS 設定を削除します。
--loglevel	ログレベルを指定します。デフォルト値は null で、ログファイルが生成されないことを示します。有効値： ● info: プロンプトログを生成します。 ● debug: 対応する HTTP リクエストとレスポンス情報を含む詳細なログを生成します。
--proxy-host	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 がサポートされています。URL の例: http://120.79. **. **:3128、socks5://120.79. **. **:1080 など。
--proxy-user	プロキシサーバーのユーザー名を指定します。デフォルト値は null です。
--proxy-pwd	プロキシサーバーのパスワードを指定します。デフォルト値は null です。

② 説明 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.3.8. cors-options

cors-options コマンドは、バケットにクロスオリジンリソース共有 (CORS) が設定されているかどうかをテストする場合に使用します。

② 説明

- CORS の詳細は、「[CORS ルールの設定](#)」をご参照ください。
- CORS を設定するコマンドの詳細は、「[cors](#)」をご参照ください。

コマンド構文

```
./ossutil cors-options --acr-method <value> --origin <value> --acr-headers <value> oss://bucket/[object]
```

cors-options コマンドは、HTTP OPTIONS リクエストを OSS に送信して、クロスオリジンリクエストが許可されているかどうかをテストする場合に使用します。

- **--acr-method**: HTTP ヘッダーの Access-Control-Request-Method フィールドの値を指定します。有効値: GET、PUT、POST、DELETE、および HEAD。
- **--origin**: HTTP ヘッダーの Origin フィールドの値を指定します。このパラメーターには、クロスオリジンリクエストの送信元を指定します。このパラメーターを使用して、バケットへのアクセスを許可するオリジンを識別します。
- **--acr-headers**: HTTP ヘッダーの Access-Control-Request-Headers フィールドの値を指定します。このパラメーターには、一般的に使用されるヘッダーを除く実際のヘッダーを指定します。複数のヘッダーを指定するには、異なるヘッダーをコンマ (,) で区切り、ヘッダーを二重引用符 (") で囲みます。例: `--acr-headers "header1,header2,header3."`

例

```
./ossutil cors-options --acr-method put --origin "www.aliyun.com" oss://bucket1
Access-Control-Allow-Methods: GET, POST, PUT
Access-Control-Allow-Origin: *
Access-Control-Max-Age: 0
```

一般的なオプション

次の表に、**cors-options** コマンドに追加できるオプションを示します。

オプション	説明
<code>--origin</code>	HTTP ヘッダーの Origin フィールドの値を指定します。このパラメーターには、クロスオリジンリクエストの送信元を指定します。このパラメーターを使用して、バケットへのアクセスを許可するオリジンを識別します。
<code>--acr-method</code>	HTTP ヘッダーの Access-Control-Request-Method フィールドの値を指定します。有効値: GET、PUT、POST、DELETE、および HEAD。
<code>--acr-headers</code>	HTTP ヘッダーの Access-Control-Request-Headers フィールドの値を指定します。このパラメーターには、一般的に使用されるヘッダーを除く実際のヘッダーを指定します。複数のヘッダーを指定するには、異なるヘッダーをコンマ (,) で区切り、ヘッダーを二重引用符 (") で囲みます。例: <code>--acr-headers "header1,header2,header3."</code>
<code>--encoding-type</code>	オブジェクト名のエンコードタイプを指定します。このオプションを指定する場合、値は url でなければなりません。このオプションが指定されていない場合、オブジェクト名はエンコードされません。バケット名は URL エンコードできません。

オプション	説明
--loglevel	ログレベルを指定します。デフォルト値は null で、ログファイルが生成されないことを示します。有効値: - info: プロンプトログを生成します。 - debug: 対応する HTTP リクエストとレスポンス情報を含む詳細なログを生成します。
--proxy-host	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 がサポートされています。URL の例: http://120.79. **. **:3128、socks5://120.79. **. **:1080 など。
--proxy-user	プロキシサーバーのユーザー名を指定します。デフォルト値は null です。
--proxy-pwd	プロキシサーバーのパスワードを指定します。デフォルト値は null です。

② 説明 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.3.9. cp

3.3.10. cp

cp コマンドは、オブジェクトをアップロード、ダウンロード、またはコピーする場合に使用します。

コマンド構文

- オブジェクトをアップロードする

```
./ossutil cp file_url cloud_url [-r] [-f] [-u] [--output-dir=odir] [--bigfile-threshold=size] [--checkpoint-dir=cdir] [--snapshot-path=sdir] [--payer requester]
```

- オブジェクトをダウンロードする

```
./ossutil cp cloud_url file_url [-r] [-f] [-u] [--output-dir=odir] [--bigfile-threshold=size] [--checkpoint-dir=cdir] [--range=x-y] [--payer requester]
```

- オブジェクトをコピーする

```
./ossutil cp cloud_url cloud_url [-r] [-f] [-u] [--output-dir=odir] [--bigfile-threshold=size] [--checkpoint-dir=cdir] [--payer requester]
```

例

- オブジェクトをアップロードする
 - 単一のオブジェクトをアップロードする

```
./ossutil cp a.txt oss://bucket/path
```


- 単一のオブジェクトをアップロードし、--meta オプションを指定する

オブジェクトをアップロードするとき、--meta オプションを使用して、オブジェクトメタデータを `header:value#header:value...` の形式で設定できます。次のコマンドを実行して `a.txt` ファイルをアップロードし、メタデータを設定します。

```
./ossutil cp a.txt oss://bucket/path --meta=Cache-Control:no-cache#Content-Encoding:gzip
```

❓ 説明 メタ設定の詳細は、「[set-meta](#)」をご参照ください。

- フォルダをアップロードする

-r オプションを cp コマンドに追加して、フォルダを OSS にアップロードできます。コマンドは次のとおりです。

```
./ossutil cp -r dir oss://bucket/path
```

❓ 説明 アップロードするオブジェクトがローカルフォルダにマップするシンボリックリンクの場合、cp コマンドを使用してオブジェクトをアップロードするとき、シンボリックリンクにスラッシュ (/) を追加します。

```
./ossutil cp -r symbolic_link/ oss://bucket/path
```

- オブジェクトまたはフォルダをアップロードし、最大アップロード速度を設定する

オブジェクトをアップロードするとき、--maxupspeed オプションを使用して最大アップロード速度 (KB/s) を設定します。デフォルト値は 0 です。これは、最大アップロード速度を設定しないことを示します。

- オブジェクトをアップロードし、最大アップロード速度を 1 MB/s に設定する

```
./ossutil cp a.jpg oss://bucket/path --maxupspeed 1024
```

- フォルダをアップロードし、最大アップロード速度を 1 MB/s に設定する

```
./ossutil cp -r dir oss://bucket/path --maxupspeed 1024
```

- 特定の条件を満たす複数のオブジェクトをアップロードする

複数のオブジェクトをコピーするとき、--include/--exclude オプションを CP コマンドに追加して、オブジェクトの選定条件を指定できます。

--include/--exclude オプションは、次の形式をサポートしています。

- *: すべての文字に一致します。たとえば、`*.txt` は、すべての `txt` オブジェクトを表します。
- ?: 任意の 1 文字に一致します。たとえば、`abc?.jpg` は、`abc` の後に任意の 1 文字を含む名前を持つ JPG オブジェクト (`abc1.jpg` など) を表します。
- [sequence]: シーケンス内の任意の文字に一致します。たとえば、`abc[1-5].jpg` は、名前が `abc` で始まり、その後にシーケンス [1-5] に含まれる数字が続く名前を持つオブジェクト (`abc1.jpg~abc5.jpg` など) を示します。

- **[! sequence]**: シーケンスに存在しない文字に一致します。たとえば、*abc[! 0-7].jpg* は、名前が *abc* で始まり、シーケンス *[0-7]* に含まれる数字が後に続かないオブジェクト (*abc0.jpg*~*abc7.jpg* など) を示します。

? 説明

- `--include "/usr/test/.jpg"` など、ディレクトリを含む名前はサポートされていません。
- `--include/--exclude` オプションを指定するには、`--recursive(-r)` オプションも指定する必要があります。
- `include` と `exclude` で指定された複数の条件を 1 つのルールに含めることができます。これらの条件が設定されている場合、ossutil は各ルールを左から右に順番に取り、最終的な一致結果を取得します。

test.txt オブジェクトが有効なフォルダーに存在する場合、マッチングルールに基づいて結果が生成されます。

- **ルール 1**: `--include "*test*" --exclude "*.txt"`。ossutil が `--include "*test*"` 条件を読み取ったとき、*test.txt* オブジェクトは条件に一致します。ossutil が `--exclude "*.txt"` 条件を読み取ったとき、*test.txt* オブジェクトは TXT 形式なので除外されます。最終的な一致結果に *test.txt* オブジェクトは含まれません。
- **ルール 2**: `--exclude "*.txt" --include "*test*"`。ossutil が `--exclude "*.txt"` 条件を読み取ったとき、一致結果に *test.txt* オブジェクトは含まれません。ossutil が `--include "*test*"` 条件を読み取ったとき、*test.txt* オブジェクト名に *test* が含まれるので条件に一致します。最終的な一致結果に *test.txt* オブジェクトが含まれます。
- **ルール 3**: `--include "*test*" --exclude "*.txt" --include "te? t.txt"`。ossutil が `--include "*test*"` 条件を読み取ったとき、*test.txt* オブジェクトは条件に一致します。ossutil が `--exclude "*.txt"` 条件を読み取ったとき、*test.txt* オブジェクトは TXT 形式なので除外されます。ossutil が `--include "te? t.txt"` 条件を読み取ったとき、*test.txt* オブジェクトは条件に一致します。最終的な一致結果に *test.txt* オブジェクトが含まれます。

例:

- *txt* 形式のすべてのオブジェクトをアップロードする

```
./ossutil cp dir/ oss://my-bucket/path --include "*.txt" -r
```

- 名前に *abc* を含み、形式が *jpg* または *txt* 以外のすべてのオブジェクトをアップロードする

```
./ossutil cp dir/ oss://my-bucket/path --include "*abc*" --exclude "*.jpg" --exclude "*.txt" -r
```

- オブジェクトをアップロードし、ストレージクラスを変更する

オブジェクトをアップロードするとき、`--meta` オプションを使用して、オブジェクトのストレージクラスを変更できます。

- オブジェクトをアップロードし、ストレージクラスを IA に設定する

```
./ossutil cp dir/sys.log oss://my-bucket/path --meta X-oss-Storage-Class:IA
```

- フォルダーをアップロードし、フォルダー内のすべてのオブジェクトのストレージクラスを IA に設定する

```
./ossutil cp dir/ oss://my-bucket/path --meta X-oss-Storage-Class:IA -r
```

- オブジェクトをアップロードし、サーバー側の暗号化方式を指定する

オブジェクトをアップロードするとき、サーバー側の暗号化方式を指定できます。オブジェクトを暗号化した後、そのオブジェクトをバケットに保存できます。サーバー側暗号化の詳細は、「[サーバー側暗号化](#)」をご参照ください。

- オブジェクトをアップロードし、サーバー側の暗号化方式を AES256 に設定する

```
./ossutil cp a.txt oss://my-bucket/path --meta=x-oss-server-side-encryption:AES256
```

- オブジェクトをアップロードし、サーバー側の暗号化方式を KMS に設定する

```
./ossutil cp a.txt oss://my-bucket/path --meta=x-oss-server-side-encryption:KMS
```

KMS を使用してオブジェクトを暗号化すると、KMS コンソールにオブジェクトのマスターキーが作成されます。KMS API 操作が呼び出されたときに少額の料金が発生します。

- 既存のオブジェクトをアップロードせずにフォルダーをアップロードする

複数のオブジェクトをアップロードするときに `--update` (短縮形 `-u`) オプションを指定すると、ターゲットに存在しないオブジェクト、または最終変更時刻がターゲットオブジェクトの最終変更時刻より後のオブジェクトのみがアップロードされます。コマンドは次のとおりです。

```
./ossutil cp -r dir oss://bucket1/path -u
```

このオプションを使用して、正常にアップロードされていないオブジェクトに対して増分アップロードを実行できます。

- フォルダをアップロードし、スナップショット情報を生成する

複数のオブジェクトをアップロードするときに `--snapshot-path` オプションを指定すると、アップロードのスナップショットが取得され、指定したディレクトリにスナップショット情報が保存されます。オブジェクトの次回アップロード時にこのオプションが指定されていると、ossutil は指定されたディレクトリからスナップショット情報を読み取り、増分アップロードを実行します。コマンドは次のとおりです。

```
./ossutil cp -r dir oss://bucket1/path --snapshot-path=path
```

注意

- `--snapshot-path` オプションは、複数のオブジェクトの増分アップロードまたはダウンロードを加速するシナリオで使用されます。このオプションは、オブジェクトの数が多く、2 回のアップロード中に他のユーザーが OSS のオブジェクトを変更しない場合に使用できます。このオプションは、オブジェクトのコピーには使用できません。
- `--snapshot-path` オプションは、アップロードまたはダウンロードされたオブジェクトのローカルの `lastModifiedTime` を記録し、記録された `lastModifiedTime` を次回アップロードまたはダウンロードするオブジェクトの `lastModifiedTime` と比較して、どのオブジェクトをスキップできるか判断します。このオプションを使用する場合、2 回のアップロードまたはダウンロード中に OSS のオブジェクトが変更されないようにしてください。2 回のアップロードまたはダウンロード中に OSS のオブジェクトが更新されるシナリオの場合、`--update` オプションを使用して、オブジェクトの増分アップロードまたはダウンロードを実行します。
- ossutil は、`snapshot-path` で指定されたディレクトリからスナップショット情報を自動的に削除しません。スナップショット情報を保持する必要がない場合、ディレクトリを削除します。
- スナップショット情報を読み書きするには、追加のオーバーヘッドが必要です。次のシナリオの場合、このオプションを使用しないことを推奨します。アップロードまたはダウンロードするオブジェクトの数が少ない。ネットワークが正しく接続されている。他のユーザーが、これらのオブジェクトに対して操作を実行する必要がある。この場合、`--update` オプションを使用して、増分アップロードまたはダウンロードを実行できます。
- `--update` オプションと `--snapshot-path` オプションは一緒に使用できます。ossutil は、最初に `--snapshot-path` オプションを使用して、オブジェクトのアップロードまたはダウンロードをスキップするかどうかを決定します。`--snapshot-path` によってアップロードまたはダウンロードがスキップされない場合、ossutil は `--update` オプションを使用して、オブジェクトのアップロードまたはダウンロードをスキップするかどうかを決定します。

- `pay-by-requester` モードが有効になっているバケットにオブジェクトをアップロードする

```
./ossutil cp dir/test.mp4 oss://payer/ --payer=requester
```

- オブジェクトをダウンロードする

- 単一のオブジェクトをダウンロードする

```
./ossutil cp oss://my-bucket/path/test1.txt /dir
```

- オブジェクトの特定の範囲をダウンロードする

--range オプションを使用して、オブジェクトの特定の範囲をダウンロードできます。

たとえば、*test1.txt* オブジェクトの 10~20 文字目をダウンロードし、ローカルデバイスにファイルとしてダウンロードします。

```
./ossutil cp oss://my-bucket/path/test1.txt /dir --range=10-20
Succeed: Total num: 1, size: 11. OK num: 1(download 1 objects).
0.290769(s) elapsed
```

- pay-by-requester モードが有効になっているバケットからオブジェクトをダウンロードする

```
./ossutil cp oss://payer/test.mp4 dir/ --payer=requester
```

- フォルダをダウンロードする

```
./ossutil cp -r oss://my-bucket/path /dir
```

- フォルダをダウンロードし、--update オプションを指定する

複数のオブジェクトをダウンロードするときに --update オプションを指定すると、ターゲットに存在しないオブジェクト、または最終変更時刻がターゲットオブジェクトの最終変更時刻より後のオブジェクトのみがダウンロードされます。コマンドは次のとおりです。

```
./ossutil cp -r oss://bucket/path /dir --update
```

このオプションを使用して、正常にダウンロードされていないオブジェクトに対して増分ダウンロードを実行できます。

- フォルダをダウンロードし、スナップショット情報を生成する

複数のオブジェクトをダウンロードするときに --snapshot-path オプションを指定すると、ダウンロードのスナップショットが取得され、指定したディレクトリにスナップショット情報が保存されます。オブジェクトの次回アップロード時にこのオプションが指定されていると、ossutil は指定されたディレクトリからスナップショット情報を読み取り、増分アップロードを実行します。詳細は、[--snapshot-path](#) をご参照ください。コマンドは次のとおりです。

```
./ossutil cp -r oss://bucket1/path dir --snapshot-path=path
```

- 特定の条件を満たす複数のオブジェクトをダウンロードする

複数のオブジェクトをダウンロードするとき、--include /--exclude を使用してオブジェクトの選定条件を指定できます。詳細は、「[cp](#)」をご参照ください。

- 形式が *jpg* 以外のオブジェクトをダウンロードする

```
./ossutil cp oss://my-bucket/path dir/ --exclude "*.jpg" -r
```

- 名前に *abc* を含み、形式が *jpg* または *txt* 以外のすべてのオブジェクトをダウンロードする

```
./ossutil cp oss://my-bucket1/path dir/ --include "abc*" --exclude "*.jpg" --exclude "*.txt" -r
```

- バージョン管理が有効になっているバケットにあるオブジェクトの特定バージョンをダウンロードする

```
./ossutil cp oss://bucket1/test.jpg dir/ --version-id CAEQARiBgID8rumR2hYiIGUyOTAYZGY2MzU5MjQ5ZjlhYzQzZjNlYTAyZDE3MDRk
```

--version-id オプションを使用するには、**ls --all-versions** コマンドを実行して、オブジェクトのバージョン ID を取得する必要があります。

④ 説明 --version-id オプションは、バージョン管理が有効になっているバケット内のオブジェクトにのみ使用できます。バケットのバージョン管理を有効にするコマンドの詳細は、「**bucket-versioning**」をご参照ください。

- オブジェクトをコピーする

オブジェクトのみをコピーできます。パートはコピーできません。リージョン間でオブジェクトをコピーすることはできません。

- 単一のオブジェクトをコピーする

```
./ossutil cp oss://bucket/path1/a oss://bucket/path2/
```

- 単一のオブジェクトをコピーし、名前を変更する

```
./ossutil cp oss://bucket/path1/a oss://bucket/path2/b
```

- 特定の条件を満たす複数のオブジェクトをコピーする

複数のオブジェクトをコピーするとき、--include/--exclude コマンドを使用してオブジェクトの選択条件を指定できます。詳細は、「**cp**」をご参照ください。

- 形式が *jpg* 以外のオブジェクトをコピーする

```
./ossutil cp oss://my-bucket1/path oss://my-bucket2/path --exclude "*.jpg" -r
```

- 名前に *abc* を含み、形式が *jpg* または *txt* 以外のすべてのオブジェクトをコピーする

```
./ossutil cp oss://my-bucket1/path oss://my-bucket2/path --include "*abc*" --exclude "*.jpg" --exclude "*.txt" -r
```

- オブジェクトをコピーし、ストレージクラスを変更する

- 既存のオブジェクトのストレージクラスをアーカイブに設定する

```
./ossutil cp oss://my-bucket/path/0104_6.jpg oss://my-bucket/path/0104_6.jpg --meta X-oss-St  
orage-Class:Archive
```

- 既存のフォルダー内のすべてのファイルのストレージクラスを標準に設定する

```
./ossutil cp oss://my-bucket/path/ oss://my-bucket/path/ --meta X-oss-Storage-Class:Stand  
ard -r
```

 注意

- cp コマンドを使用して、オブジェクトのストレージクラスをアーカイブから他のストレージクラスに変更することはできません。最初に **restore** コマンドを使用してオブジェクトを読み取り可能な状態に復元し、次に cp コマンドを使用してそのストレージクラスを変更する必要があります。
- cp コマンドを実行してオブジェクトを上書きした場合、上書きしたと見なされ、料金が発生する場合があります。IA またはアーカイブストレージクラスのオブジェクトが、作成後 30 日または 60 日以内に上書きされた場合、早期削除料金が発生します。詳細は、「**課金項目**」をご参照ください。

- オブジェクトをコピーし、サーバー側の暗号化方式を指定する

オブジェクトをコピーするときに、サーバー側の暗号化方式を指定できます。オブジェクトを暗号化した後、そのオブジェクトをバケットに保存できます。サーバー側暗号化の詳細は、「**サーバー側暗号化**」をご参照ください。

- オブジェクトをコピーし、サーバー側の暗号化方式を AES256 に設定する

```
./ossutil cp oss://bucket/path1/a oss://bucket/path2/ --meta=x-oss-server-side-encryption:AE  
S256
```

- オブジェクトをコピーし、サーバー側の暗号化方式を KMS に設定する

```
./ossutil cp oss://bucket/path1/a oss://bucket/path2/ --meta=x-oss-server-side-encryption:K  
MS
```

KMS を使用してオブジェクトを暗号化すると、KMS コンソールにオブジェクトのマスターキーが作成されます。KMS API 操作が呼び出されたときに少額の料金が発生します。

- 単一のオブジェクトをコピーし、--meta オプションを指定する

オブジェクトをコピーするとき、--meta オプションを使用して、`header:value#header:value...` 形式のオブジェクトメタデータを設定できます。

```
./ossutil cp oss://bucket/path1/a oss://bucket/path2/ --meta=Cache-Control:no-cache
```

- 同じリージョン内の異なるバケット間で複数のオブジェクトをコピーする

-r オプションを cp コマンドに追加して、同じリージョン内の異なるバケット間で複数のオブジェクトをコピーできます。

```
./ossutil cp oss://your_src_bucket/path1/ oss://your_dest_bucket/path2/ -r
```

- 同じリージョン内の異なるバケット間でオブジェクトの増分コピーを実行する

複数のオブジェクトをコピーするときに --update オプションを指定すると、ターゲットに存在しないオブジェクト、または最終変更時刻がターゲットオブジェクトの最終変更時刻より後のオブジェクトのみがコピーされます。コマンドは次のとおりです。

```
./ossutil cp oss://your_src_bucket/path1/ oss://your_dest_bucket/path2/ -r --update
```

このオプションを使用して、正常にコピーされていないオブジェクトに対して増分コピーを実行できます。

- pay-by-requester モードが有効になっているバケットから一般的なバケットにオブジェクトをコピーする

```
./ossutil cp oss://payer/test.mp4 oss://my-bucket/path --payer=requester
```

- 一般的なバケットから、pay-by-requester モードが有効になっているバケットにオブジェクトをコピーする

```
./ossutil cp oss://my-bucket/path/test.mp4 oss://payer --payer=requester
```

- バージョン管理が有効になっているバケットにあるオブジェクトの特定バージョンをコピーする

```
./ossutil cp oss://bucket1/test.jpg oss://bucket2/ --version-id CAEQARiBgID8rumR2hYilGUyOTAYZGY2MzU5MjQ5ZjNhYzQzZjNlYTAyZDE3MDRk
```

--version-id オプションを使用するには、ls --all-versions コマンドを実行して、オブジェクトのバージョン ID を取得する必要があります。

❓ 説明 --version-id オプションは、バージョン管理が有効になっているバケット内のオブジェクトにのみ使用できます。バケットのバージョン管理を有効にするコマンドの詳細は、「[bucket-versioning](#)」をご参照ください。

- オブジェクトのアップロード、ダウンロード、コピーのパフォーマンスを最適化する

--jobs オプションと --parallel オプションを cp コマンドに追加して、同時操作数を指定できます。ossutil で設定されるデフォルトの同時操作数がパフォーマンス要件を満たしていない場合、これら 2 つのオプションの値を変更してパフォーマンスを調整できます。

- --jobs オプションには、複数のファイルがアップロード、ダウンロード、またはコピーされときの同時タスク数を指定します。
- --parallel オプションには、マルチパートのアップロード、ダウンロード、またはコピー時に大きなオブジェクトで実行される同時操作数を指定します。

デフォルトでは、オブジェクトサイズに基づいて同時操作数が計算されます。このオプションは、小さいオブジェクトには機能しません。マルチパートモードでアップロード、ダウンロード、またはコピーする大きなオブジェクトのしきい値は、`--bigfile-threshold` オプションで指定できます。複数のオブジェクトをアップロード、ダウンロード、またはコピーするとき、実際の同時操作数はジョブ数に同時操作数を乗算して計算されます。

警告

- ECS インスタンスまたはサーバーのリソース (ネットワーク帯域幅、メモリ、CPU など) が制限されている場合、同時操作数を 100 未満の値に調整することを推奨します。ネットワーク、メモリ、CPU などのリソースが完全に占有されていない場合、同時操作数を増やすことができます。
- 同時操作が多すぎる場合、ossutil のアップロード、ダウンロード、コピーのパフォーマンスが低下するか、スレッド間のリソースの切り替えやスナッチが原因で EOF エラーが発生する可能性があります。この問題を解決するには、必要に応じて `--jobs` オプションと `--parallel` オプションの値を調整する必要があります。負荷テストを実行するには、最初に 2 つのオプションを小さな値に設定し、徐々に最適な値に調整します。

一般的なオプション

次の表に、`cp` コマンドに追加できるオプションを示します。

オプション	説明
<code>-r, --recursive</code>	バケット内のオブジェクトに対して操作を再帰的に実行します。このオプションが指定されている場合、このオプションをサポートするコマンドは、指定された条件を満たすバケット内のすべてのオブジェクトに対して操作を実行します。このオプションが指定されていない場合、コマンドは指定された 1 つのオブジェクトに対してのみ操作を実行します。
<code>-f, --force</code>	ユーザーに確認を求めずに操作を強制します。
<code>-u, --update</code>	ターゲットに存在しないオブジェクト、または最終変更時刻がターゲットオブジェクトの最終変更時刻より後のオブジェクトのみがアップロード、ダウンロード、またはコピーされます。
<code>--output-dir</code>	出力オブジェクトが存在するディレクトリを指定します。出力オブジェクトには、レポートオブジェクトが含まれます。レポートオブジェクトは、 <code>cp</code> コマンドで複数のオブジェクトをコピーするときに発生したエラーが原因で生成されます。デフォルト値は、現在のディレクトリの <code>ossutil_output</code> ディレクトリです。
<code>--bigfile-threshold</code>	再開可能なデータ転送を開始するオブジェクトサイズを指定します。単位: バイト。有効値: 負でない整数。デフォルト値: 100 MB。
<code>--part-size</code>	大きなオブジェクトをマルチパートアップロード、ダウンロード、またはコピーするときの各パートのサイズを指定します。単位: バイト。デフォルトでは、オブジェクトサイズに基づいて適切なパートサイズが計算されます。パフォーマンスを最適化する場合がある場合、または特別な制約の下で動作している場合、このオプションを任意の正の整数に設定できます。

オプション	説明
--checkpoint-dir	チェックポイントディレクトリを指定します。デフォルト値: .ossutil_checkpoint。再開可能なデータ転送が失敗すると、自動的にディレクトリが作成され、そのディレクトリにチェックポイント情報が記録されます。再開可能な転送が成功すると、このディレクトリは削除されます。このオプションを指定する場合、指定されたディレクトリが削除可能なことを確認してください。
--range	ダウンロードするオブジェクトの範囲を指定します。このオプション値は、3-9、3-、または -9 の形式でなければなりません。
--encoding-type	オブジェクト名のエンコードタイプを指定します。このオプションを指定する場合、値は url でなければなりません。このオプションが指定されていない場合、オブジェクト名はエンコードされません。バケット名は URL エンコードできません。
--include	*.jpg など、特定の文字列に一致するオブジェクトを含めます。
--exclude	*.txt など、特定の文字列に一致するオブジェクトを除外します。
--meta	オブジェクトのメタデータを [header:value#header:value...] の形式で設定します。例: Cache-Control:no-cache#Content-Encoding:gzip。詳細は、「 set-meta 」をご参照ください。
--acl	オブジェクトの ACL を設定します。デフォルト値: default。有効値: • default: オブジェクトは、所属するバケットの ACL を継承します。 • private • public-read • public-read-write
	複数のオブジェクトをアップロードまたはダウンロードするときに --snapshot-path オプションを指定すると、アップロードまたはダウンロードのスナップショットが取得され、指定したディレクトリにスナップショット情報が保存されます。オブジェクトの次回アップロードまたはダウンロード時にこのオプションが指定されていると、ossutil は指定されたディレクトリからスナップショット情報を読み取り、増分アップロードまたはダウンロードを実行します。

オプション	説明 注意
--snapshot-path	• --snapshot-path オプションは、オブジェクトの増分アップロードまたはダウンロードを加速するシナリオで使用されます。このオプションは、オブジェクトの数が多く、2 回のアップロード中に他のユーザーが OSS のオブジェクトを変更しない場合に使用できます。このオプションは、オブジェクトのコピーには使用できません。 • --snapshot-path オプションは、アップロードまたはダウンロードされたオブジェクトのローカルの lastModifiedTime を記録し、記録された lastModifiedTime を次回アップロードまたはダウンロードするオブジェクトの lastModifiedTime と比較して、どのオブジェクトをスキップできるか判断します。このオプションを使用する場合、2 回のアップロードまたはダウンロード中に OSS のオブジェクトが変更されないようにしてください。2 回のアップロードまたはダウンロード中に OSS のオブジェクトが更新されるシナリオの場合、--update オプションを使用して、オブジェクトの増分アップロードまたはダウンロードを実行します。 • ossutil は、snapshot-path で指定されたディレクトリからスナップショット情報を自動的に削除しません。スナップショット情報を保持する必要がない場合、ディレクトリを削除します。 • スナップショット情報を読み書きするには、追加のオーバーヘッドが必要です。次のシナリオの場合、このオプションを使用しないことを推奨します。アップロードまたはダウンロードするオブジェクトの数が少ない。ネットワークが正しく接続されている。他のユーザーが、これらのオブジェクトに対して操作を実行する必要がある。この場合、--update オプションを使用して、増分アップロードまたはダウンロードを実行できます。 • --update オプションと --snapshot-path オプションは一緒に使用できます。ossutil は、最初に --snapshot-path オプションを使用して、オブジェクトのアップロードまたはダウンロードをスキップするかどうかを決定します。--snapshot-path によってアップロードまたはダウンロードがスキップされない場合、ossutil は --update オプションを使用して、オブジェクトのアップロードまたはダウンロードをスキップするかどうかを決定します。
--disable-crc64	CRC64 を無効にします。デフォルトでは、データ送信時に CRC64 は有効化されています。
--maxuploadspeed	最大アップロード速度 (KB/s) を指定します。デフォルト値は 0 です。これは、最大アップロード速度を指定しないことを示します。
--payer	リクエストの支払者を指定します。pay-by-requester モードを有効にするには、このオプションを requester に設定します。

オプション	説明
--partition-download	オブジェクトパーティションのダウンロードに使用されます。このオプションの値は、"partition number:total number of partitions" の形式です。1:5 という値は、合計 5 つのパーティションの中からパーティション 1 がダウンロードされることを示します。オブジェクトのパーティションルールは、ossutil によって決定されます。このオプションは、ダウンロードするオブジェクトを、複数の ossutil コマンドでダウンロード可能な複数のパーティションに分割します。各 ossutil コマンドは、独自のパーティションをダウンロードします。異なる ECS インスタンスで複数の ossutil コマンドを並行して実行できます。
-j, --jobs	複数のオブジェクトで実行される同時操作数を指定します。有効値: 1 ~ 10000。デフォルト値: 3。
--parallel	単一のオブジェクトで実行される同時操作数を指定します。有効値: 1 ~ 10000。このオプションのデフォルトの値は、操作タイプとオブジェクトサイズに基づいて決定されます。
--loglevel	ログレベルを指定します。デフォルト値は null で、ログファイルが生成されないことを示します。有効値: • info: プロンプトログを生成します。 • debug: 対応する HTTP リクエストとレスポンス情報を含む詳細なログを生成します。
--retry-times	操作が失敗した場合に操作を再試行する回数を指定します。有効値: 1 ~ 500。デフォルト値: 10。
--version-id	ダウンロードまたはコピーするオブジェクトのバージョン ID を指定します。オブジェクトの存在するバケットは、バージョン管理が有効になっている必要があります。
--proxy-host	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 がサポートされています。URL の例: http://120.79. **. **:3128、socks5://120.79. **. **:1080 など。
--proxy-user	プロキシサーバーのユーザー名を指定します。デフォルト値は null です。
--proxy-pwd	プロキシサーバーのパスワードを指定します。デフォルト値は null です。

❓ 説明 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.3.11. create-symlink

create-symlink コマンドは、シンボリックリンク (ソフトリンクとも呼ばれます) を作成する場合に使用します。

❓ 説明 シンボリックリンクの詳細は、「[シンボリックリンクの設定](#)」をご参照ください。

コマンド構文

```
./ossutil create-symlink cloud_url target_object [--encoding-type url] [--payer requester] [-c file]
```

例

```
./ossutil create-symlink oss://bucket1/b oss://bucket1/path/a.txt
```

このコマンドは、bucket1 の *path* フォルダーにある *a.txt* オブジェクトに *b* という名前のシンボリックリンクオブジェクトを作成します。

② 説明 シンボリックリンクの作成時、次のチェックは実行されません。

- マップされたオブジェクトが存在するかどうか。
- マップされたオブジェクトが有効かどうか。
- マップされたオブジェクトへのアクセス許可をユーザーが持っているかどうか。
- GetObject などの操作を呼び出して、マップされたオブジェクトへのアクセスを試みるとき、前述のチェックが実行されます。
- 追加するオブジェクトが既に存在し、そのオブジェクトにアクセスする権限がある場合、新しく追加されるオブジェクトは既存のオブジェクトを上書きします。

stat または **read-symlink** コマンドを実行して、シンボリックリンクからマップされるオブジェクトを確認できます。

一般的なオプション

次の表に、**create-symlink** コマンドに追加できるオプションを示します。

オプション	説明
--encoding-type	オブジェクト名のエンコードタイプを指定します。このオプションを指定する場合、値は url でなければなりません。このオプションが指定されていない場合、オブジェクト名はエンコードされません。バケット名は URL エンコードできません。
--retry-times	操作が失敗した場合に操作を再試行する回数を指定します。有効値: 1 ~ 500。デフォルト値: 10。
--loglevel	ログレベルを指定します。デフォルト値は null で、ログファイルが生成されないことを示します。有効値: • info: プロンプトログを生成します。 • debug: 対応する HTTP リクエストとレスポンス情報を含む詳細なログを生成します。
--payer	リクエストの支払者を指定します。pay-by-requester モードを有効にするには、このオプションを requester に設定します。
--proxy-host	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 がサポートされています。URL の例: http://120.79. **. **:3128、socks5://120.79. **. **:1080 など。
--proxy-user	プロキシサーバーのユーザー名を指定します。デフォルト値は null です。

オプション	説明
--proxy-pwd	プロキシサーバーのパスワードを指定します。デフォルト値は null です。

② 説明 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.3.12. du

du コマンドは、バケットのバケット容量、または特定のオブジェクトまたはフォルダーのサイズを取得する場合に使用します。

コマンド構文

```
./ossutil du oss://bucket[/prefix] [--payer requester]
```

例

- 特定のバケットの容量をクエリする

```
./ossutil du oss://bucket1
object count:9 object sum size:471075
part count:0 part sum size:0
total du size(byte):471075
```

- 特定のフォルダーのサイズを表示する

```
./ossutil du oss://bucket1/test/
object count:2 object sum size:64482
part count:0 part sum size:0
total du size(byte):64482

0.176018(s) elapsed
```

- 特定のオブジェクトのサイズを表示する

```
./ossutil du oss://bucket1/test.txt
object count:1 object sum size:27856
part count:0 part sum size:0
total du size(byte):27856

0.241024(s) elapsed
```

- 名前が特定のプレフィックスで始まるすべてのオブジェクトのサイズを表示する

```
./ossutil du oss://bucket1/test
object count:3 object sum size:92338
part count:0 part sum size:0
total du size(byte):92338

0.184018(s) elapsed
```

一般的なオプション

次の表に、du コマンドに追加できるオプションを示します。

オプション	説明
--loglevel	ログレベルを指定します。デフォルト値は null で、ログファイルが生成されないことを示します。有効値: • info: プロンプトログを生成します。 • debug: 対応する HTTP リクエストとレスポンス情報を含む詳細なログを生成します。
--proxy-host	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 がサポートされています。URL の例: http://120.79.**.*:3128、socks5://120.79.**.*:1080 など。
--proxy-user	プロキシサーバーのユーザー名を指定します。デフォルト値は null です。
--proxy-pwd	プロキシサーバーのパスワードを指定します。デフォルト値は null です。
--payer	リクエストの支払者を指定します。pay-by-requester モードを有効にするには、このオプションを requester に設定します。

❓ 説明 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.3.13. getallpartsize

getallpartsize コマンドは、バケット内のオブジェクトに対する未完了のマルチパートアップロードタスクで生成された各パートのサイズと、すべてのパートの合計サイズを取得する場合に使用します。

コマンド構文

```
./ossutil getallpartsize oss://bucket
```

例

バケット内のオブジェクトに対する未完了のマルチパートアップロードタスクで生成された各パートのサイズと、すべてのパートの合計サイズを一覧表示します。

```
./ossutil getallpartsize oss://bucket1
PartNumber UploadId Size(Byte) Path
1 F18A92392DFD4B3FA897C267829FE417 52428800 oss://bucket1/test/test1.txt
2 F18A92392DFD4B3FA897C267829FE417 52428800 oss://bucket1/test/test1.txt
3 F18A92392DFD4B3FA897C267829FE417 52428800 oss://bucket1/test/test1.txt
4 F18A92392DFD4B3FA897C267829FE417 52428800 oss://bucket1/test/test1.txt
5 F18A92392DFD4B3FA897C267829FE417 52428800 oss://bucket1/test/test1.txt
6 F18A92392DFD4B3FA897C267829FE417 52428800 oss://bucket1/test/test1.txt

total part count:6 total part size(MB):300.00
```

一般的なオプション

次の表に、getallpartsize コマンドに追加できるオプションを示します。

オプション	説明
--encoding-type	オブジェクト名のエンコードタイプを指定します。このオプションを指定する場合、値は url でなければなりません。このオプションが指定されていない場合、オブジェクト名はエンコードされません。バケット名は URL エンコードできません。
--proxy-host	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 がサポートされています。URL の例: http://120.79.**.**:3128、socks5://120.79.**.**:1080 など。
--proxy-user	プロキシサーバーのユーザー名を指定します。デフォルト値は null です。
--proxy-pwd	プロキシサーバーのパスワードを指定します。デフォルト値は null です。

 **説明** 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.3.14. hash

hash コマンドは、ローカルファイルの CRC64 値または MD5 値を計算する場合に使用します。

コマンド構文

```
./ossutil hash file_url [--type=hashtype]
```

--type には、計算に使用されるアルゴリズムを指定します。有効値: crc64 と md5。デフォルト値: crc64。

? 説明

- **stat** コマンドを実行して、OSS のオブジェクトの CRC64 値または Content-MD5 値を表示することができます。 `X-Oss-Hash-Crc64ecma` フィールドは、CRC64 値を示します。 `Content-Md5` フィールドは、MD5 値を示します。
- OSS が CRC64 をサポートする前にアップロードされたオブジェクトの場合、**stat** コマンドは CRC64 値の取得に失敗する場合があります。
- **stat** コマンドは、追加アップロードまたはマルチパートアップロードでアップロードされたオブジェクトの Content-MD5 値の取得に失敗する場合があります。
- `--type` が md5 に設定されている場合、MD5 値と Content-MD5 値が同時に表示されます。Content-MD5 値は、MD5 ハッシュを計算して 128 ビットの数値を取得し、その数値を Base64 でエンコードすることで取得されます。
- CRC64 値は、**Standard ECMA-182** に基づいて計算されます。
- Content-MD5 の詳細は、『**RFC 1864**』をご参照ください。

例

- ローカルファイルの CRC64 値を計算する

```
./ossutil hash test.txt --type=crc64
CRC64-ECMA : 295992936743767023
```

- ローカルファイルの MD5 値を計算する

```
./ossutil hash test.txt --type=md5
MD5 : 01C3C45C03B2AF225EFAD9F911A33D73
Content-MD5 : AcPEXA0yryJe+tn5EaM9cw==
```

一般的なオプション

次の表に、**hash** コマンドに追加できるオプションを示します。

オプション	説明
<code>--type</code>	計算に使用されるアルゴリズムを指定します。有効値: <code>crc64</code> と <code>md5</code> 。デフォルト値: <code>crc64</code> 。
<code>--loglevel</code>	ログレベルを指定します。デフォルト値は <code>null</code> で、ログファイルが生成されないことを示します。有効値: • <code>info</code>: プロンプトログを生成します。 • <code>debug</code>: 対応する HTTP リクエストとレスポンス情報を含む詳細なログを生成します。

? 説明 一般的なオプションの詳細は、「**サポートされているすべてのオプションの表示**」をご参照ください。

3.3.15. help

help コマンドは、コマンドのヘルプ情報を取得する場合に使用します。コマンドの使用方法を確認する場合、help コマンドを実行することを推奨します。

コマンド構文

```
./ossutil help [command]
```

例

- すべてのコマンドのヘルプ情報を取得する

```
./ossutil help
```

- cp コマンドのヘルプ情報を取得する

```
./ossutil help cp
```

- すべてのオプションの情報を表示する

```
./ossutil help -h
```

- ls コマンドのヘルプ情報を中国語で表示する

```
./ossutil help ls -L ch
```

3.3.16. lifecycle

lifecycle コマンドは、ライフサイクル設定を追加、変更、クエリ、または削除する場合に使用します。

❓ 説明 ライフサイクルルールの詳細は、「[ライフサイクルルールの管理](#)」をご参照ください。

コマンド構文

- ライフサイクル設定を追加または変更する

```
./ossutil lifecycle --method put oss://bucket local_xml_file [options]
```

ossutil は *local_xml_file* の設定を読み取り、バケットの設定ファイルから取得したルールを追加します。設定ファイルにルール ID が存在する場合、既存のルール ID の設定は上書きされます。

② 説明 ライフサイクル設定ファイルは XML ファイルで、期限切れオブジェクトを削除するかどう、期限切れオブジェクトのストレージクラスを変更するかどうかなど、期限切れオブジェクトの処理に関するルールが記述されています。ニーズに基づいてルールを設定できます。次のコードは、ライフサイクル設定の例を示しています。

```
<? xml version="1.0" encoding="UTF-8"? >
<LifecycleConfiguration>
  <Rule>
    <ID>RuleID</ID>
    <Prefix>Prefix</Prefix>
    <Status>Status</Status>
    <Expiration>
      <Days>Days</Days>
    </Expiration>
    <Transition>
      <Days>Days</Days>
      <StorageClass>StorageClass</StorageClass>
    </Transition>
    <AbortMultipartUpload>
      <Days>Days</Days>
    </AbortMultipartUpload>
  </Rule>
</LifecycleConfiguration>
```

ライフサイクル設定の詳細は、「[ライフサイクルルールの設定例](#)」をご参照ください。

- ライフサイクル設定を取得する

```
ossutil lifecycle --method get oss://bucket [local_file]
```

local_xml_file パラメーターには、設定ファイルの名前を指定します。このパラメーターが指定されている場合、ossutil は取得したライフサイクル設定をローカルファイルとして保存します。このパラメーターが指定されていない場合、ossutil は取得したライフサイクル設定を表示します。

- ライフサイクル設定を削除する

```
./ossutil lifecycle --method delete oss://bucket
```

例

- 10 日間アップロードされ、プレフィックスが *test/* のオブジェクトのストレージクラスを IA に変更するライフサイクルルールを追加する

```
./ossutil lifecycle --method put oss://bucket1 /file/lifecycle.xml
```

設定ファイル *lifecycle.xml* のコンテンツは、次のとおりです。

```
<? xml version="1.0" encoding="UTF-8"? >
<LifecycleConfiguration>
<Rule>
<ID>1</ID>
<Prefix>test/</Prefix>
<Status>Enabled</Status>
<Transition>
<Days>10</Days>
<StorageClass>IA</StorageClass>
</Transition>
</Rule>
</LifecycleConfiguration>
```

- 2019 年 1 月 1 日より前に変更され、プレフィックスが *test/* で始まるオブジェクトを削除するライフサイクルルールを追加する

```
./ossutil lifecycle --method put oss://bucket1 /file/lifecycle.xml
```

設定ファイル *lifecycle.xml* のコンテンツは、次のとおりです。

```
<? xml version="1.0" encoding="UTF-8"? >
<LifecycleConfiguration>
<Rule>
<ID>2</ID>
<Prefix>test/</Prefix>
<Status>Enabled</Status>
<Expiration>
<CreatedBeforeDate>2014-10-11T00:00:00.000Z</CreatedBeforeDate>
</Expiration>
</Rule>
</LifecycleConfiguration>
```

- ライフサイクル設定を取得する

```
./ossutil lifecycle --method get oss://bucket1 /file/lifecycle.xml
```

- ライフサイクル設定を削除する

```
./ossutil lifecycle --method delete oss://bucket1
```

一般的なオプション

次の表に、*lifecycle* コマンドに追加できるオプションを示します。

オプション	説明
--method	HTTP リクエストメソッドを指定します。有効値: • put: 設定を追加または変更します。 • get: 設定を取得します。 • delete: 設定を削除します。
--loglevel	ログレベルを指定します。デフォルト値は null で、ログファイルが生成されないことを示します。有効値: • info: プロンプトログを生成します。 • debug: HTTP リクエストとレスポンスを含む詳細なログを生成します。
--proxy-host	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 プロキシがサポートされています。URL の例: http://120.79. **.**:3128、socks5://120.79. **. **:1080 など。
--proxy-user	プロキシサーバーのユーザー名を指定します。デフォルト値は null です。
--proxy-pwd	プロキシサーバーのパスワードを指定します。デフォルト値は null です。

 **説明** 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.3.17. listpart

listpart コマンドは、マルチパートアップロードタスクが失敗したときに生成されるパートの情報を一覧表示する場合に使用します。

コマンド構文

```
./ossutil listpart oss://bucket/object uploadid [options]
```

このコマンドを実行してパートの情報を取得する前に、**ls -m** コマンドを実行して、バケットとオブジェクトに対応するアップロード ID の情報を取得します。

例

マルチパートアップロードタスクが完了していない場合に生成されるオブジェクトのパートを一覧表示します。

```
./ossutil listpart oss://bucket/object 15754AF7980C4DFB8193F190837520BB
```

一般的なオプション

次の表に、**listpart** コマンドに追加できるオプションを示します。

オプション	説明
--encoding-type	オブジェクト名のエンコードタイプを指定します。このオプションを指定する場合、値は url でなければなりません。このオプションが指定されていない場合、オブジェクト名はエンコードされません。バケット名は URL エンコードできません。
--proxy-host	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 がサポートされています。URL の例: http://120.79.**.**:3128、socks5://120.79.**.**:1080 など。
--proxy-user	プロキシサーバーのユーザー名を指定します。デフォルト値は null です。
--proxy-pwd	プロキシサーバーのパスワードを指定します。デフォルト値は null です。

② 説明 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.3.18. logging

logging コマンドは、特定のバケットのログ設定を追加、変更、クエリ、または削除する場合に使用します。

② 説明 ログの詳細は、「[サーバーアクセスログ](#)」をご参照ください。

コマンド構文

- ログ設定を追加または変更する

```
./ossutil logging --method put oss://bucket oss://target-bucket/[prefix]
```

バケットのログが無効になっている場合、このコマンドを実行して、target-bucket パラメーターで指定したバケット内のオブジェクトとしてアクセスログを保存できます。また、バケットのログが有効になっている場合、このコマンドを実行して、アクセスログが保存されるディレクトリを変更できます。

prefix パラメーターには、アクセスログのプレフィックスと、バケットアクセスログが保存されるディレクトリを指定します。このパラメーターを指定した場合、target-bucket パラメーターで指定したバケットのディレクトリにアクセスログが保存されます。このパラメーターを指定していない場合、target-bucket パラメーターで指定したバケットのルートディレクトリにアクセスログが保存されます。ログファイルの命名規則の詳細は、「[ログの設定](#)」をご参照ください。

- ログ設定を取得する

```
./ossutil logging --method get oss://bucket [local_xml_file]
```

local_xml_file パラメーターには、設定ファイルの名前を指定します。このパラメーターを指定した場合、取得したログ設定はローカルファイルとして保存されます。このパラメーターを指定していない場合、取得したログ設定が表示されます。

- ログ設定を削除する

```
./ossutil logging --method delete oss://bucket
```

例

- ログ設定を追加する

```
./ossutil logging --method put oss://bucket1 oss://bucket2/logging
```

- ログ設定を取得する

```
./ossutil logging --method get oss://bucket1 /file/logging.xml
```

- ログ設定を削除する

```
./ossutil logging --method delete oss://bucket1
```

一般的なオプション

次の表に、logging コマンドに追加できるオプションを示します。

オプション	説明
--method	HTTP リクエストメソッドを指定します。有効値： - put: ログ設定を追加または変更します。 - get: ログ設定を取得します。 - delete: ログ設定を削除します。
--loglevel	ログレベルを指定します。デフォルト値は null で、ログファイルが生成されないことを示します。有効値： - info: プロンプトログを生成します。 - debug: 対応する HTTP リクエストとレスポンス情報を含む詳細なログを生成します。
--proxy-host	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 がサポートされています。URL の例: http://120.79.**.*:3128、socks5://120.79.**.*:1080 など。
--proxy-user	プロキシサーバーのユーザー名を指定します。デフォルト値は null です。
--proxy-pwd	プロキシサーバーのパスワードを指定します。デフォルト値は null です。

 **説明** 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.3.19. ls

ls コマンドは、バケット、オブジェクト、またはパートを一覧表示する場合に使用します。

コマンド構文

```
./ossutil ls [oss://bucket[/prefix]] [-s] [-d] [--limited-num num] [--marker marker] [--upload-id-marker u
marker] [--payer requester] [-c file] [--include include-pattern] [--exclude exclude-pattern] [--payer req
uester]
```

例

- バケットを一覧表示する

```
./ossutil ls
CreationTime Region StorageClass BucketName
2016-10-21 16:18:37 +0800 CST oss-cn-hangzhou Archive oss://bucket1
2016-12-01 15:06:21 +0800 CST oss-cn-hangzhou Standard oss://bucket2
2016-07-20 10:36:24 +0800 CST oss-cn-hangzhou IA oss://bucket3
Bucket Number is:3
0.252174(s) elapsed
```

 **説明** `./ossutil ls oss://` コマンドを使用して、バケットを一覧表示することもできます。

- ページごとにバケットを一覧表示する

```
./ossutil ls oss:// --limited-num=${num} --marker=${bucketname}
```

バケット数が多い場合、`--limited-num` と `--marker` オプションを使用して、ページごとにバケットを一覧表示できます。

- `--limited-num` は、各ページに表示されるエントリ数の制御に使用されます。
- `--marker` には、一覧表示を開始するバケットを指定します。marker 値が設定された後、表示される結果は、先頭のバケットからアルファベット順に返されます。ほとんどの場合、このパラメーター値には、前のページに表示された最後のバケット名を設定します。

次のコードは、各ページに先頭 2 つのバケットを一覧表示する方法の例を示しています。

```
./ossutil ls oss:// --limited-num=1 -s
oss://bucket1
Bucket Number is:1
0.303869(s) elapsed
./ossutil ls oss:// --limited-num=1 -s --marker=bucket1
oss://bucket2
Bucket Number is:1
0.257636(s) elapsed
```

- 特定のバケット内のすべてのオブジェクトを一覧表示する


```
./ossutil ls oss://bucket1
LastModifiedTime Size(B) StorageClass ETAG ObjectName
2016-12-0115:06:37 +0800 CST 10363812 Standard 61DE142E5AFF9A6748707D4A77BFBCFB oss://bucket1/a1
2016-12-0115:06:42 +0800 CST 10363812 Standard 61DE142E5AFF9A6748707D4A77BFBCFB oss://bucket1/a2
2016-12-0115:06:45 +0800 CST 10363812 Standard 61DE142E5AFF9A6748707D4A77BFBCFB oss://bucket1/a3
Object Number is:3
0.007379(s) elapsed
```

- すべてのオブジェクトをページごとに一覧表示する

```
./ossutil ls oss://bucket --limited-num=${num} --marker=${obj}
```

ページごとにバケットを一覧表示する操作と同様に、--limited-num オプションと --marker オプションを使用して、ページごとにオブジェクトを一覧表示できます。次のコードは、ルートディレクトリ bucket1 の先頭 2 つのオブジェクトを一覧表示する方法の例を示しています。

```
./ossutil ls oss://bucket1 --limited-num=1
LastModifiedTime Size(B) StorageClass ETAG ObjectName
2016-12-0115:06:37 +0800 CST 10363812 Standard 61DE142E5AFF9A6748707D4A77BFBCFB oss://bucket1/a1
Object Number is:1
0.007379(s) elapsed
$./ossutil ls oss://bucket1 --limited-num=1 --marker=a1
LastModifiedTime Size(B) StorageClass ETAG ObjectName
2016-12-0115:06:42 +0800 CST 10363812 Standard 61DE142E5AFF9A6748707D4A77BFBCFB oss://bucket1/a2
Object Number is:1
0.008392(s) elapsed
```

- 特定の条件を満たすオブジェクトを一覧表示する

--include/--exclude を設定して、特定の条件を満たすオブジェクトを一覧表示できます。--include/--exclude の詳細は、「[cp](#)」をご参照ください。

- 形式が *jpg* 以外のすべてのオブジェクトを一覧表示する

```
./ossutil ls oss://my-bucket/path --exclude "*.jpg" -r
```

- 名前に *abc* を含み、形式が *jpg* または *txt* 以外のすべてのオブジェクトを一覧表示する

```
./ossutil ls oss://my-bucket1/path --include "*abc*" --exclude "*.jpg" --exclude "*.txt" -r
```

- アップロード ID を一覧表示する

-m オプションを ls コマンドに追加して、特定のプレフィックスで始まるオブジェクトに対する未完了のマルチパートアップロードタスクのアップロード ID を一覧表示します。

```
./ossutil ls oss://bucket1/obj1 -m
InitiatedTime UploadID ObjectName
2017-01-13 03:45:26 +0000 CST 15754AF7980C4DFB8193F190837520BB oss://bucket1/obj1
2017-01-13 03:43:13 +0000 CST 2A1F9B4A95E341BD9285CC42BB950EE0 oss://bucket1/obj1
UploadId Number is: 2
0.070070(s) elapsed
```

- すべてのオブジェクトと未完了のマルチパートアップロードタスクを一覧表示する

-a オプションを ls コマンドに追加して、特定のプレフィックスで始まるオブジェクトに対する未完了のマルチパートアップロードタスクのアップロード ID と、このプレフィックスで始まるアップロード済みのオブジェクトを一覧表示します。

```
./ossutil ls oss://bucket1 -a
LastModifiedTime Size(B) StorageClass ETAG ObjectName
2015-06-0514:06:29 +0000 CST 201933 Standard 7E2F4A7F1AC9D2F0996E8332D5EA5B41 oss://bucket1/dir1/obj11
2015-06-0514:36:21 +0000 CST 201933 Standard 6185CA2E8EB8510A61B3A845EAFE4174 oss://bucket1/obj1
2016-04-0814:50:47 +0000 CST 6476984 Standard 4F16FDAE7AC404CEC8B727FCC67779D6 oss://bucket1/sample.txt
Object Number is:3
InitiatedTime UploadID ObjectName
2017-01-1303:45:26 +0000 CST 15754AF7980C4DFB8193F190837520BB oss://bucket1/obj1
2017-01-1303:43:13 +0000 CST 2A1F9B4A95E341BD9285CC42BB950EE0 oss://bucket1/obj1
2017-01-1303:45:25 +0000 CST 3998971ACAF94AD9AC48EAC1988BE863 oss://bucket1/obj2
2017-01-2011:16:21 +0800 CST A20157A7B2FEC4670626DAE0F4C0073C oss://bucket1/tobj
UploadId Number is:4
0.191289(s) elapsed
```

- 一覧表示の結果を短い形式で表示する

```
./ossutil ls oss://bucket1 -s
oss://bucket1/a1
oss://bucket1/a2
oss://bucket1/a3
Object Number is:3
0.007379(s) elapsed
```

- バケットをディレクトリ形式で一覧表示する

-d オプションを使用して、すべてのサブディレクトリ内のすべてのオブジェクトを再帰的に表示するのではなく、現在のディレクトリ内のオブジェクトとサブディレクトリを一覧表示します。例：

```
./ossutil ls oss://bucket1 -s -d
oss://bucket1/obj1
oss://bucket1/sample.txt
oss://bucket1/dir1/
Object and Directory Number is:3
0.119884(s) elapsed
```

- 現在のディレクトリ内で、特定のプレフィックスで始まるオブジェクトまたはサブディレクトリを一覧表示する

```
./ossutil ls oss://bucket1/test -d
oss://bucket1/test.jpg
oss://bucket1/test/
Object and Directory Number is: 2
```

- pay-by-requester モードでバケットを一覧表示する

```
./ossutil ls oss://bucket --payer=requester
```

- バージョン管理が有効になっているバケット内の全オブジェクトの特定バージョンを一覧表示する

```
./ossutil ls oss://bucket1 --all-versions
LastModifiedTime Size(B) StorageClass ETAG VERSIONID IS-LATEST DELETE-MARKER ObjectName
2019-06-11 10:54:51 +0800 CST 0 CAEQARiBgICUsOuR2hYiIDl3NWVjNmEyYmM0NTRkZWNiMTkxY2VjMD
MwZjFlMDA3 true true oss://bucket1/test1.jpg
2019-06-11 11:03:37 +0800 CST 0 CAEQARiBgIDZtvuR2hYiIDNhYjRkN2M5NTA5OTRlN2Q4YTZyODQwMz
Q4NDYwZDdm true true oss://bucket1/test.jpg
2019-06-11 10:53:46 +0800 CST 118076 Standard FFDB300F053AAF06F4C4C58A4869C427 CAEQARiBgID8
rumR2hYiIGUyOTAYzGY2MzU5MjQ5ZjIhYzQzZjNlYTAyZDE3MDRk false false oss://bucket1/test1.jpg
2019-06-11 11:02:05 +0800 CST 345374 Standard 078A9852BCF81DC4811E6EDCBFD121BE CAEQARiBgIC
Nz_iR2hYiIGjZTBjNDQxYWRhNTQ2ZTNiNmMzYzQ1YzMzM5ZjUw false false oss://bucket1/test.jpg
Object Number is: 4

0.692000(s) elapsed
```

- バージョン管理が有効になっているバケット内の特定のオブジェクトの全バージョンを一覧表示する

```
./ossutil ls oss://bucket1/test.jpg --all-versions

LastModifiedTime Size(B) StorageClass ETAG VERSIONID IS-LATEST DELETE-MARKER ObjectName
2019-06-11 11:03:37 +0800 CST 0 CAEQARiBgIDZtvuR2hYiIDNhYjRkN2M5NTA5OTRIN2Q4YTYzODQwMz
Q4NDYwZDdm true true oss://bucket1/test.jpg
2019-06-11 11:02:05 +0800 CST 345374 Standard 078A9852BCF81DC4811E6EDCBFD121BE CAEQARiBgIC
Nz_iR2hYiIGjZTBjNDQxYWRhNTQ2ZTNiNmMzYzQ1YzMzMMA5ZjUw false false oss://bucket1/test.jpg
Object Number is: 2

0.361000(s) elapsed
```

一般的なオプション

次の表に、一覧表示する項目と一覧表示の方法を指定するために ls コマンドに追加できるオプションを示します。

オプション	説明
-s	項目を短い形式で一覧表示します。
-d	すべてのサブディレクトリ内のすべてのオブジェクトを再帰的に表示するのではなく、現在のディレクトリ内のオブジェクトとサブディレクトリを一覧表示します。
-m	バケット内の未完了のマルチパートアップロードタスクのみを一覧表示します。オブジェクトは表示されません。
-a	バケット内のオブジェクトと未完了のマルチパートアップロードタスクの両方を一覧表示します。
--limited-num	返される結果の最大数を指定します。ページごとにオブジェクトを一覧表示するには、このオプションを <code>--marker</code> オプションと一緒に使用します。
--marker	一覧表示を開始するバケット名、オブジェクト名、またはマルチパートアップロード ID を指定します。
--upload-id-marker	一覧表示を開始するマルチパートアップロード ID を指定します。
--payer	リクエストの支払者を指定します。pay-by-requester モードを有効にするには、このオプションを requester に設定します。
--encoding-type	オブジェクト名のエンコードタイプを指定します。このオプションを指定する場合、値は url でなければなりません。このオプションが指定されていない場合、オブジェクト名はエンコードされません。バケット名は URL エンコードできません。
--loglevel	ログレベルを指定します。デフォルト値は null で、ログファイルが生成されないことを示します。有効値: • info: プロンプトログを生成します。 • debug: 対応する HTTP リクエストとレスポンス情報を含む詳細なログを生成します。

オプション	説明
--include	*.jpg など、特定の文字列に一致するオブジェクトを含めます。
--exclude	*.txt など、特定の文字列に一致するオブジェクトを除外します。
--all-versions	特定のオブジェクトの全バージョンを一覧表示します。このオプションは、バージョン管理が有効になっているバケット内のオブジェクトにのみ使用できます。
--version-id-marker	一覧表示を開始するオブジェクトバージョンを指定します。
--proxy-host	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 がサポートされています。URL の例: http://120.79. **. **:3128、socks5://120.79. **. **:1080 など。
--proxy-user	プロキシサーバーのユーザー名を指定します。デフォルト値は null です。
--proxy-pwd	プロキシサーバーのパスワードを指定します。デフォルト値は null です。

 **説明** 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.3.20. mb

mb コマンドは、バケットを作成する場合に使用します。

コマンド構文

```
./ossutil mb oss://bucketname [--acl=ACL][--storage-class sc][-c file]
```

例

- バケットを作成する

```
./ossutil mb oss://bucket1
```

 **注意** バケット名は一意である必要があります。指定した名前のバケットが既に存在する場合、エラーメッセージが表示されます。

- バケットの作成時にアクセス制御リスト (ACL) を指定する

```
./ossutil mb oss://bucket1 --acl=public-read-write
```

- バケットの作成時にストレージクラスを指定する

```
./ossutil mb oss://bucket1 --storage-class IA
```

- 特定のリージョンにバケットを作成する

```
./ossutil mb oss://bucket1 -e oss-cn-beijing.aliyuncs.com
```

- 特定の設定ファイルからバケットを作成する

```
./ossutil mb oss://bucket1 -c your_config_file_path
```

一般的なオプション

次の表に、バケット属性を指定するために mb コマンドに追加できるオプションを示します。

オプション	説明
--acl	バケットの ACL を指定します。デフォルト値: private。有効値: • private • public-read • public-read-write ACL の詳細は、「 ACL 」をご参照ください。
--storage-class	バケットのデフォルトのストレージクラスを指定します。デフォルト値: Standard。有効値: • Standard • IA • Archive ストレージクラスの詳細は、「 概要 」をご参照ください。
-c, --config-file	バケットの作成に使用される設定ファイルを指定します。このオプションを指定しない場合、デフォルトの設定ファイルを使用してバケットが作成されます。設定ファイルの詳細は、「 config 」をご参照ください。
-e, --endpoint	バケットが存在するリージョンに対応するエンドポイントを指定します。このオプションが指定されていない場合、デフォルトの設定ファイルのエンドポイントを使用して、バケットのリージョンが指定されます。
-L, --language	ossutil の使用言語を指定します。有効値: CH と EN。デフォルト値: CH。このオプションを CH に設定する場合、システムで UTF-8 エンコードがサポートされていることを確認してください。
--loglevel	ログレベルを指定します。デフォルト値は null で、ログファイルが生成されないことを示します。有効値: • info: プロンプトログを生成します。 • debug: 対応する HTTP リクエストとレスポンス情報を含む詳細なログを生成します。
--retry-times	操作が失敗した場合に操作を再試行する回数を指定します。有効値: 1~500。デフォルト値: 10。
--proxy-host	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 がサポートされています。URL の例: http://120.79.**.*:3128、socks5://120.79.**.*:1080 など。
--proxy-user	プロキシサーバーのユーザー名を指定します。デフォルト値は null です。
--proxy-pwd	プロキシサーバーのパスワードを指定します。デフォルト値は null です。

② 説明 一般的なオプションの詳細は、「サポートされているすべてのオプションの表示」をご参照ください。

3.3.21. mkdir

mkdir コマンドは、バケットにディレクトリを作成する場合に使用します。

コマンド構文

```
./ossutil mkdir oss://bucket/dir/
```

例

- ディレクトリを作成する

```
./ossutil mkdir oss://bucket1/dir1/
```

② 説明

- ディレクトリはスラッシュ (/) で終わる必要があります。指定したディレクトリ名がスラッシュで終わっていない場合、自動的にスラッシュ (/) が1つ追加されます。
- 指定したディレクトリが既に存在する場合、エラーが表示されます。

- 複数階層のディレクトリを作成する

```
./ossutil mkdir oss://bucket/dir1/dir2/
```

複数階層のディレクトリを作成する場合、ディレクトリの最後の階層のみが作成されます。*dir2/*が削除され、*dir1/*にオブジェクトが存在しない場合、コマンドの実行後に *dir1/* も削除されます。

一般的なオプション

次の表に、mkdir コマンドに追加できるオプションを示します。

オプション	説明
--encoding-type	オブジェクト名のエンコードタイプを指定します。このオプションを指定する場合、値は url でなければなりません。このオプションが指定されていない場合、オブジェクト名はエンコードされません。バケット名は URL エンコードできません。
--loglevel	ログレベルを指定します。デフォルト値は null で、ログファイルが生成されないことを示します。有効値: - info: プロンプトログを生成します。 - debug: 対応する HTTP リクエストとレスポンス情報を含む詳細なログを生成します。
--proxy-host	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 がサポートされています。URL の例: http://120.79.**.*:3128、socks5://120.79.**.*:1080 など。

オプション	説明
--proxy-user	プロキシサーバーのユーザー名を指定します。デフォルト値は null です。
--proxy-pwd	プロキシサーバーのパスワードを指定します。デフォルト値は null です。

② 説明 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.3.22. object-tagging

object-tagging コマンドは、オブジェクトのタグ設定を追加、変更、クエリ、または削除する場合に使用します。

② 説明 オブジェクトのタグの詳細は、「[オブジェクトのタグ付け](#)」をご参照ください。

コマンド構文

- オブジェクトのタグ設定を追加または変更する

```
./ossutil object-tagging --method put oss://bucket[/prefix] key#value [--encoding-type url] [-r] [--payer requester] [--version-id versionId] [-c file]
```

オブジェクトにタグが設定されていない場合、このコマンドを実行して、タグをオブジェクトに追加できます。オブジェクトにタグが設定されている場合、このコマンドを実行して既存のタグを上書きできます。

② 説明

- オブジェクトごとに最大 10 個のタグを設定できます。各オブジェクトタグには一意のタグキーが必要です。
- タグキーは最大 128 文字です。タグの値は最大 256 文字です。
- キーと値は大文字と小文字が区別されます。
- タグのキーと値には、文字、数字、スペース、および次の特殊文字を含めることができます。
+ = . _ : /
- オブジェクトのタグの読み書き権限を持っているのは、バケット所有者と許可されたユーザーのみです。これらの権限は、オブジェクトの ACL に依存しません。
- リージョン間レプリケーション時に、オブジェクトタグもターゲットバケットにレプリケートされます。

- オブジェクトのタグ設定をクエリする

```
./ossutil object-tagging --method get oss://bucket[/prefix] [--encoding-type url] [-r] [--payer requester] [--version-id versionId] [-c file]
```


- オブジェクトのタグ設定を削除する

```
./ossutil object-tagging --method delete oss://bucket[/prefix] [--encoding-type url] [-r] [--payer requester] [--version-id versionId] [-c file]
```

例

- オブジェクトのタグ設定を追加する

```
./ossutil object-tagging --method put oss://bucket1/test.jpg a#1 b#2 c#3
0.168034(s) elapsed
```

- オブジェクトのタグ設定をクエリする

```
./ossutil object-tagging --method get oss://bucket1/test.jpg
object index tag index tag key tag value object
-----
1 0 "a" "1" 123/test.jpg
1 1 "b" "2" 123/test.jpg
1 2 "c" "3" 123/test.jpg

0.228023(s) elapsed
```

- オブジェクトのタグ設定を削除する

```
./ossutil object-tagging --method delete oss://bucket1/test.jpg
0.200020(s) elapsed
```

一般的なオプション

次の表に、object-tagging コマンドに追加できるオプションを示します。

オプション	説明
--method	HTTP リクエストメソッドを指定します。有効値: ● put: オブジェクトのタグ設定を追加または変更します。 ● get: オブジェクトのタグ設定を取得します。 ● delete: オブジェクトのタグ設定を削除します。
--loglevel	ログレベルを指定します。デフォルト値は null で、ログファイルが生成されないことを示します。有効値: ● info: プロンプトログを生成します。 ● debug: 対応する HTTP リクエストとレスポンス情報を含む詳細なログを生成します。
--proxy-host	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 がサポートされています。URL の例: http://120.79.**.*:3128、socks5://120.79.**.*:1080 など。

オプション	説明
--proxy-user	プロキシサーバーのユーザー名を指定します。デフォルト値は null です。
--proxy-pwd	プロキシサーバーのパスワードを指定します。デフォルト値は null です。
--payer	リクエストの支払者を指定します。pay-by-requester モードを有効にするには、このオプションを requester に設定します。
-j, --jobs	複数のオブジェクトが操作される場合の同時タスク数を指定します。有効値: 1 ~ 10000。デフォルト値: 3。
-r, --recursive	バケット内のオブジェクトに対して操作を再帰的に実行します。このオプションが指定されている場合、このオプションをサポートするコマンドは、指定された条件を満たすバケット内のすべてのオブジェクトに対して操作を実行します。このオプションが指定されていない場合、コマンドは指定された 1 つのオブジェクトに対してのみ操作を実行します。
--encoding-type	オブジェクト名のエンコードタイプを指定します。このオプションを指定する場合、値は url でなければなりません。このオプションが指定されていない場合、オブジェクト名はエンコードされません。バケット名は URL エンコードできません。

❓ 説明 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.3.23. probe

probe コマンドは、OSS へのアクセスをモニタリングし、アップロードプロセスやダウンロードプロセス中にネットワーク障害や不適切なパラメーター設定によって発生した問題をトラブルシューティングする場合に使用します。

コマンド構文

- オブジェクトの URL を使用してバケットからローカルデバイスにオブジェクトをダウンロードし、トラブルシューティングレポートを生成する

```
./ossutil probe --download --url http_url [--addr=domain_name] [file_name]
```

オブジェクトの URL を使用してオブジェクトをバケットからローカルデバイスにダウンロードした後、ネットワーク伝送品質をテストし、トラブルシューティングレポートを生成できます。

- `--url`: 特定のバケット内のオブジェクトの URL を指定します。
 - オブジェクトの ACL が公開読み取り/書き込みの場合、URL は署名を含みません。例: `https://bucketname.oss-cn-beijing.aliyuncs.com/myphoto.jpg`
 - オブジェクトの ACL が非公開の場合、URL は署名を含み、先頭と末尾は二重引用符 (") です。
例: `"https://bucketname.oss-cn-beijing.aliyuncs.com/myphoto.jpg?Expires=1552015472&OSSAccessKeyId=TMP.xxxxxxxx5r9f1FV12y8_Qis6LUVmvoSCUSs7aboCCHtydQ0axN32Sn-UvyY3AAAwLAIUarYNLcO87AKMEcE503AxxxxxxoCFAQuRdZYyVFyqOW8QkGAN-bamUiQ&Signature=bla4llbMbldrl7rwckr%2FXxvTtxw%3D"`

❓ 説明 オブジェクトの URL を取得する方法の詳細は、「[URL の取得](#)」をご参照ください。

- `--addr=domain_name`: オプション。このパラメーターには、オブジェクトのダウンロード時に ping するドメイン名または IP アドレスを指定します。このパラメーターが指定されていない場合、ping 操作は実行されません。
 - `--addr` オプションがデフォルト値に設定されている場合、`www.aliyun.com` に対して ping が実行されます。
 - `--addr` オプションがドメイン名または IP アドレスに設定されている場合、指定されたドメイン名または IP アドレスに対して ping が実行されます。
- `file_name`: オプション。このパラメーターには、ダウンロードしたオブジェクトを保存するファイルパスを指定します。`file_name` が指定されていない場合、ossutil はダウンロードしたオブジェクトを現在のディレクトリに保存し、オブジェクト名を決定します。`file_name` を使用してオブジェクト名やディレクトリ名を指定する場合、ossutil は指定された `file_name` 値を使用して、ダウンロードしたオブジェクトに名前を付けるか、ダウンロードしたオブジェクトを `file_name` で指定したディレクトリに保存します。
- バケットからオブジェクトをダウンロードし、トラブルシューティングレポートを生成する

```
./ossutil probe --download --bucketname bucket-name [--object=object_name] [--addr=domain_name] [file_name]
```

- `--bucketname`: ダウンロードするオブジェクトを含むバケットの名前を指定します。
- `--object=`: オプション。このパラメーターには、ダウンロードしたオブジェクトを保存するファイルパスを指定します。例: `path/myphoto.jpg`。 `--object=` が指定されていない場合、ossutil は一時オブジェクトを生成し、`bucket-name` パラメーターで指定されたバケットにアップロードした後でダウンロードします。このオブジェクトがダウンロードされた後、ossutil はこのオブジェクトをローカルデバイスとバケットから削除します。
- オブジェクトをアップロードしてトラブルシューティングレポートを生成する

```
./ossutil probe --upload [file_name] --bucketname bucket-name [--object=object_name] [--addr=domain_name] [--upmode]
```

- `file_name`: オプション。このパラメーターには、`bucket-name` パラメーターで指定されたバケットにアップロードするオブジェクトの名前を指定します。`file_name` が指定されていない場合、ossutil は一時オブジェクトを生成し、指定されたバケットにアップロードします。probe の完了後、この一時オブジェクトは削除されます。

- `--object=`: オプション。このパラメータには、オブジェクトまたはディレクトリの名前を指定します。例: `path/myphoto.jpg`。 `--object=` が指定されていない場合、ossutil はアップロードされたオブジェクトの名前を生成します。probe の完了後、このオブジェクトは削除されます。
- `--upmode`: オプション。このパラメータには、アップロード方法を指定します。デフォルト値: `normal`。有効値:
 - `normal`
 - `append`
 - `multipart`

例

- オブジェクトの URL を使用してバケットからローカルデバイスにオブジェクトをダウンロードし、トラブルシューティングレポートを生成する

```
./ossutil probe --download --url https://bucket1.oss-cn-beijing.aliyuncs.com/myphoto.jpg --addr=www.aliyun.com /file/myphoto.jpg
```

- バケットからオブジェクトをダウンロードし、トラブルシューティングレポートを生成する

```
./ossutil probe --download --bucketname bucket1 --object=myphoto.jpg --addr=www.aliyun.com &nbsp;sp;/file/myphoto.jpg
```

- アップロード結果を確認し、トラブルシューティングレポートを生成する

```
./ossutil probe --upload /file/myphoto.jpg --bucketname bucket1 --object=myphoto.jpg &nbsp;--upmode normal
```

トラブルシューティングレポートの表示

probe コマンドの実行後、各タスクの実行手順と、アップロードまたはダウンロードの全体的な結果を表示できます。

- 手順の後に `x` が表示されている場合、操作は失敗しています。
- 手順の後に `x` が表示されていない場合、操作は成功しています。アップロードまたはダウンロードが成功した場合、オブジェクトのサイズと、オブジェクトのアップロード時刻またはダウンロード時刻が表示されます。アップロードまたはダウンロードが失敗した場合、失敗の原因やトラブルシューティングのアドバイスが表示されます。

❓ 説明 エラーの数が少ない場合、トラブルシューティングのアドバイスが生成されないことがあります。この場合、OSS エラーコードの説明に従って、エラーコードに応じて問題をトラブルシューティングできます。

probe コマンドの実行後、名前の先頭が `logOssProbe` で始まるオブジェクトが現在のディレクトリに生成されます。このオブジェクトには、問題をトラブルシューティングするために実行したコマンドの詳細が含まれています。

一般的なオプション

次の表に、probe コマンドに追加できるオプションを示します。

オプション	説明
--url	オブジェクトのネットワークアドレスを指定します。ossutilはこのアドレスを使用してオブジェクトをダウンロードします。
--bucketname	OSS のバケットの名前を指定します。
--object	OSS のオブジェクトの名前を指定します。
--addr	ping するドメイン名を指定します。デフォルト値: <code>www.aliyun.com</code> 。
--upmode	アップロード方法を指定します。デフォルト値: normal。有効値: • normal • append • multipart
--upload	オブジェクトを OSS にアップロードします。
--download	OSS からオブジェクトをダウンロードします。
--loglevel	ログレベルを指定します。デフォルト値は null で、ログファイルが生成されないことを示します。有効値: • info: プロンプトログを生成します。 • debug: 対応する HTTP リクエストとレスポンス情報を含む詳細なログを生成します。
--proxy-host	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 がサポートされています。URL の例: <code>http://120.79.**.*:3128</code> や <code>socks5://120.79.**.*:1080</code> など。
--proxy-user	プロキシサーバーのユーザー名を指定します。デフォルト値は null です。
--proxy-pwd	プロキシサーバーのパスワードを指定します。デフォルト値は null です。

② 説明 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.3.24. read-symlink

read-symlink コマンドは、シンボリックリンクオブジェクトの内容を読み取る場合に使用します。

- ② 説明
- シンボリックリンクの詳細は、「[シンボリックリンクの設定](#)」をご参照ください。
 - シンボリックリンクを作成するコマンドの詳細は、「[create-symlink](#)」をご参照ください。

コマンド構文

```
./ossutil read-symlink oss://bucket/object [--encoding-type url] [--payer requester] [-c file]
```

この操作には、シンボリックリンクオブジェクトの読み取り権限が必要です。出力内の X-Oss-Symlink-Target は、シンボリックリンクが指すオブジェクトを示します。管理するオブジェクトがシンボリックリンクオブジェクトでない場合、NotSymlink が返されます。

例

object1 のシンボリックリンク情報を取得します。

```
./ossutil read-symlink oss://bucket1/object1
Etag : 455E20DBFF1D588B67D092C46B16DB6
Last-Modified      : 2017-04-17 14:49:42 +0800 CST
X-Oss-Symlink-Target : a
```

一般的なオプション

次の表に、read-symlink コマンドに追加できるオプションを示します。

オプション	説明
--encoding-type	オブジェクト名のエンコードタイプを指定します。このオプションを指定する場合、値は url でなければなりません。このオプションが指定されていない場合、オブジェクト名はエンコードされません。バケット名は URL エンコードできません。
--loglevel	ログレベルを指定します。デフォルト値は null で、ログファイルが生成されないことを示します。有効値： - info: プロンプトログを生成します。 - debug: 対応する HTTP リクエストとレスポンス情報を含む詳細なログを生成します。
--retry-times	操作が失敗した場合に操作を再試行する回数を指定します。有効値: 1~500。デフォルト値: 10。
--payer	リクエストの支払者を指定します。pay-by-requester モードを有効にするには、このオプションを requester に設定します。
--proxy-host	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 がサポートされています。URL の例: http://120.79. **. **:3128、socks5://120.79. **. **:1080 など。
--proxy-user	プロキシサーバーのユーザー名を指定します。デフォルト値は null です。
--proxy-pwd	プロキシサーバーのパスワードを指定します。デフォルト値は null です。

② 説明 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.3.25. set-meta

set-meta コマンドは、アップロードされたオブジェクトのオブジェクトメタデータを設定する場合に使用します。

コマンド構文

```
./ossutil set-meta oss://bucket[/prefix] [header:value#header:value...] [--update] [--delete] [-r] [-f] [-c file]
```

例

- 特定のオブジェクトのメタデータを設定する

```
./ossutil set-meta oss://bucket1/path/object x-oss-object-acl:private
```

--update オプションと --delete オプションを指定しない場合、既存のメタデータは、指定したメタデータに置き換えられます。 [header:value#header:value...] オプションを指定しない場合、削除できないヘッダー (プレフィックスが X-Oss-meta 以外のヘッダーなど) の値は変更されません。他のメタデータは削除されます。

❓ 説明 ヘッダーは大文字と小文字が区別されませんが、値は大文字と小文字が区別されます。オブジェクトに設定できるヘッダーのリストは次のとおりです。

```
Headers:
Expires(time.RFC3339:2006-01-02T15:04:05Z07:00)
X-Oss-Object-Acl
Origin
X-Oss-Storage-Class
Content-Encoding
Cache-Control
Content-Disposition
Accept-Encoding
X-Oss-Server-Side-Encryption
Content-Type
and headers with the X-Oss-Meta- prefix.
```

詳細は、「[Object Meta の管理](#)」をご参照ください。

- 特定のプレフィックスで始まるオブジェクトのメタデータを設定する

```
./ossutil set-meta oss://bucket1/path/ Cache-Control:no-cache#x-oss-object-acl:private -r
```

-r オプションを指定すると、指定したプレフィックスで始まる複数のオブジェクトのメタデータが設定されます。オブジェクトの操作中にエラーが発生すると、オブジェクトのエラー情報がレポートオブジェクトに記録され、他のオブジェクトの操作が続行されます。処理されたオブジェクトの情報は、レポートオブジェクトに記録されません。複数のメタデータ項目をシャープ記号 (#) で区切ります。

- 特定のオブジェクトのメタデータを更新する

```
./ossutil set-meta oss://bucket1/path/object x-oss-object-acl:private --update
```

このコマンドでは、入力値として指定したオブジェクトの指定したヘッダーのみが更新されます。ヘッダー値は null にできます。指定したオブジェクトの他のメタデータは変更されません。--update (短縮形 -u) オプションと --delete オプションは同時に指定できません。

- 特定のオブジェクトのメタデータを削除する

```
./ossutil set-meta oss://bucket1/obj1 X-Oss-Meta-delete --delete
```

--delete オプションを指定した場合、指定したオブジェクトヘッダーが削除され、値は null になります。オブジェクトの他のメタデータは変更されません。このオプションは、削除できないヘッダー (プレフィックスが X-Oss-Meta 以外のヘッダーなど) には無効です。--update オプションと --delete オプションは同時に指定できません。

- 特定の条件を満たす複数のオブジェクトのメタデータを設定する

メタデータを設定するとき、--include/--exclude コマンドを使用して、指定した条件を満たすオブジェクトを選択できます。詳細は、「[cp](#)」をご参照ください。

- 拡張子が *jpg* のオブジェクトを低頻度アクセス (IA) ストレージクラスに設定する

```
./ossutil64 set-meta oss://my-bucket/path X-Oss-Storage-Class:IA --include "*.jpg" -u -r
```

- 名前に *abc* を含むオブジェクトと、形式が *jpg* または *txt* 以外のオブジェクトを標準ストレージクラスに設定する

```
./ossutil set-meta oss://my-bucket/path X-Oss-Storage-Class:Standard --include "abc*" --exclude "*.jpg" --exclude "*.txt" -u -r
```

- バージョン管理が有効になっているバケットにあるオブジェクトの特定バージョンのメタデータを設定する

```
./ossutil set-meta oss://bucket1/test.jpg X-Oss-Storage-Class:Standard --version-id CAEQARiBgID8rumR2hYilGUyOTAYZGY2MzU5MjQ5ZjIhYzQzZjNIYTAYZDE3MDRk
```

--version-id オプションを使用するには、[ls --all-versions](#) コマンドを実行して、オブジェクトの使用可能な全バージョンを取得する必要があります。

② 説明 --version-id オプションは、バージョン管理が有効になっているバケット内のオブジェクトにのみ使用できます。バケットのバージョン管理を有効にするコマンドの詳細は、「[bucket-versioning](#)」をご参照ください。

一般的なオプション

次の表に、set-meta コマンドに追加できるオプションを示します。

オプション	説明
-u, --update	特定のオブジェクトのメタデータを更新します。
--delete	特定のオブジェクトのメタデータを削除します。

オプション	説明
-r、--recursive	バケット内のオブジェクトに対して操作を再帰的に実行します。このオプションが指定されている場合、このオプションをサポートするコマンドは、指定された条件を満たすバケット内のすべてのオブジェクトに対して操作を実行します。このオプションが指定されていない場合、コマンドは指定された 1 つのオブジェクトに対してのみ操作を実行します。
-f、--force	ユーザーに確認を求めずに操作を強制します。
--include	*.jpg など、特定の文字列に一致するオブジェクトを含めます。
--exclude	*.txt など、特定の文字列に一致するオブジェクトを除外します。
--encoding-type	オブジェクト名のエンコードタイプを指定します。このオプションを指定する場合、値は url でなければなりません。このオプションが指定されていない場合、オブジェクト名はエンコードされません。バケット名は URL エンコードできません。
-j、--jobs	複数のオブジェクトが操作される場合の同時タスク数を指定します。有効値: 1~10000。デフォルト値: 3。
-L、--language	ossutil の使用言語を指定します。有効値: CH と EN。デフォルト値: CH。このオプションを CH に設定するには、システムで UTF-8 エンコードがサポートされていることを確認してください。
--output-dir	出力オブジェクトのディレクトリを指定します。出力オブジェクトには、cp コマンドで複数のオブジェクトをコピーしたときのエラーが原因で生成されるレポートオブジェクトが含まれます。レポートオブジェクトの詳細は、cp コマンドのヘルプ情報をご参照ください。デフォルト値は、現在のディレクトリの <i>ossutil_output</i> ディレクトリです。
--loglevel	ログレベルを指定します。デフォルト値は null で、ログファイルが生成されないことを示します。有効値: - info: プロンプトログを生成します。 - debug: 対応する HTTP リクエストとレスポンス情報を含む詳細なログを生成します。
--version-id	バージョン管理が有効になっているバケットにあるオブジェクトのバージョン ID を指定します。
--retry-times	操作が失敗した場合に操作を再試行する回数を指定します。有効値: 1~500。デフォルト値: 10。
--proxy-host	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 がサポートされています。URL の例: http://120.79.**.*:3128、socks5://120.79.**.*:1080 など。
--proxy-user	プロキシサーバーのユーザー名を指定します。デフォルト値は null です。
--proxy-pwd	プロキシサーバーのパスワードを指定します。デフォルト値は null です。

❓ 説明 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.3.26. stat

stat コマンドは、バケットまたはオブジェクトの説明を取得する場合に使用します。たとえば、stat コマンドを実行して、set-meta コマンドで設定されたオブジェクトメタデータを確認できます。

コマンド構文

```
./ossutil stat oss://bucket[/object] [--encoding-type url] [--payer requester] [-c file]
```

例

- バケットの情報を取得する

```
./ossutil stat oss://bucket1
```

- 特定のオブジェクトの情報を取得する

```
./ossutil stat oss://bucket1/object
```

- 名前に特殊文字が含まれるオブジェクトの情報を取得する

ossutil は、オブジェクト名の URL エンコードのみをサポートします。オブジェクト名に特殊文字が含まれている場合、コマンドでオブジェクト名を使用する前にこれらの特殊文字をエンコードできます。

```
./ossutil stat oss://bucket1/%E7%A4%BA%E4%BE%8B.txt --encoding-type url
```

 **注意** バケット名は URL エンコードできません。

- バージョン管理が有効になっているバケットにあるオブジェクトの特定バージョンを表示する

```
./ossutil stat oss://bucket1/test.jpg --version-id CAEQARiBgID8rumR2hYiIGUyOTAYZGY2MzU5MjQ5Zj  
lhYzQzZjNIYTAyZDE3MDRk
```

`--version-id` オプションを使用するには、`ls --all-versions` コマンドを実行して、使用可能な全オブジェクトバージョンを取得する必要があります。

 **説明** `--version-id` オプションは、バージョン管理が有効になっているバケット内のオブジェクトにのみ使用できます。バケットのバージョン管理を有効にするコマンドの詳細は、「[bucket-versioning](#)」をご参照ください。

一般的なオプション

次の表に、stat コマンドに追加できるオプションを示します。

オプション	説明
-------	----

オプション	説明
--encoding-type	オブジェクト名のエンコードタイプを指定します。このオプションを指定する場合、値は url でなければなりません。このオプションが指定されていない場合、オブジェクト名はエンコードされません。バケット名は URL エンコードできません。
--loglevel	ログレベルを指定します。デフォルト値は null で、ログファイルが生成されないことを示します。有効値: • info: プロンプトログを生成します。 • debug: 対応する HTTP リクエストとレスポンス情報を含む詳細なログを生成します。
--retry-times	操作が失敗した場合に操作を再試行する回数を指定します。有効値: 1 ~ 500。デフォルト値: 10。
--version-id	バージョン管理が有効になっているバケットにあるオブジェクトのバージョン ID を指定します。
--payer	リクエストの支払者を指定します。pay-by-requester モードを有効にするには、このオプションを requester に設定します。
--proxy-host	プロキシサーバーの URL を指定します。HTTP、HTTPS、SOCKS5 がサポートされています。URL の例: http://120.79. **. **:3128、socks5://120.79. **. **:1080 など。
--proxy-user	プロキシサーバーのユーザー名を指定します。デフォルト値は null です。
--proxy-pwd	プロキシサーバーのパスワードを指定します。デフォルト値は null です。

❓ 説明 一般的なオプションの詳細は、「[サポートされているすべてのオプションの表示](#)」をご参照ください。

3.4. サポートされているすべてのオプションの表示

-h オプションを使用すると、ossutil でサポートされているすべてのオプションを表示できます。

```

$./ossutil -h
Usage of ossutil:
Options:
-s --short-format Used to display the short format. If this option is not specified, the long format is displayed by default.
--snapshot-path= Used to accelerate incremental uploading of files in batches in some scenarios. (File downloading and copying do not support this option currently.) This option is used when ossutil uses the cp command to upload files. Ossutil takes a snapshot of file uploads and stores it in the specified directory. It will read the snapshot from the specified directory for incremental upload when this option is used the next time. The specified snapshot directory must be a writable directory in the local file system. If the directory does not exist, ossutil creates a file to record the snapshot information. If the dir

```

ectory already exists, ossutil reads the snapshot information in the directory, performs incremental uploading accordingly, and updates the snapshot information. (Ossutil only uploads files that fail to be uploaded last time and have been locally modified.) NOTE: By using this option, the local lastModifiedTime of files that have been successfully uploaded is recorded and compared with that of files to be uploaded next time to determine whether to skip uploading of same files. When using this option, ensure that the corresponding objects on the OSS are not modified during the two uploading periods. In other scenarios than this one, use the --update option to incrementally upload files in batches. In addition, ossutil does not actively delete the snapshot information under snapshot-path. To avoid too much snapshot information, clear snapshot-path when confirming that the snapshot information is useless.

-j --jobs= Specifies the number of concurrent tasks when multiple files are operated. The value ranges from 1 to 10000, and the default value is 5.

-v --version Used to display ossutil version (1.0.0.Beta2) and exit.

--output-dir= Specifies the directory in which output files are located. The output files include the report file generated when an error occurs for copying files in batches using the cp command. (For details about the report file, refer to the cp command help.) The default value is the ossutil_output sub-directory in the current directory.

--parallel= Specifies the number of concurrent tasks operated in a single file. The value ranges from 1 to 10000. The default value is determined by ossutil based on the operation type and file size.

-L --language= Specifies the language of ossutil. The value can be CH or EN, and the default value is CH.

-t --sts-token= Specifies the STSToken used to access the OSS (the option value will overwrite the corresponding settings in the configuration file). It is optional.

-m --multipart Indicates that the operation objects are the incomplete Multipart events in the bucket, rather than the default objects.

-b --bucket Used to operate a bucket, confirming that an operation is for the bucket.

--delete Used to delete an operation.

-e --endpoint= Specifies the basic endpoint configuration of ossutil (the option value will overwrite the corresponding settings in the configuration file). It must be a second-level domain.

-k --access-key-secret= Specifies the AccessKeySecret used to access the OSS (the option value will overwrite the corresponding settings in the configuration file).

--bigfile-threshold= Specifies the threshold for enabling the resumable data transfer for large files. The value ranges from 0 B to 9223372036854775807 B, and the default value is 100 MB.

--retry-times= Specifies the number of retries when an error occurs. The value ranges from 1 to 500, and the default value is 3.

-a --all-type Indicates that the operation objects are the objects and incomplete Multipart events in the bucket.

-r --recursive Indicates a recursive operation. If this option is specified, commands supporting this option will operate on all objects meeting the criteria in the bucket. Otherwise, the commands operate only on a single object specified in the URL.

-f --force Indicates a forcible operation without asking.

-u --update Indicates an update operation.

`-c --config-file=` Specifies the configuration file path of ossutil. Ossutil reads configuration from the configuration file during startup and writes configuration to the file using the config command.

`-i --access-key-id=` Specifies the AccessKeyID used to access the OSS (the option value will overwrite the corresponding settings in the configuration file).

`--acl=` Used to configure the ACL.

`-d --directory` Used to return files and sub-directories in the current directory, rather than recursively displaying all objects in all sub-directories.

`--checkpoint-dir=` Specifies the checkpoint directory path (the default value is `.ossutil_checkpoint`). If a resumable data transfer fails, ossutil automatically creates this directory and records the checkpoint information in the directory. If a resumable data transfer succeeds, ossutil deletes this directory. If this option is specified, ensure that the specified directory can be deleted.

`--type=` Specifies the calculation type. The value can be `crc64` or `md5`, and the default value is `crc64`.

`-h --help` Show usage message

ossutil のすべてのコマンドは、上記のオプションの一部をサポートしています。各コマンドでサポートされているオプションを確認するには、`ossutil help` コマンドを使用します。

3.5. バケット関連のコマンド

このトピックでは、Alibaba Cloud OSS のオープンソースツールであるossutilを使用してバケット関連のコマンドを実行する方法について説明します。特に、バケットの作成、削除、一覧表示、ACL の変更方法について説明します。また、オブジェクトや未完了のマルチパートアップロードタスクなどのバケット関連アイテムの管理、CORS (クロスオリジンリソース共有)、ログ出力、Anti-leech ルールの管理、さらに OSS ネットワークのトラブルシューティングについても説明します。

② 説明

- バケット関連のコマンドを実行する前に、まず ossutil を最新バージョンにアップグレードし、`config` コマンドを実行してアクセスキーを設定してください。詳細については、「[クイックスタート](#)」をご参照ください。
- 現在 ossutil でサポートされていないバケット管理機能については説明されていません。それらの機能が必要な場合は、`osscli` ツールを使用してください。詳細については、「[クイックインストール](#)」をご参照ください。

バケットの作成

- バケットの作成

```
./ossutil mb oss://bucketname [--acl=ACL][--storage-class sc][-c file]
```

バケットが作成されると、ossutil はバケットの作成に要した時間を出力して終了します。バケットの作成に失敗した場合、ossutil は対応するエラー情報を出力します。

② 説明 `mb` コマンドの使用方法を参照するには、`ossutil help mb` コマンドを実行します。

- バケットの作成と ACL の設定

`--acl` パラメーターでバケットの ACL を設定できます。デフォルトでは、*private* に設定されています。利用可能な ACL は次のとおりです。

- *private*: 匿名ユーザーには、バケット内オブジェクトの読み取りおよび書き込みが許可されません。アクセスには署名が必要となります。
- *public-read*: 匿名ユーザーに、バケット内オブジェクトの読み取りが許可されます。
- *public-read-write*: 匿名ユーザーに、バケット内オブジェクトの読み取りおよび書き込みが許可されます。

❓ 説明 アクセス制御の詳細については、「[ACL](#)」をご参照ください。

たとえば、バケットを作成し、ACL を *public-read-write* に設定するには、次のコマンドを実行します。

```
./ossutil mb oss://bucket --acl=public-read-write
```

● バケットの作成とストレージクラスの設定

`--storage-class` を使用して、バケットのストレージクラスを設定できます。デフォルトでは、*Standard* に設定されています。使用可能なストレージクラスは次のとおりです。

- *Standard* (標準)
- *IA* (低頻度アクセス)
- *Archive* (アーカイブ)

❓ 説明 ストレージクラスの詳細については、「[Overview](#)」をご参照ください。

たとえば、バケットを作成し、ストレージクラスを *Infrequent Access* に設定するには、次のコマンドを実行します。

```
./ossutil mb oss://bucket --storage-class IA
```

バケットの ACL の変更

バケットの ACL を変更するには、`set-acl` コマンドを実行します。このコマンドには、`-b` パラメーターを指定する必要があります。

たとえば、バケットの ACL を *private* に変更するには、次のコマンドを実行します。

```
./ossutil set-acl oss://bucket1 private -b
```

❓ 説明 `set-acl` コマンドの使用方法を参照するには、`ossutil help set-acl` コマンドを実行します。

バケットの削除

● 空バケットの削除

```
./ossutil rm oss://bucket -b
```

注意

- バケットを削除する場合、`-b` パラメーターを指定する必要があります。
- 削除したバケットと同じ名前のバケットが、別のユーザーによって新たに作成される可能性があります。この場合、バケットの所有権は新たにバケットを作成したユーザーにあります。
- For information about how to use the `rm` コマンドの使用方法を参照するには、`ossutil help rm` コマンドを実行します。

● バケットのクリアと削除

バケットにオブジェクトまたはマルチパートデータが残っている場合、バケットを削除する前に、残っているオブジェクトまたはマルチパートデータを削除する必要があります。

```
./ossutil rm oss://bucket -bar
```

 **警告** このコマンドを実行すると、バケット内のすべてのデータが削除されます。

バケットの一覧表示

● すべてのバケットの一覧表示

次の2つのコマンドのいずれかを実行して、すべてのバケットを一覧表示できます。

```
./ossutil ls
```

```
./ossutil ls oss://
```

 **説明** `-s` パラメーターを使用すると、簡易な形式でバケットが一覧表示されます。`ls` コマンドの使用方法を参照するには、`ossutil help ls` コマンドを実行します。

例:

```
./ossutil ls
CreationTime Region StorageClass BucketName
2016-10-2116:18:37 +0800 CST oss-cn-hangzhou Archive oss://go-sdk-test-bucket-xyz-for-object
2016-12-0115:06:21 +0800 CST oss-cn-hangzhou Standard oss://ossutil-test
2016-07-1817:54:49 +0800 CST oss-cn-hangzhou Standard oss://ossutilconfig
2016-07-2010:36:24 +0800 CST oss-cn-hangzhou IA oss://ossutilupdate
2016-11-1413:08:36 +0800 CST oss-cn-hangzhou IA oss://yyyyy
2016-08-2509:06:10 +0800 CST oss-cn-hangzhou Archive oss://ztzt
2016-11-2121:18:39 +0800 CST oss-cn-hangzhou Archive oss://ztztzt
Bucket Number is:7
0.252174(s) elapsed
```

- バケットの一覧表示 (ページ単位)

```
./ossutil ls oss:// --limited-num=${num} --marker=${bucketname}
```

多数のバケットが作成されている場合、`--limited-num` および `--marker` パラメーターを使用して、バケットをページ単位で一覧表示できます。

- `--limited-num` : 1ページあたりに表示されるバケット数を指定します。
- `--marker` : 一覧の先頭に表示されるバケットを指定します。バケットは名前のアルファベット順にソートされ、ページごとに表示されます。通常は、前回表示された一覧の末尾に表示されたバケットの名前を指定します。

たとえば、次のコマンドを実行すると、最初の2つのバケットがページごとに一覧表示されます。

```
./ossutil ls oss:// --limited-num=1 -s
oss://bucket1
Bucket Number is:1
0.303869(s) elapsed
$ ./ossutil ls oss:// --limited-num=1 -s --marker=bucket1
oss://bucket2
Bucket Number is:1
0.257636(s) elapsed
```

オブジェクトの一覧表示

- バケット内の全オブジェクトの一覧表示

```
./ossutil ls oss://bucketname
```

例:


```

$./ossutil ls oss://ossutil-test
LastModifiedTime Size(B) StorageClass ETAG ObjectName
2016-12-0115:06:37 +0800 CST 10363812 Standard 61DE142E5AFF9A6748707D4A77BFBCFB oss://ossut
il-test/a1
2016-12-0115:06:42 +0800 CST 10363812 Standard 61DE142E5AFF9A6748707D4A77BFBCFB oss://ossut
il-test/a2
2016-12-0115:06:45 +0800 CST 10363812 Standard 61DE142E5AFF9A6748707D4A77BFBCFB oss://ossut
il-test/a3
Object Number is:3
0.007379(s) elapsed

```

- すべてのオブジェクトおよび未完了のマルチパートアップロードタスクの一覧表示

```
./ossutil ls oss://bucket -a
```

例:

```

./ossutil ls oss://bucket1 -a
LastModifiedTime Size(B) StorageClass ETAG ObjectName
2015-06-0514:06:29 +0000 CST 201933 Standard 7E2F4A7F1AC9D2F0996E8332D5EA5B41 oss://bucket1
/dir1/obj11
2015-06-0514:36:21 +0000 CST 201933 Standard 6185CA2E8EB8510A61B3A845EAFE4174 oss://bucket1
/obj1
2016-04-0814:50:47 +0000 CST 6476984 Standard 4F16FDAE7AC404CEC8B727FCC67779D6 oss://bucket
1/sample.txt
Object Number is:3
InitiatedTime UploadID ObjectName
2017-01-1303:45:26 +0000 CST 15754AF7980C4DFB8193F190837520BB oss://bucket1/obj1
2017-01-1303:43:13 +0000 CST 2A1F9B4A95E341BD9285CC42BB950EE0 oss://bucket1/obj1
2017-01-1303:45:25 +0000 CST 3998971ACAF94AD9AC48EAC1988BE863 oss://bucket1/obj2
2017-01-2011:16:21 +0800 CST A20157A7B2FEC4670626DAE0F4C0073C oss://bucket1/tobj
Uploadid Number is:4
0.191289(s) elapsed

```

- すべてのオブジェクトの一覧表示 (ページ単位)

```
./ossutil ls oss://bucket --limited-num=${num} --marker=${obj}
```

--limited-num および --marker パラメーターを使用して、オブジェクトをページ単位で一覧表示できます。「バケットの一覧表示」でバケットをページ単位で一覧表示した方法と同様です。

```
./ossutil ls oss://ossutil-test --limited-num=1
LastModifiedTime Size(B) StorageClass ETAG ObjectName
2016-12-0115:06:37 +0800 CST 10363812 Standard 61DE142E5AFF9A6748707D4A77BFBCFB oss://ossut
il-test/a1
Object Number is:1
0.007379(s) elapsed
$./ossutil ls oss://ossutil-test --limited-num=1 --marker=a1
LastModifiedTime Size(B) StorageClass ETAG ObjectName
2016-12-0115:06:42 +0800 CST 10363812 Standard 61DE142E5AFF9A6748707D4A77BFBCFB oss://ossut
il-test/a2
Object Number is:1
0.008392(s) elapsed
```

- オブジェクトの一覧表示 (簡易表示)

```
./ossutil ls oss://bucket -s
```

例:

```
./ossutil ls oss://ossutil-test
oss://ossutil-test/a1
oss://ossutil-test/a2
oss://ossutil-test/a3
Object Number is:3
0.007379(s) elapsed
```

- オブジェクトの一覧表示 (ディレクトリ構造)

```
./ossutil ls oss://bucket -d
```

現在のディレクトリにあるサブディレクトリ内の全オブジェクトを繰り返し表示したくない場合は、`-d` パラメーターを指定してオブジェクトとサブディレクトリを一覧表示します。

例:

```
./ossutil ls oss://bucket1 -s -d
oss://bucket1/obj1
oss://bucket1/sample.txt
oss://bucket1/dir1/
Object and Directory Number is:3
0.119884(s) elapsed
```

- ドメイン名がバインドされているバケット内のすべてのオブジェクトの一覧表示

詳細については、[オブジェクト関連のコマンド](#) をご参照ください。

未完了のマルチパートアップロードタスクと関連情報の一覧表示

- 未完了のマルチパートアップロードタスクの一覧表示

```
./ossutil ls oss://bucket -m
```

-m パラメーターを使用して、バケット内の未完了のマルチパートアップロードタスクを一覧表示できます。

例:

```
./ossutil ls oss://bucket1 -m
InitiatedTime UploadID ObjectName
2017-01-1303:45:26 +0000 CST 15754AF7980C4DFB8193F190837520BB oss://bucket1/obj1
2017-01-1303:45:25 +0000 CST 3998971ACAF94AD9AC48EAC1988BE863 oss://bucket1/obj2
2017-01-2011:16:21 +0800 CST A20157A7B2FEC4670626DAE0F4C0073C oss://bucket1/tobj
UploadID Number is:30.009424(s) elapsed
```

- アップロード待ちパーツの一覧表示 (全オブジェクト対象)

```
./ossutil getallpartsize oss://bucket
```

- アップロード待ちパーツの一覧表示 (特定オブジェクト対象)

```
./ossutil listpart oss://bucket/object uploadid
```

The uploadid パラメーターで、アップロード待ちパーツがあるオブジェクトのアップロードタスク IDを指定します。

❓ 説明 マルチパートアップロードの詳細については、「[マルチパート関連コマンド](#)」を参照してください。

バケットのタグ付け設定

bucket-tagging コマンドの実行時に method パラメーターの値を *put*、*get*、または delete に設定して、バケットにタグを追加したり、タグを変更、照会、または削除したりできます。バケットのタグ付け設定の詳細については、[Bucket tagging](#) をご参照ください。

② 説明

- バケットにタグを追加できるのは、バケット所有者と承認された RAM ユーザーに限られます。その他のユーザーに対しては、403 Forbidden エラーとエラーコード `AccessDenied` が返されます。
- タグのキーと値は、番号記号 (#) で区切る必要があります。
- バケットには最大 20 個のタグ (キーと値のペア) を追加できます。タグはスペースで区切る必要があります。
- タグには、文字、数字、スペース、および次の記号を含めることができます。
+ - = . _ : /
- キーの最大長は 64 バイトです。タグのキーを空に設定することはできません。また、タグを ` ` `http://` , `https://` , or `Aliyun` で開始することもできません。
- 値の最大長は 128 バイトです。タグの値は空に設定できます。
- タグのキーと値は UTF-8 でエンコードする必要があります。
- `bucket-tagging` コマンドを実行する前に、` ` `ossutil help bucket-tagging` コマンドを実行して詳細な情報を確認することを推奨します。

- バケットにタグを追加したり、バケットに追加されたタグを変更したりするには、次のコマンドを実行します。

```
./ossutil bucket-tagging --method put oss://bucket tagkey1#tagvalue1 tagkey2#tagvalue2
```

バケットにタグが追加されていない場合はバケットに指定されたタグを追加し、すでに追加されている場合は既存のタグを上書きします。

- バケットに追加されたタグを照会するには、次のコマンドを実行します。

```
./ossutil bucket-tagging --method get oss://bucket
```

- バケットに追加されたタグを削除するには、次のコマンドを実行します。

```
./ossutil bucket-tagging --method delete oss://bucket
```

バケットの暗号化設定

`bucket-encryption` コマンドの実行時に ` ` ` ` `method` / パラメーターの値を ` ` `put`、`get`、または ` ` `delete` に設定して、バケットに暗号化設定を追加したり、暗号化設定を変更、照会、または削除したりできます。バケットの暗号化設定の詳細については、[Server-side encryption](#) をご参照ください。

- バケットの暗号化設定を追加するには、次のコマンドを実行します。

```
./ossutil bucket-encryption --method put oss://bucket --sse-algorithm algorithmName [--kms-masterkey-id keyid]
```

- `--sse-algorithm` : バケットの暗号化方法 (KMS または AES) を指定します。

- `--kms-masterkey-id` : 暗号化に使用する CMK ID を指定します。 `--sse-algorithm` が KMS の場合、必要に応じて指定します。 This parameter cannot be set if the value of `--sse-algorithm` が AES256 の場合は指定できません。

上記のコマンドは、バケットに暗号化方法が設定されていない場合はバケットの暗号化方法を設定し、設定されている場合はバケットの暗号化設定を変更します。

例:

```
./ossutil bucket-encryption --method put oss://bucket --sse-algorithm KMS --kms-masterkey-id 9468da86-3509-4f8d-a61e-6eab1eac****
```

- バケットの ACL を取得するには、次のコードを実行します。

```
./ossutil bucket-encryption --method get oss://bucket
```

- バケットの暗号化設定を削除するには、次のコードを実行します。

```
./ossutil bucket-encryption --method delete oss://bucket
```

CORS (クロスオリジンリソース共有) ルールの管理

`cors` コマンドの実行時に `method` パラメーターの値を `put`、`get`、または `delete` に設定して、バケットに CORS ルールを追加したり、CORS ルールを変更、照会、または削除したりできます。詳細については、「[クロスオリジンアクセス設定](#)」をご参照ください。

❓ 説明 `cors` コマンドのの使用方法を参照するには、`ossutil help cors` コマンドを実行します。

- バケットの CORS ルールの追加または変更

```
./ossutil cors --method put oss://bucket local_xml_file
```

ossutil は、構成ファイル `local_xml_file` から CORS ルールを取得します。バケットに CORS ルールが設定されていない場合、ossutil は設定ファイルから取得した対応する CORS ルールをバケットに追加します。バケットに CORS ルールが設定されている場合、ossutil は構成ファイルから取得した CORS ルールに変更します。

② 説明 XML 形式の構成ファイル *local_xml_file* の内容は次のとおりです。

```
<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration>
<CORSRule>
<AllowedOrigin>www.aliyun.com</AllowedOrigin>
<AllowedMethod>PUT</AllowedMethod>
<MaxAgeSeconds>10000</MaxAgeSeconds>
</CORSRule>
</CORSConfiguration>
```

- バケットの CORS ルールを取得する

```
./ossutil cors --method get oss://bucket  [local_xml_file]
```

`local_xml_file` パラメーターが設定されている場合、ossutil は取得した CORS ルールを構成ファイル / `ocal_xml_file` に保存します。このパラメーターが null の場合、ossutil は取得した CORS ルールを画面に表示します。

- バケットの CORS ルールの削除

```
./ossutil cors --method delete oss://bucket
```

ログールの管理

logging コマンドの実行時に method パラメーターの値を *put*、*get*、または *delete* に設定して、バケットにログルールを追加したり、ログルールを変更、照会、または削除したりできます。詳細については、[ログの設定](#) をご参照ください。

② 説明 cors コマンドの使用方法を参照するには、`ossutil help logging` コマンドを実行します。

- バケットのログルールの追加または変更

```
./ossutil logging --method put oss://bucket  oss://target-bucket/[prefix]
```

ログ管理が有効化されていない場合は、このコマンドを実行して、*taarget-bucket* パラメーターで指定したバケットにバケットアクセスログをオブジェクトとして保存します。ログ管理が有効になっている場合は、このコマンドを実行して、バケットアクセスログを格納するためのディレクトリを変更できます。

prefix パラメーターで、バケットアクセスログを格納するためのディレクトリとプレフィックスを指定します。このパラメーターが設定されている場合、ossutil はバケットアクセスログを *target-bucket* パラメーターで指定したバケットの指定したディレクトリに保存します。このパラメーターが null の場合、ossutil はバケットアクセスログを *target-bucket* パラメーターで指定したバケットのルートディレクトリに保存します。ログオブジェクトの命名規則については、[ログの設定](#) をご参照ください。

- バケットのリージョンの取得

```
./ossutil logging --method get oss://bucket    [local_xml_file]
```

local_xml_file パラメーターが設定されている場合、ossutil は取得したログルールを構成ファイル *local_xml_file* に保存します。このパラメーターが null の場合、ossutil は取得したログルールを画面に表示します。

- バケットのログルールの削除

```
./ossutil logging --method delete oss://bucket
```

Anti-leech ルールの管理

referer コマンドの実行時に method パラメーターの値を *put*、*get*、または *delete* に設定して、バケットのAnti-leech ルールを追加したり、ルールを変更、照会、または削除したりできます。詳細については、[OSS Anti-leech 保護](#)をご参照ください。

❓ 説明 referer コマンドの使用方法を参照するには、ossutil help referer コマンドを実行します。

- バケットの Anti-leech ルールの追加または変更

```
./ossutil referer --method put oss://bucket referer-value [--disable-empty-referer]
```

バケットに Anti-leech ルールが設定されていない場合、ossutil は指定された Anti-leech ルールを追加します。すでに設定されている場合、ossutil は既存のルールを指定されたルールに変更します。

- referer-value : スペースで区切られたドメイン名の一覧が定義された Referer ホワイトリストの使用を開始し、設定します。ホワイトリストには、ワイルドカード文字 (*) と疑問符 (?) を含めることができます。ホワイトリストに定義されたドメインからの OSS アクセス要求のみが許可されます。
- --disable-empty-referer : 未設定のReferer フィールドの使用可否を指定します。referer コマンドで --disable-empty-referer パラメーターが使用される場合、Referer フィールドは未設定にできません。OSS へのアクセスは、HTTP または HTTPS ヘッダーに Referer フィールドが設定されたリクエストに限られます。--disable-empty-referer パラメーターが使用されていない場合、Referer フィールドの設定は不要です。

たとえば、Referer フィールドを未設定にできない Anti-leech ルールをバケットに設定するには、次のコマンドを実行します。

```
./ossutil referer --method put oss://ossutil-test www.test1.com www.test2.com --disable-empty-referer
```

- バケットの Anti-leech ルールの取得

```
./ossutil referer --method get oss://bucket [local_xml_file]
```

local_xml_file パラメーターが設定されている場合、ossutil は取得した Anti-leech ルールを構成ファイル *local_xml_file* に保存します。null に設定されている場合、ossutil は取得した Anti-leech ルールを画面に表示します。

- バケットの Anti-leech ルールの取得

```
./ossutil referer --method delete oss://bucket
```

OSS ネットワークのトラブルシューティング

probe コマンドを実行すると、ossutil はアップロードまたはダウンロードの失敗を引き起こしたと考えられる原因を表示します。OSS ネットワークの障害や、正しくない基本パラメーターの設定が原因となる場合もあります。

❓ 説明 probe コマンドの使用方法を参照するには、ossutil help probe コマンドを実行します。

- オブジェクトの URL を使用したバケットからのオブジェクトのダウンロード、およびトラブルシューティング レポートの出力

```
./ossutil probe --download --url http_url [--addr=domain_name] [file_name]
```

オブジェクトの URL を使用してバケットからオブジェクトをダウンロードした後、ネットワーク伝送品質をテストしてトラブルシューティングレポートを出力できます。

- `--url` : バケット内のオブジェクトの URL を指定します。
 - オブジェクトの ACL が *public-read* の場合、URL には署名が含まれません。(例: `https://bucketname.oss-cn-beijing.aliyuncs.com/myphoto.jpg`)
 - オブジェクトの ACL が *private* の場合、URL は引用符 (") で挟まれ、また署名を含んでいます。(例: `"https://bucketname.oss-cn-beijing.aliyuncs.com/myphoto.jpg?Expires=1552015472&OSSAccessKeyId=TMP.xxxxxxx5r9f1FV12y8_Qis6LUVmvoSCUSs7aboCCHtydQ0axN32Sn-UvyY3AAAwLAIUarYNLcO87AKMEcE503AxxxxxxoCFAQuRdZYyVFyqOW8QkGAN-bamUiQ&Signature=bla4llbMblldrl7rwckr%2FXXvTtxw%3D"`)

❓ 説明 オブジェクトの URL を取得する方法については、[OSSでオブジェクトのリンクアドレスを取得する方法](#) をご参照ください。

- `--addr=domain_name` : オブジェクトのダウンロード中に ping コマンドが送信されるドメインまたは IP アドレスを指定します。このパラメーターは省略可能です。指定されない場合、ossutil は別のドメインまたは IP アドレスを確認しません。
 - デフォルト値が指定されている場合、ossutil は ping コマンドを実行して、利用中の OSS ネットワークと `www.aliyun.com` との間の通信が正常に行われていることを確認します。
 - ドメイン名または IP アドレスが指定されている場合、ossutil は ping コマンドを実行して、利用中の OSS ネットワークと指定したドメインまたは IP アドレス間の通信が正常に行われていることを確認します。
 - `file_name` : ダウンロードしたオブジェクトを保存するディレクトリを指定します。このパラメーターは省略可能です。このパラメーターが指定されていない場合、ossutil はダウンロードしたオブジェクトを現在のディレクトリに保存し、オブジェクト名を決定します。オブジェクト名またはディレクトリ名が指定されている場合、ossutil は、ダウンロードしたオブジェクトを指定された名前で保存するか、または指定されたディレクトリに保存します。
- バケットからのオブジェクトのダウンロード、およびトラブルシューティングレポートの出力


```
./ossutil probe --download --bucketname bucket-name [--object=object_name]
[--addr=domain_name] [file_name]
```

- `--bucketname` : ダウンロードするオブジェクトが保存されているバケットの名前を指定します。
- `--object=` : ダウンロードしたオブジェクトの保存先ディレクトリを指定します。このパラメーターは省略可能です。このパラメーターが指定されていない場合、ossutil は一時オブジェクトを生成して `bucket-name` パラメーターで指定されたバケットにアップロードしてからオブジェクトをダウンロードします。オブジェクトのダウンロード後、一時オブジェクトは ossutil によってローカルコンピュータとバケットから削除されます。

- アップロード結果の確認とトラブルシューティングレポートの出力

```
./ossutil probe --upload [file_name] --bucketname bucket-name [--object=obj
ect_name] [--addr=domain_name] [--upmode]
```

- `file_name` : `bucket-name` パラメーターで指定したバケットにアップロードするオブジェクトの名前を指定します。 `file_name` パラメーターは省略可能です。このパラメーターが指定されていない場合、ossutil は一時オブジェクトを生成し、指定したバケットにアップロードします。検査の完了後、ossutil は一時オブジェクトを削除します。
- `--object=` : オブジェクトまたはディレクトリの名前 このパラメーターは省略可能です。たとえば、`path/myphoto.jpg` のように、オブジェクトのアップロード後の名前を指定します。このパラメーターが指定されていない場合、ossutil がアップロードされたオブジェクトの名前を生成します。検査の完了後、ossutil はこのオブジェクトを削除します。
- `--upmode` : アップロード方法を指定します。このパラメーターは省略可能です。デフォルト値は *normal* です。利用可能なアップロード方法は次のとおりです。
 - *normal*
 - *append*
 - *multipart*

- トラブルシューティングレポートの取得

`probe` コマンドを実行すると、タスク実行ステップおよびアップロードまたはダウンロードの全体的な結果が表示されます。

- × が表示されたステップは失敗しています。× が表示されていないステップは正常に完了しています。
- アップロードまたはダウンロードが正常に完了した場合、ossutil はオブジェクトのサイズ、アップロード時刻またはダウンロード時刻を出力します。アップロードまたはダウンロードが失敗した場合、ossutil は障害の原因またはトラブルシューティングに関するアドバイスを出力します。

❓ 説明 一部のエラーについては、アドバイスが出力されない場合があります。その場合、「例外処理」に記載されている手順に従い、エラーコードに基づいてトラブルシューティングを行うことができます。

`probe` コマンドを実行すると、ossutil は名前が `probe` で始まるオブジェクトを現在のディレクトリに作成します。このオブジェクトには、実行したコマンドの詳細情報が記録されています。

3.6. オブジェクト関連のコマンド

Ossutil では、ファイルのアップロード / ダウンロード / コピー、オブジェクトの ACL とメタの設定、およびオブジェクトのメタ情報の表示を行うことができます。

前述のコマンドを実行する前に、config コマンドを実行し、AccessKey ペアを設定します。

ファイルのアップロード / ダウンロード / コピー

cp コマンドを実行する前に、ossutil help cp を実行し、ヘルプ情報を表示することを強く推奨します。

ファイルをアップロード / ダウンロード / コピーするため、cp コマンドを実行する際は下記を参照してください。:

- フォルダをコピーするには、-r オプションを使用します。Ossutil は、既定では大きなファイルに対してマルチパートアップロードを実装しており、再開可能なデータ転送をサポートしています (マルチパートアップロードが有効になっている大きなファイルのしきい値は、`--bigfile-threshold` オプションを使用して設定します)。
- 既定の設定で、ファイルを強制的にアップロードするには、-f オプションを使用します。アップロード先に同じ名前のファイルが存在する場合、そのファイルは直接上書きされます。

ファイルをアップロード、ダウンロード、またはバッチ内でコピーしたときにファイルに関するエラーが発生した場合、ossutil はエラーをレポートファイルに記録、ファイルをスキップし、既定の設定では、その他のファイルに対して操作を実行します。詳しくは、「ossutil help cp」を参照してください。

❓ 説明 バケットが存在しない場合、または AccessKeyID または AccessKeySecret が正しくないために権限の検証結果が無効である場合、Ossutil はその他のファイルのコピーを続行しません。

Ossutil は、増分アップロードポリシー `--update` と `--snapshot-path` を、特定のシナリオでサポートしています。詳しくは、「ossutil help cp」を参照してください

ossutil 1.0.0. ossutil 1.0.0.Beta1 から、既定では、ファイルのアップロード中は crc64 が有効になっています。

- 1 つのファイルをアップロード

```
./ossutil cp a oss://ossutil-test
Succeed: Total num: 1, size: 230. OK num: 1(upload 1 files).
0.699795(s) elapsed
```

- フォルダをアップロード

```
./ossutil cp -r dir oss://ossutil-test
Succeed: Total num: 35, size: 464,606. OK num: 35(upload 34 files, 1 directories).
0.896320(s) elapsed
```

Ossutil 1.4.2 以降のバージョンでは、要求者支払いモードが有効になっている、バケットに格納されたオブジェクトのアップロード、ダウンロード、およびコピーをサポートしています。

- 次のコマンドを実行し、要求者支払いモードが有効になっているバケットにファイルをアップロードします。

```
./ossutil cp ~/Downloads/test.mp4 oss://payer/ --payer=requester
```

- 次のコマンドを実行し、要求者支払いモードが有効になっているバケットからオブジェクトをダウンロードします。

```
./ossutil cp oss://payer/test.mp4 ~/Downloads/ --payer=requester
```

- 次のコマンドを実行し、要求者支払いモードが有効になっているバケットから、通常のバケットにオブジェクトをコピーします。

```
./ossutil cp oss://payer/test.mp4 oss://hello-hangzws/ --payer=requester
```

- 次のコマンドを実行し、通常のバケットから要求者支払いモードが有効になっているバケットにデータをコピーします。

```
./ossutil cp oss://hello-hangzws/test.mp4 oss://payer/ --payer=requester
```

オブジェクトのストレージクラスを変更

5 GB 以下のオブジェクトのストレージクラスを変更するには、`set-meta` コマンドを実行します。5 GB を超えるオブジェクトのストレージクラスを変更するには、`cp` コマンドを実行します。

- `set-meta` コマンドを実行すると、オブジェクトのストレージクラスを変更できます。
 - 次のコマンドを実行して、1 つのオブジェクトのストレージクラスを IA に設定します。

```
./ossutil set-meta oss://hello-hangzws/0104_6.jpg X-Oss-Storage-Class:IA -u
```

- 次のコマンドを実行して、フォルダ内のすべてのオブジェクトのストレージクラスを Standard に設定します。

```
./ossutil set-meta oss://hello-hangzws/abc/ X-Oss-Storage-Class:Standard -r -u
```

- `--meta` オプションを使用してオブジェクトのストレージクラスを変更しつつ、`cp` コマンドを実行してオブジェクトをアップロードします。
 - 次のコマンドを実行して 1 つのファイルをアップロードし、そのファイルのストレージクラスを IA に設定します。

```
ossutil cp ~/Downloads/sys.log oss://hello-hangzws/test/ --meta X-oss-Storage-Class:IA
```

- 次のコマンドを実行してフォルダをアップロードし、そのフォルダ内のすべてのファイルのストレージクラスを IA に設定します。

```
./ossutil cp ~/libs3/ oss://hello-hangzws/test/ --meta X-oss-Storage-Class:IA -r
```

- 次のコマンドを実行して、既存のオブジェクトのストレージクラスを Archive に変更します。

```
./ossutil cp oss://hello-hangzws/0104_6.jpg oss://hello-hangzws/0104_6.jpg --meta X-oss-Storage-Class:Archive
```

- 次のコマンドを実行して、既存のフォルダ内のすべてのオブジェクトのストレージクラスを Standard に変更します。

```
./ossutil cp oss://hello-hangzws/test/ oss://hello-hangzws/test/ --meta X-oss-Storage-Class:Standard -r
```

注意

- `set-meta` または `cp` コマンドを実行し、オブジェクトのストレージクラスを Archive から他のクラスに変更することはできません。最初に `restore` コマンドを実行し、オブジェクトのストレージクラスを IA に変更してから、`set-meta` コマンドまたは `cp` コマンドを実行し、オブジェクトのストレージクラスを他のクラスに変更する必要があります。
- オブジェクトを上書きするため、`cp` コマンドを実行した際にオブジェクトが上書きされたと思われ、課金される場合があります。IA クラスまたは Archive クラスのオブジェクトは、作成後 30 日および 60 日以内に別々に上書きされた場合、課金されます。たとえば、オブジェクトの作成から 10 日後に `cp` コマンドを実行し、オブジェクトのストレージクラスを IA から Archive、または Standard に変更した場合、20 日間の早期削除料金が発生します。

ファイルのアップロード、ダウンロード、またはコピーのためのパフォーマンスのチューニング

`cp` コマンドでは、`--jobs` および `--parallel` オプションを使用し、並行操作数を制御します。ossutil によって設定された既定の同時操作数がパフォーマンス要件を満たせない場合、2 つのオプションの値を変更し、パフォーマンスを調整することができます。

- `--jobs` オプションは、複数のファイルがアップロード、ダウンロード、またはコピーされたとき、ファイル間で有効な同時操作数を制御します。
- `--parallel` オプションは、大きなファイルがマルチパートメソッドでアップロード / ダウンロード / コピーされたときに、サイズのファイルに対して有効な同時操作の数を制御します。

既定では、ossutil はファイルサイズに基づき、同時操作数を計算します (このオプションは小さなファイルでは機能せず、マルチパート方式によりアップロード / ダウンロード / コピーする大きなファイルのしきい値を `--bigfile -threshold` オプションで制御することができます)。大きなファイルを一括でアップロード / ダウンロード / コピーする場合、実際の同時操作数は、ジョブ数に並列操作数を掛けて計算されます。

 警告

- ECS インスタンスまたはサーバーのリソース (ネットワーク帯域幅、メモリ、CPU など) が限られている場合は、同時操作数を 100 未満の値に調整することを推奨します。リソースに余裕がある場合は、同時操作数を増やすことを試すことができます。
- ただし、実行されている同時操作が多すぎると、ossutil のアップロード、ダウンロード、およびコピーのパフォーマンスが低下したり、スレッド間でのリソースの交換やスナッチのため、EOF エラーが発生する可能性があります。この問題を解決するには、実際の条件に基づき、`-jobs` と `--parallel` の値を調整する必要があります。圧力テストを実行するには、最初に 2 つのオプションを小さい値に設定し、ゆっくりと最適値に調整していきます。

オブジェクトの管理

- オブジェクトの ACL の設定

Ossutil では、`set-acl` コマンドを使用して、オブジェクトの ACL を設定します。`-r` オプションを使用することで、オブジェクトの ACL を一括で設定できます。詳しくは、「`ossutil help set-acl`」をご参照ください。

```
./ossutil set-acl oss://dest/a private
0.074507(s) elapsed
```

オブジェクトの ACL を一括で設定する

```
./ossutil set-acl oss://dest/a private -r
Do you really mean to recursively set acl on objects of oss://dest/a(y or N)? y
Succeed: Total 3 objects. Setted acl on 3 objects.
0.963934(s) elapsed
```

- オブジェクトのメタの設定

Ossutil では `set-meta` コマンドを使用し、オブジェクトのメタ情報を設定します。`-r` オプションを使用することで、オブジェクトのメタを一括で設定できます。詳しくは、`ossutil help set-meta` をご参照ください。

```
./ossutil set-meta oss://dest/a x-oss-object-acl:private -u
```

- オブジェクトのメタに関する説明の表示

Ossutil では、`stat` コマンドを使用して、オブジェクトのメタの説明を表示します。詳しくは、`ossutil help stat` をご参照ください。

```
./ossutil stat oss://dest/a
ACL : default
Accept-Ranges : bytes
Content-Length : 230
Content-Md5 : +5vbQC/MSQK0xXSiyKBZog==
Content-Type : application/octet-stream
Etag : FB9BDB402FCC4902B4C574A2C8A059A2
Last-Modified : 2017-01-13 15:14:22 +0800 CST
Owner : aliyun
X-Oss-Hash-Crc64ecma : 12488808046134286088
X-Oss-Object-Type : Normal
0.125417(s) elapsed
```

- オブジェクトを凍結状態から読み取り可能な状態に復元

Ossutil では、オブジェクトを凍結状態から読み取り可能な状態に復元するため、restore コマンドを使用します。 `-r` オプションを使用することで、オブジェクトを凍結状態から読み取り可能な状態に一括で復元できます。詳しくは、ossutil help restore を実行してください。

```
./ossutil restore oss://utiltest/a
0.037729(s) elapsed
```

- シンボリックリンクの作成

Ossutil では、create-symlink コマンドを使用して、シンボリックリンクを作成します。詳しくは、ossutil help create-symlink を実行してください。

```
./ossutil create-symlink oss://utiltest/b a
0.037729(s) elapsed
```

- シンボリックリンクファイルに関する説明の読み取り

Ossutil では、read-symlink コマンドを使用することで、シンボリックリンクファイルの説明を読み取ります。詳しくは、ossutil help read-symlink を実行してください。

```
./ossutil read-symlink oss://utiltest/b
Etag : D7257B62AA6A26D66686391037B7D61A
Last-Modified : 2017-04-26 15:34:27 +0800 CST
X-Oss-Symlink-Target : a
0.112494(s) elapsed
```

3.7. マルチパート関連コマンド

Ossutil を使用すると、マルチパートアップロードでアップロードされたファイルと未完成のアップロードタスク (UploadID) を一覧表示し、指定したオブジェクトのアップロードファイルと未完成のアップロードタスク (UploadID) を削除できます。

マルチパートアップロードの詳細については、「[マルチパートアップロード](#)」をご参照ください。

❓ 説明 大きなファイルをアップロードまたはコピーする場合、ossutil は UploadPart コマンドを実行せず、再開可能なアップロードを自動的に使用します。

- UploadID を一覧表示する

-m オプションを指定して `ls` コマンドを実行し、指定 -a オプションを指定して `ls` コマンドを実行し、指定されたプレフィックスを持つオブジェクトのすべてのアップロード済みファイルおよび未完成のアップロードタスク (UploadID) を一覧表示します。

```
$ ossutil ls oss://bucket1/obj1 -m
InitiatedTime UploadID ObjectName
2017-01-13 03:45:26 +0000 CST 15754AF7980C4DFB8193F190837520BB oss://bucket1/obj1
2017-01-13 03:43:13 +0000 CST 2A1F9B4A95E341BD9285CC42BB950EE0 oss://bucket1/obj1
UploadId Number is: 2
0.070070(s) elapsed
```

```
$ ossutil ls oss://bucket1/obj1 -a
LastModifiedTime Size(B) StorageClass ETAG ObjectName
2015-06-05 14:36:21 +0000 CST 241561 Standard 6185CA2E8EB8510A61B3A845EAFE4174 oss://bucket1/obj1/test.txt
2016-04-08 14:50:47 +0000 CST 6476984 Standard 4F16FDAE7AC404CEC8B727FCC67779D6 oss://bucket1/obj1/sample.txt
Object Number is: 2
InitiatedTime UploadID ObjectName
2017-01-13 03:45:26 +0000 CST 15754AF7980C4DFB8193F190837520BB oss://bucket1/obj1
2017-01-13 03:43:13 +0000 CST 2A1F9B4A95E341BD9285CC42BB950EE0 oss://bucket1/obj1
UploadId Number is: 2
0.091229(s) elapsed
```

- 指定したオブジェクトのデータを削除

rm コマンドを実行して、指定したオブジェクトの未完成のアップロードタスク (UploadID) を削除します。

たとえば、bucket1 に次のオブジェクトが含まれている場合、`ls` コマンドを実行して、アップロードされたすべてのファイルと未完成のアップロードタスク (UploadID) を一覧表示します。

```
$ ossutil ls oss://bucket1 -a
LastModifiedTime Size(B) StorageClass ETAG ObjectName
2015-06-05 14:06:29 +0000 CST 201933 Standard 7E2F4A7F1AC9D2F0996E8332D5EA5B41 oss://bucket1
/dir1/obj11
2015-06-05 14:36:21 +0000 CST 241561 Standard 6185CA2E8EB8510A61B3A845EAFE4174 oss://bucket1
/obj1/test.txt
2016-04-08 14:50:47 +0000 CST 6476984 Standard 4F16FDAE7AC404CEC8B727FCC67779D6 oss://bucke
t1/sample.txt
Object Number is: 3
InitiatedTime UploadID ObjectName
2017-01-13 03:45:26 +0000 CST 15754AF7980C4DFB8193F190837520BB oss://bucket1/obj1
2017-01-13 03:43:13 +0000 CST 2A1F9B4A95E341BD9285CC42BB950EE0 oss://bucket1/obj1
2017-01-13 03:45:25 +0000 CST 3998971ACAF94AD9AC48EAC1988BE863 oss://bucket1/obj2
2017-01-20 11:16:21 +0800 CST A20157A7B2FEC4670626DAE0F4C0073C oss://bucket1/tobj
UploadID Number is: 4
0.191289(s) elapsed
```

-m オプションを指定して **rm** コマンドを実行し、指定された未完了のアップロードタスク (UploadID) を削除します。

```
$/ossutil rm -m oss://bucket1/obj1/test.txt
Succeed: Total 1 uploadIds. Removed 1 uploadIds.
0.900715(s) elapsed
```

-m オプションと **-r** オプションを指定して **rm** コマンドを実行し、指定されたプレフィックスを持つ未完了のアップロードタスク (UploadID) をすべて削除します。

```
$/ossutil rm -m oss://bucket1/ob -r
Do you really mean to remove recursively multipart uploadIds of oss:bucket1/ob(y or N)? y
Succeed: Total 4 uploadIds. Removed 4 uploadIds.
1.922915(s) elapsed
```

-a オプションと **-r** オプションを指定して **rm** コマンドを実行し、指定されたプレフィックスを持つアップロードされたすべてのファイルと未完了のアップロードタスク (UploadID) を削除します。

```
$/ossutil rm oss://hello-hangzws-1/obj -a -r
Do you really mean to remove recursively objects and multipart uploadIds of oss://obj(y or N)? y
Succeed: Total 1 objects, 3 uploadIds. Removed 1 objects, 3 uploadIds.
```


4.ossimport

4.1. アーキテクチャと構成

概要

OssImport ツールを使用すると、ローカルまたは他のクラウドストレージシステムに保存されているデータを OSS に移行できます。次のような機能があります。

- ローカルドライブ、Qiniu、Baidu BOS、AWS S3、Azure Blob など、さまざまなデータソースをサポートしています。Youpai クラウド、Tencent Cloud COS、Kingsoft KS3、HTTP、OSS など、必要に応じて拡張できます。
- 再開可能なデータ転送をサポートします。
- トラフィック制御をサポートします。
- 指定された時刻後、または指定されたプレフィックスのオブジェクトを移行することをサポートします。
- 並列データのアップロードとダウンロードをサポートします。
- スタンドアロンモードと分散モードをサポートします。スタンドアロンモードは展開と使用が簡単で、分散モードは大規模データ移行に適しています。

環境

- Java 1.7 以降

アーキテクチャ

OssImport には、スタンドアロンモードと分散モードの 2 つの展開モードがあります。

- スタンドアロンモードは、30 TB 未満の小規模なデータ移行には十分です。 [ダウンロード](#)
- ダウンロード大規模なデータ移行には、分散モードを推奨します。 [ダウンロード](#)
- スタンドアロン

master、worker、tracker、および console は同じマシン上で実行されます。システムには 1 つの worker しかありません。スタンドアロンモードの展開と実行はカプセル化され、最適化されています。スタンドアロンの展開と実行は非常に簡単です。スタンドアロンモードでは、master、worker、tasktracker、および console モジュールは `ossimport2.jar` にパッケージ化されています。

スタンドアロンモードのファイル構造は次のとおりです。

```

ossimport
├── bin
│   └── ossimport2.jar # The JAR including master, worker, tracker, and console modules
├── conf
│   ├── local_job.cfg # Standalone job configuration file
│   └── sys.properties # Configuration file for the system running
├── console.bat # Windows command line, which can run distributed call-in tasks
├── console.sh # Linux command line, which can run distributed call-in tasks
├── import.bat # The configuration file for one-click import and execution in Windows is the data
migration job configured in conf/local_job.cfg, including start, migration, validation, and retry
├── import.sh # The configuration file of one-click import and execution in Linux is the data migrat
ion job configured in conf/local_job.cfg, including start, migration, validation, and retry
├── logs # Log directory
└── README.md # Description documentation. We recommend that you carefully read the docume
ntation before using this feature

```

注記:

- mport.bat または import.sh ファイルはワンクリックのインポートスクリプトであり、local_job.cf
g への変更が完了した後、直接実行します。
- console.bat または console.sh はコマンドラインツールであり、コマンドの分散実行に使用しま
す。
- スクリプトやコマンドは、ossimport ディレクトリ、つまり *.bat/*.sh と同じレベルのディレク
トリで実行します。

● 分散

下の図に示すように、OssImport は master-worker 分散アーキテクチャに基づいています。

```

Master ----- Job ----- Console
|
|
TaskTracker
|_____
|Task | Task | Task
|||
Worker Worker Worker

```

図の説明:

- Job: ユーザーによって送信されたデータ移行ジョブ。ユーザーは 1 つのジョブは 1 つの構成ファイ
ル job.cfg に対応します。

- Task: ジョブはデータサイズとファイル数によって複数のタスクに分割できます。各タスクはファイルの一部を移行します。ジョブをタスクに分割するための最小単位はファイルです。1つのファイルを複数のタスクに分割することはできません。

OssImport ツールモジュールは次のテーブルに一覧表示されています。

ロール	説明
Master	マスターは、データサイズとファイル数によってジョブを複数のタスクに分割する役割があります。データサイズとファイル数は <code>sys.properties</code> で設定します。ジョブを複数のタスクに分割するための詳細なプロセスは次のとおりです。 i. マスターノードは、ローカル / 他のクラウドストレージデバイスから移行するファイルの一覧を全てスキャンします。 ii. マスターはデータサイズとファイル数によってファイルの一覧全体をタスクに分割し、各タスクはファイルの一部の移行または検証をする役割があります。
Worker	○ ワーカーは、ファイルの移行とタスクのデータ検証をする役割があります。データソースから特定のファイルを取り出し、そのファイルを指定されたディレクトリの OSS にアップロードします。<code>job.cfg</code> または <code>local_job.cfg</code> で、移行するデータソースと OSS 設定を指定します。 ○ ワーカーデータの移行では、トラフィックの制限と同時タスク数の指定がサポートされています。<code>sys.properties</code> で設定を構成します。
TaskTracker	TaskTracker は Tracker と略されます。タスクの分散とタスク状態を追跡する役割があります。
Console	コンソールは、ユーザーとのやり取りやコマンドの表示結果の受け取りなどの役割があります。また、 <code>deploy</code> 、 <code>start</code> 、 <code>stop</code> などのシステム管理コマンド、および <code>submit</code> 、 <code>retry</code> 、 <code>clean</code> などのジョブ管理コマンドをサポートしています。

分散モードでは、データ移行のため、複数のワーカーノードを開始できます。タスクはワーカーノードに均等に割り当てられ、1つのワーカーノードで複数のタスクを実行できます。1台のマシンで開始できるワーカーノードは1つだけです。マスターは、`workers` で設定された最初のワーカーノードと同時に開始され、タスクトラッカーとコンソールもマシン上で実行されます。

分散モードのファイル構造は次のとおりです。

```

ossimport
Miss -- Bin
| |—— console.jar # The JAR package of the console module
| |—— master.jar # The JAR package of the master module
| |—— tracker.jar # The JAR package of the tracker module
| |—— worker.jar # The JAR package of the worker module
|—— conf
| |—— job.cfg # The template of the job configuration file
| |—— sys.properties # Configuration file for the system running parameters
| |—— workers # Worker list
|—— console.sh # The command line tool. Currently it only supports Linux
|—— logs # Log directory
|—— README.md # Description documentation. We recommend that you carefully read the documentation before using the feature

```

注記:

- 分散コマンドラインツール console.sh は、現在、Linux のみをサポートし、Windows はサポートしていません。

構成ファイル

スタンドアロンモードでは、`sys.properties` と `local_job.cfg` の 2 つの構成ファイルがあります。分散モードでは、`sys.properties`、`local_job.cfg`、`workers` の 3 つの構成ファイルがあります。具体的には、`local_job.cfg` と `job.cfg` は同一ですが、名前は異なります。`workers` ファイルは分散環境専用です。

● sys.properties

システム実行パラメーター

フィールド	意味	説明
workingDir	作業ディレクトリ	○ ツールキット展開後のディレクトリです。 ○ このオプションはスタンドアロンモードでは変更しないでください。 ○ 分散モードの各マシンの作業ディレクトリは同一である必要があります。

フィールド	意味	説明
workerUser	ワーカーマシンの SSH ユーザー名	◦ privateKeyFile を設定した場合は、privateKeyFile が優先的に使用されます。 ◦ privateKeyFile が設定されていない場合は、workerUser/workerPassword の組み合わせが使用されます。 ◦ このオプションはスタンドアロンモードでは変更しないでください。
workerPassword	ワーカーマシンの SSH ユーザーパスワード	このオプションはスタンドアロンモードでは変更しないでください。
privateKeyFile	秘密鍵のファイルパス	◦ SSH チャネルを確立している場合は、公開鍵ファイルのパスを指定できます。それ以外の場合は空のままにします。 ◦ privateKeyFile を設定している場合は、privateKeyFile が優先されます。 ◦ privateKeyFile が設定されていない場合は、workerUser/workerPassword が使用されます。 ◦ このオプションはスタンドアロンモードでは変更しないでください。
sshPort	SSH ポート	既定値は "22" です。通常は変更する必要はありません。このオプションはスタンドアロンモードでは変更しないでください。

フィールド	意味	説明
workerTaskThreadNum	ワーカーがタスクを実行する最大スレッド数	- このパラメーターは、マシンのメモリとネットワークに関連しています。推奨値は 60 です。 - 値は、物理マシンの場合は 150 などに増やすことができます。ネットワーク帯域幅がすでにいっぱいになっている場合は、それ以上値を大きくしないでください。 - ネットワークが貧弱な場合は、必要に応じて値を 30 などに下げます。このようにして、要求の競合による多数の要求のタイムアウトを回避します。
workerMaxThroughput(KB/s)	ワーカーノード上のデータ移行トラフィックの上限	この値は、トラフィックを制限します。既定値 0 は、トラフィック制限が課されていないことを示します。
dispatcherThreadNum	トラッカーのタスク分散および状態確認のためのスレッド数	通常は、既定値で十分です。特殊な要件がない場合は、既定値を変更する必要はありません。
workerAbortWhenUncatchedException	不明なエラーが発生した場合、スキップするかキャンセルするかを指定します。	未知のエラーは既定ではスキップされます。
workerRecordMd5	OSS 内の移行されたファイルの MD5 値を、メタデータ x-oss-meta-md5 を使用して記録するかどうか。既定値の設定は "no" になっています。	主に MD5 を使用したファイルデータの検証に使用されます。

- job.cfg

データ移行ジョブの構成。local_job.cfg と job.cfg のオプションは、名前以外は同一です。

フィールド	意味	説明
-------	----	----

フィールド	意味	説明
jobName	ジョブ名、文字列	○ ジョブの一意の識別子。命名規則は [a-zA-Z0-9_-]{4,128} です。異なる名前の複数のジョブの申請をサポートします。 ○ 別のジョブと同じ名前のジョブを送信すると、システムはそのジョブがすでに存在することを通知するメッセージを表示します。元のジョブを名前でクリーンアップする前に、同じ名前のジョブを送信することはできません。
jobType	ジョブタイプ、文字列	"import" と "audit" の 2 つのタイプがあります。既定値は、"import" です。既定値は、"import" です。 ○ import: データ移行を実行し、移行されたデータの整合性を検証します。 ○ audit: データの一貫性のみを検証します。
isIncremental	増分移行モードをブール値で有効にするかどうか	○ 既定値: "false" ○ "true" に設定されている場合、増分データは incrementalModelInterval (単位: 秒) で指定された間隔で再スキャンされ、OSS と同期されます。
incrementalModelInterval	増分モードでの同期間隔、整数値。単位: 秒	isIncremental=true のときに有効です。設定可能な最小間隔は 900 秒です。大量の要求が浪費され、システムオーバーヘッドが増加するため、3,600 秒よりも小さい値に設定することは推奨されません。

フィールド	意味	説明
importSince	この時間値 (整数値) より後にデータを移行します。単位: 秒	- この時間値は UNIX のタイムスタンプです。つまり、1970年 1 月 1 日の UTC 00:00 からの秒数です。date +%s コマンドで値を取得します。 - 既定値は "0" で、すべてのデータを移行することを示します。
srcType	同期ソースタイプ、文字列。大文字と小文字を区別します。	現在、このパラメーターは local、oss、qiniu、bos、ks3、s3、youpai、HTTP、cos、azure を含む 10 種類をサポートしています。 - local: ローカルファイルから OSS にデータを移行します。このオプションには srcPrefix を入力するだけでよく、srcAccessKey、srcSecretKey、srcDomain、および srcBucket を入力する必要はありません。 - あるバケットから別のバケットにデータを移行します。 - qiniu: Qiniu クラウドストレージから OSS にデータを移行します。 - bos: Baidu クラウドストレージから OSS にデータを移行します。 - ks3: Kingsoft クラウドストレージから OSS にデータを移行します。 - s3: AWS S3 から OSS にデータを移行します。 - youpai: Youpai Cloud から OSS にデータを移行します。 - HTTP: 提供された HTTP リンク一覧を介して、データを OSS に移行します。 - cos: Tencent クラウドストレージ COS から OSS にデータを移行します。 - azure: Azure Blob から OSS にデータを移行します。

フィールド	意味	説明
srcAccessKey	ソース AccessKey、文字列	srcType が oss、qiniu、baidu、ks3、または s3 に設定されている場合は、データソースの AccessKey を入力します。 ◦ または local と HTTP タイプの場合、このオプションは空のままにすることができます。 ◦ youpai タイプと azure タイプの場合は、AccountName を入力します。
srcSecretKey	ソース SecretKey、文字列	srcType が oss、qiniu、baidu、ks3、または s3 に設定されている場合は、データソースの SecretKey を入力します。 ◦ local と HTTP タイプの場合、このオプションは空のままにすることができます。 ◦ youpai: オペレーターのパスワードを入力します。 ◦ azure: AccountKey を入力します。

フィールド	意味	説明
srcDomain	ソースエンドポイント	srcType が local または HTTP に設定されている場合、この設定項目は不要です。 - oss: コンソールから取得したドメイン名です。これは、バケットプレフィックスのない第2レベルのドメイン名です。完全な一覧はドメイン名一覧にあります。 - qiniu: Qiniu コンソールから取得した対応するバケットのドメイン名です。 - bos: <code>http://bj.bcebos.com</code> や <code>http://gz.bcebos.com</code> などの Baidu BOS ドメイン名です。 - ks3: <code>http://kss.ksyun.com</code>、<code>http://ks3-cn-beijing.ksyun.com</code>、または <code>http://ks3-us-west-1.ksyun.com</code> などの Kingsoft KS3 のドメイン名。 - さまざまなリージョンの S3 および AWS S3 ドメイン名は、S3 エンドポイントにあります。 - youpai: Youpai Cloud のドメイン名。例えば光回線の <code>http://v0.api.upyun.com</code>、通信回線の <code>http://v1.api.upyun.com</code>、China Unicom または China Netcom line の <code>http://v2.api.upyun.com</code>、China Mobile または China Railcom line の <code>http://v3.api.upyun.com</code> があります。 - cos: Tencent Cloud のバケットのリージョンです。南中国: gz、北中国: tj、東中国: sh などがあります。 - azure: <code>core.chinacloudapi.cn</code> などの EndpointSuffix が Azure Blob の接続文字列にあります。

フィールド	意味	説明
srcBucket	ソースバケットまたはコンテナの名前	srcType が local または HTTP に設定されている場合、この設定項目は不要です。azure: Azure Blob のコンテナ名を入力し、他のバケット名を入力します。
srcPrefix	ソースプレフィックス、文字列。 既定値は "empty" です。	srcType が local に設定されている場合は、c:/example/ または /data/example/ のように、ローカルディレクトリを完全に区切ります。区切り文字は "/" を入力します。srcType が oss、qiniu、bos、ks3、youpai、または s3 の場合、値は data/to/oss/ のように、バケット名を除いた同期対象オブジェクトのプレフィックスです。すべてのオブジェクトを同期したい場合は、srcPrefix を空のままにしてください。
	宛先 AccessKey、文字列	OSS の AccessKeyId を表示するには、 コンソール にログインします。
destSecretKey	宛先 SecretKey、文字列	OSS の AccessKeySecret を表示するには、 コンソール にログインします。
destDomain	宛先エンドポイント、文字列	コンソール から取得します。これは、バケットプレフィックスのない第 2 レベルのドメイン名です。完全な一覧はドメイン名一覧にあります。
destBucket	宛先バケット、文字列	OSS バケット名 "/" で終わる必要はありません。

フィールド	意味	説明
destPrefix	宛先プレフィックス、文字列。既定値は "empty" です。	- 宛先プレフィックス。既定値は "empty" です。その場合、オブジェクトは宛先バケットに配置されます。 - OSS 上の特定のディレクトリにデータを同期する場合は、data/in/oss/ のようにプレフィックスを "/" で終わらせます。 - OSSは "/" をオブジェクトプレフィックスとしてサポートしていないので、destPrefixを "/" で始まるように設定しないでください。 - パス srcPrefix+relativePath 内のローカルファイルは、OSS 上の destDomain/destBucket/destPrefix + relativePath に移行されます。 - パス srcDomain/srcBucket/srcPrefix+relativePath にあるクラウド上のオブジェクトは、OSS 上の destDomain/destBucket/destPrefix + relativePath に移行されます。
taskObjectCountLimit	タスク内のファイルの最大数 (整数)。既定値は "10,000" です。	この構成オプションは、実行されたジョブの並行性に影響します。通常、構成はファイルの総数 / ワーカーの総数 / 移行スレッド数 (workerTaskThreadNum) に設定され、最大数は 50,000 です。ファイルの総数がわからない場合は、既定値を使用します。
taskObjectSizeLimit	タスク内の最大データサイズ (整数)。Unit: bytes. 既定値は "1GB" です。	この構成オプションは、実行されたジョブの並行性に影響します。通常、構成は、合計データサイズ / 合計ワーカー数 / 移行スレッド数 (workerTaskThreadNum) に設定されます。合計データサイズが不明な場合は、既定値を使用します。

フィールド	意味	説明
isSkipExistFile	データ移行中に既存のオブジェクトをスキップするかどうか (ブール値)。	"true" に設定すると、オブジェクトはサイズと LastModifiedTime に従ってスキップされます。"false" に設定すると、既存のオブジェクトは上書きされます。既定値は "false" です。jobType が "audit" に設定されている場合、このオプションは無効です。
scanThreadCount	並列ファイルスキャンのスレッド数 (整数)。既定値は "1" です。	この設定オプションは、ファイルのスキャン効率に関連しています。特殊な要件がない場合は、設定を変更しないでください。
maxMultiThreadScanDepth	並列スキャンのディレクトリの最大許容深度 (整数)。既定値は "1" です。	- 既定値の1は、最上位ディレクトリでの並列スキャンを示します。 - 特殊な要件がない場合は、設定を変更しないでください。値が大きすぎると、ジョブは正常に実行されない可能性があります。
appld	Tencent COS の appld (整数)。	srcType が cos に設定されている場合に有効です。

フィールド	意味	説明
httpListFilePath	HTTP 一覧ファイルの絶対パス、文字列。	- srcType が HTTP に設定されている場合に有効です。ソースが HTTP リンクアドレスの場合、c:/example/http.list のように、コンテンツとして HTTP リンクアドレスを含むファイルの絶対パスを指定する必要があります。 - ファイル内の HTTP リンクは、以下の内容を含む c:/example/http.list のように、それぞれアップロード後の OSS 上のプレフィックスと相対パスを表すスペースで区切られた 2 つの列に分割する必要があります。 http://mingdi-hz.oss-cn-hangzhou.aliyuncs.com/aa/ bb.jpg http://mingdi-hz.oss-cn-hangzhou.aliyuncs.com/cc/dd.jpg OSS に移行した後の 2 行のオブジェクト名は、それぞれ destPrefix + bb.jpg および destPrefix + cc/dd.jpg です。

● Workers

ワーカーは分散モード専用であり、すべての IP アドレスは次のような行になります。

```
192.168.1.6
192.168.1.7
192.168.1.8
```

注記:

- 上記の設定では、最初の行の 192.168.1.6 がマスターになる必要があります。つまり、マスター、ワーカー、およびタスクトラッカーは 192.168.1.6 で開始され、コンソールもマシン上で実行される必要があります。
- ユーザー名、ログインモード、および複数のワーカーモードの作業ディレクトリが同一であることを確認します。

構成ファイルの例

次のテーブルに、分散展開のデータ移行タスクプロファイルを示します。スタンドアロンマシンの構成ファイル名は "local_job.cfg" です。構成項目と分散展開に違いはありません。

移行タイプ	設定ファイル	説明
OSSにローカルで移行	job.cfg	Srcprefix は / の末尾の絶対パスです (D:/work/oss/data/ 、 /home/user/work/oss/data/ など)。
Seven Bull Cloud ストレージから OSS へ移行	job.cfg	Srcprefix と DESTIN プレフィックスは空になるように設定できます。空でない場合は、destPrefix=docs/ のように / で終わります。
Baidu BosからOSS への転送	job.cfg	Srcprefix と DESTIN プレフィックスは空になるように設定できます。空でない場合は、destPrefix=docs/ のように / で終わります。
AWS S3 から OSS への移行	job.cfg	S3 のドメイン名
クラウドストレージから OSS への再移行	job.cfg	Srcaccesskey/Scanner はオペレーターのアカウント番号とパスワードを入力します。
Tencent CO から OSS への移行	job.cfg	srcDomain=sh などの V4 バージョンに従ってください。Srcprefix は空でもかまいません。空でない場合は、srcPrefix=/docs/ のように、前後に / を付けます。
azure Blob からOSSへの移行	job.cfg	Srcaccesskey/srcsecretkey はストレージの cun chu と key を埋めます。srcdomain は、core.chinacloudapi.cn などの接続文字列 Endpointsuffix を入力します。

移行タイプ	設定ファイル	説明
OSS 間での移行	<code>job.cfg</code>	異なるリージョン間、異なるストレージタイプ間、および異なるプレフィックス間のデータ移行に適しています。ECS 上に配置し、トラフィックを節約するため、内部ドメイン名を使用することを推奨します。
高速チャンネルを使用した OSS へのデータ移行	<code>job.cfg</code>	すべてのデータソースに適用されます。高速移行要件がある場合、ジョブを送信するか、またはチケットを起票し、OSS サポートセンターへお問い合わせください。

4.2. スタンドアロン展開

ダウンロード

スタンドアロン展開は Linux と Windows をサポートしています。

スタンドアロン展開用のツール、`ossimport-2.3.3.zip` をダウンロードします。ツールをローカルディレクトリにダウンロードし、ツールまたは `unzip` コマンドを使用してファイルを解凍します。解凍後のファイル構造は次のとおりです。

```
ossimport
├── bin
│   └── ossimport2.jar # The JAR including master, worker, tracker, and console modules
├── conf
│   ├── local_job.cfg # The job configuration file
│   └── sys.properties # Configuration file of the system running parameters
├── console.bat # Windows command line, which can run distributed call-in tasks
├── console.sh # Linux command line, which can run distributed call-in tasks
├── import.bat # The configuration file for one-click import and execution in Windows is the data migration job configured in conf/local_job.cfg, including start, migration, validation, and retry
├── import.sh # The configuration file for one-click import and execution in Linux is the data migration job configured in conf/local_job.cfg, including start, migration, validation, and retry
├── logs # Log directory
└── README.md # Description documentation. We recommend that you carefully read the documentation before using the feature
```

設定

スタンドアロンバージョンには、次の 2 つの設定ファイルがあります。: `conf/sys.properties` および `conf/local_job.cfg`

- `conf/sys.properties` 内の構成項目 (`workingDir` 、 `workerUserName` 、 `workerPassword` 、 および `privateKeyFile`) は変更しないでください。
- `conf/local_job.cfg` の名前と保存先を変更しないでください。 `jobName` 構成項目を変更しないでください。

他の項目を適切に設定します。

② 説明 ジョブを実行する前に、`sys.properties` および `local_job.cfg` のパラメーターを確認します。ジョブ内のパラメーターは、ジョブの実行後に変更することはできません。

実行

スタンドアロンモードでは、データ移行ジョブには ワンクリックインポートとステップバイステップ実行の2つの実行モードがあります。

ワンクリックインポートによりすべての手順がカプセル化され、スクリプトの指示に従ってデータの移行を完了させることができます。

② 説明 新規ユーザーにはワンクリックインポートを強く推奨します。

段階的な実行には、開始サービスの実行、ジョブの実行依頼、失敗したタスクの再試行が含まれます。

- ワンクリックインポート
 - i. ワンクリックインポートを実行するには、Windows の `cmd.exe` で `import.bat` を実行し、Linux の `bash import.sh` を実行します。
 - ii. 以前にこのジョブを実行したことがある場合は、最後のブレークポイントからジョブを続行するかどうか、または新しい同期ジョブを実行するかどうかを尋ねられます。新しいデータ移行ジョブを開始した場合、または同期化されたソースエンドとデスティネーションエンドを変更した場合は、同期ジョブを再度実行します。
 - iii. Windows でジョブが開始されると、新しい `cmd` ウィンドウが開き、進行中の同期ジョブとログが表示されます。古いウィンドウのジョブステータスは 10 秒ごとに更新されます。データ移行プロセス中に前述の2つのウィンドウを閉じないでください。Linux では、上記のプロセスはバックグラウンドで実行されます。
 - iv. ジョブが完了したときにタスクが失敗していた場合は、再試行するかどうかを尋ねられます。再試行するには `y` を入力、このステップをスキップして終了するには `n` を入力します。
 - v. アップロードが失敗した理由を確認するには、ファイル `master/jobs/local_test/failed_tasks/<taskid>/audit.log` を開き、失敗の原因を確認します。
- ステップバイステップ実行
 - i. 同じ名前のジョブを消去します。以前に同じ名前のジョブを実行したことがあり、そのジョブを再度実行したい場合は、まず同じ名前のジョブを消去します。ジョブを一度も実行したことがないか、失敗したジョブを再試行したい場合は、`clear` コマンドを実行しないでください。Windows では、`cmd.exe` で `console.bat clean` を実行します。Linux では、`bash console.sh clean` を実行します。
 - ii. データ移行ジョブを送信します。OssImport は、同じ名前のジョブの送信をサポートしていません。同じ名前のジョブが存在する場合は、最初に同じ名前のジョブを消去します。投入された

ジョブの構成ファイルは `conf/local_job.cfg` 、既定のジョブ名は "`local_test`" です。ジョブを送信するには、Windows の `cmd.exe` で `console.bat submit` を実行し、Linux の `bash console.sh submit` を実行します。

- iii. サービスを開始します。Windows の `cmd.exe` で `console.bat start` を実行し、Linux の `bash console.sh start` を実行します。
- iv. ジョブの状況を表示します。Windows の `cmd.exe` で `console.bat start` を実行し、Linux の `bash console.sh start` を実行します。
- v. 失敗したタスクを再試行します。ネットワークの問題やその他の原因により、タスクが失敗することがあります。失敗したタスクのみが再試行されます。Windows の `cmd.exe` で `console.bat` の再試行を実行し、`bash console.sh` の再試行を実行します。
- vi. サービスを停止します。Windows でウィンドウ `%JAVA_HOME%/bin/java.exe` を閉じ、Linux で `bash console.sh stop` を実行します。

❓ 説明 特別な要件がない場合は、データ移行にワンクリックインポートを使用することを推奨します。

- 失敗の一般的な原因
 - アップロードの処理中に、ソースディレクトリ内のファイルが変更されたとします。この原因は、`log/audit.log` の `SIZE_NOT_MATCH` エラーによって示されます。この場合、古いファイルは正常にアップロードされていますが、変更は OSS に同期されていません。
 - アップロード処理中にソースファイルが削除されると、ダウンロードに失敗します。
 - ソースファイル名が OSS の命名規則に準拠していないと、(ファイル名を / で始めることはできません)、アップロードは失敗します。
 - データソースファイルのダウンロードに失敗。
 - プログラムは予期せず終了します。ジョブ状況は "中止" です。このような場合は、チケットを起票し、サポートセンターへお問い合わせください。
- ジョブの状態とログ

ジョブが送信された後、マスターはジョブをタスクに分割し、ワーカーはタスクを実行し、トラッカーはタスクのステータスを収集します。ジョブが完了すると、ossimport ディレクトリには次のものが含まれます。

```

ossimport
├── bin
│   └── ossimport2.jar # The standalone version JAR
├── conf
│   ├── local_job.cfg # The job configuration file
│   └── sys.properties # Configuration file of the system running parameters
├── console.sh # The command line tool
├── import.sh # One-click import script
├── logs
│   ├── import.log # Archive logs
│   ├── job_stat.log # Job status record
│   ├── ossimport2.log # Running log of the standalone version
│   └── submit.log # Job submission record
├── master
│   ├── jobqueue # Store jobs that have not been fully split
│   ├── jobs # Store the job running status
│   ├── local_test # Job name
│   ├── checkpoints # The checkpoint record of the master splitting the job to tasks
│   │   └── 0
│   │       └── 034DC9DD2860B0CFE884242BC6FF92E7.cpt
│   ├── dispatched # Tasks that have been assigned to the workers but haven't been fully run
│   │   └── localhost
│   ├── failed_tasks # Tasks that failed to run
│   ├── pending_tasks # Tasks that have not been assigned
│   ├── succeed_tasks # Tasks that run successfully
│   │   └── A41506C07BF1DF2A3EDB4CE31756B93F_1499744514501@localhost
│   ├── audit.log # The task running log. You can view the error causes in the log
│   ├── DONE # Mark of successful tasks
│   ├── error.list # The task error list. You can view the error file list
│   ├── STATUS # The task status marker file. The content is Failed or Completed
│   │   └── TASK # The task description information
│   └── worker # Status of the task being run by the worker. After running, tasks are managed by the
master
├── jobs
├── local_test
└── tasks

```

注記:

- ジョブの実行情報については、`log/ossimport2.log` をご参照ください。

- タスク失敗の原因については、`master/jobs/${JobName}/failed_tasks/${TaskName}/audit.log` を表示してください。
- 失敗したタスクファイルについては、`master/jobs/${JobName}/failed_tasks/${TaskName}/error.list` を表示してください。

よくあるご質問

「よくあるご質問」をご参照ください。

4.3. 分散配置

ダウンロード

分散配置は現在、Linux のみをサポートし、Windows はサポートしていません。

分散配置用のツール、[ossimport-2.3.3.tar.gz](#) をダウンロードします。

ツールをローカルディレクトリにダウンロードし、コマンド `tar -zxvf ossimport-2.3.2.tar.gz -C $HOME/ossimport` を使用してファイルを解凍します。解凍後のファイル構造は次のとおりです。

```
ossimport
├── bin
│   ├── console.jar # The JAR package of the console module
│   ├── master.jar # The JAR package of the master module
│   ├── tracker.jar # The JAR package of the tracker module
│   └── worker.jar # The JAR package of the worker module
├── conf
│   ├── job.cfg # The template of the job configuration file
│   ├── sys.properties # Configuration file of the system running parameters
│   └── workers # Worker list
├── console.sh # The command line tool. Currently it only supports Linux
├── logs # Log directory
└── README.md # Description documentation. Read it carefully before use
```

注記:

- `OSS_IMPORT_HOME`: OssImport のルートディレクトリです。既定では、このディレクトリは `unzip` コマンドの `$HOME/ossimport` です。 `export OSS_IMPORT_HOME=<dir>` を実行するか、システム設定ファイル `$HOME/.bashrc` を変更してディレクトリを設定することもできます。
- `OSS_IMPORT_WORK_DIR`: OssImport の作業ディレクトリです。 `conf/sys.properties` の構成アイテム `workingDir` を介して、ディレクトリを指定します。推奨値は `$HOME/ossimport/workdir` です。
- `OSS_IMPORT_HOME` または `OSS_IMPORT_WORK_DIR` には、 `/home/<user>/ossimport` または `/home/<user>/ossimport/workdir` などの絶対パスを使用します。

設定

配布のバージョンには、`conf/sys.properties`、`conf/job.cfg`、`conf/workers` の3つの設定ファイルがあります。設定項目の説明については、「導入部チャプター」をご参照ください。

- `conf/job.cfg` : 分散モードのジョブの設定ファイルテンプレートです。データを移行する前に、実際のパラメーターに従って値を変更します。
- `conf/sys.properties` : 作業ディレクトリやワーカーの実行パラメーターなど、システムの実行パラメーターの設定ファイルです。
- `conf/workers` : ワーカーリストです。

注記:

- ジョブの実行を送信する前に、`sys.properties` および `job.cfg` のパラメーターを確認します。ジョブ内のパラメーターは、ジョブの投入後に変更することはできません。
- サービスを開始する前に、ワーカーリストの `ワーカー` を決定します。サービスが開始された後は、ワーカーを追加または削除することはできません。

実行

- コマンドの実行

分散展開におけるジョブ実行の一般的な手順は次のとおりです。

- ジョブ構成ファイルの変更
- サービスの配置

Linux で `bash console.sh deploy` を実行します。

❓ 説明 設定ファイル `Conf/job.cfg` と `CONF/workers` が配置する前に修正されていることを確認します。

- 同じ名前のジョブを消去

以前に同じ名前のジョブを実行し、そのジョブを再度実行する場合は、最初に同じ名前のジョブを消去します。ジョブを一度も実行したことがないか、失敗したジョブを再試行したい場合は、`clear` コマンドを実行しないでください。Linuxで `bash console.sh clean job_name` を実行します。

- データ移行ジョブの送信

OssImport は、同じ名前のジョブの送信をサポートしていません。同じ名前のジョブが存在する場合は、最初に `clean` コマンドを使用して同じ名前のジョブを消去します。ジョブを送信するには、ジョブの構成ファイルを指定する必要があります。ジョブの設定ファイルテンプレートは、`conf/job.cfg` にあります。テンプレートに基づき、設定を変更することを推奨します。Linux で `bash console.sh submit [job_cfg_file]` を実行し、構成ファイル `job_cfg_file` を使用してジョブを送信します。`job_cfg_file` はオプションのパラメータです。パラメーターを指定しない場合、既定ではパラメーターは `$OSS_IMPORT_HOME/conf/job.cfg` です。`$OSS_IMPORT_HOME` は、既定では `console.sh` ファイルがあるディレクトリです。

- 移行タスクの開始

Linux で `bash console.sh start` を実行します。

- ジョブ状態の表示

Linux で `bash console.sh stat` を実行します。

- 失敗したタスクの再試行

ネットワークの問題またはその他の原因により、タスクの実行は失敗することがあります。失敗したタスクのみが再試行されます。Linux で `bash console.sh` を実行し、`[job_name]` を再試行します。`job_name` は、`job_name` という名前のジョブのタスクを再試行することを指定するオプションのパラメーターです。`job_name` パラメーターが指定されていない場合、ジョブが失敗したタスクはすべて再試行されます。

- 移行ジョブの停止

Linux で `bash console.sh stop` を実行します。

注記:

- `bash console.sh` パラメーターにエラーがあると、`console.sh` は自動的にコマンド形式のプロンプトを出します。
- 設定ファイルと送信されたジョブのディレクトリには、絶対パスを使用することを推奨します。
- ジョブの設定 (つまり、`job.cfg` 内の設定項目) は送信後に変更できません。

- ジョブの失敗の一般的な原因

- アップロード処理中に、ソースディレクトリ内のファイルが変更されたとします。この原因は、`log /audit.log` の `SIZE_NOT_MATCH` エラーによって示されます。この場合、古いファイルは正常にアップロードされていますが、変更は OSS に同期されていません。
- アップロード処理中にソースファイルが削除されると、ダウンロードが失敗します。
- ソースファイル名が OSS の命名規則に準拠していない (ファイル名を "/" または空にすることはできません) ため、OSS へのアップロードが失敗します。
- データソースファイルのダウンロード失敗。
- プログラムは予期せず終了し、ジョブの状態は "中止" です。このような場合は、チケットを起票し、サポートセンターへお問い合わせください。

- ジョブの状態とログ

ジョブが送信されると、マスターがそのジョブをタスクに分割し、ワーカーがタスクを実行し、トラッカーがタスクの状態を収集します。ジョブが完了すると、`workdir` ディレクトリには次のものが含まれます。

```
workdir
├── bin
│   ├── console.jar # The JAR package of the console module
│   ├── master.jar # The JAR package of the master module
│   ├── tracker.jar # The JAR package of the tracker module
│   └── worker.jar # The JAR package of the worker module
├── conf
│   ├── job.cfg # The template of the job configuration file
│   ├── sys.properties # Configuration file of the system running parameters
│   └── workers # Worker list
```

```

├── logs
│   ├── import.log # Archive logs
│   ├── master.log # Master logs
│   ├── tracker.log # Tracker logs
│   └── worker.log # Worker logs
├── master
│   ├── jobqueue # Store jobs that have not been fully split
│   ├── jobs # Store the job running state
│   ├── xxtooss # Job name
│   ├── checkpoints # The checkpoint record that the master splits the job to tasks
│   │   └── 0
│   │       └── ED09636A6EA24A292460866AFDD7A89A.cpt
│   ├── dispatched # Tasks that have been assigned to the workers but haven't been fully run
│   │   └── 192.168.1.6
│   ├── failed_tasks # Tasks that failed to run
│   │   └── A41506C07BF1DF2A3EDB4CE31756B93F_1499348973217@192.168.1.6
│   │       ├── audit.log # The task running log. You can view the error causes in the log
│   │       ├── DONE # Mark of successful tasks. If the task fails, the mark is empty
│   │       ├── error.list # The task error list. You can view the error file list
│   │       └── STATUS # The task state mark file. The content is Failed or Completed, indicating that the task failed or succeeded
│   ├── TASK # The task description information
│   ├── pending_tasks # Tasks that have not been assigned
│   ├── succeed_tasks # Tasks that run successfully
│   │   └── A41506C07BF1DF2A3EDB4CE31756B93F_1499668462358@192.168.1.6
│   ├── audit.log # The task running log. You can view the error causes in the log
│   ├── DONE # Mark of successful tasks
│   ├── error.list # Task error list. If the task is successful, the list is empty
│   └── STATUS # The task state mark file. The content is Failed or Completed, indicating that the task failed or succeeded
├── TASK # The task description information
├── worker # state of the task being run by the worker. After running, tasks are managed by the master
├── jobs
├── local_test2
│   ├── tasks
│   ├── local_test_4
│   └── tasks

```

注記:

- ジョブの実行状態については、logs/tracker.log を表示します。ログを実行しているワーカーの場合は、logs/worker.log を表示します。実行中のマスターログについては、logs/master.log をご参照ください。
- タスク失敗の原因については、master/jobs/\${JobName}/failed_tasks/\${TaskName}/audit.log をご参照ください。
- 失敗したタスクファイルについては、master/jobs/\${JobName}/failed_tasks/\${TaskName}/error.list を表示してください。

よくある質問

「よくある質問」をご参照ください。

4.4. データ移行

本ページでは、主に OssImport の一般的な適用方法と代表的な要件の実装方法について説明します。

OssImport の概要

展開モード

OssImport には、スタンドアロンモードと分散モードの 2 つの展開モードがあります。

- データサイズが 30 TB 未満の小規模なデータ移行は、スタンドアロンモードで十分です。
- 大規模データサイズの移行には、分散モードを推奨します。

時間別のトラフィック制限

OssImport はマスターワーカーの分散アーキテクチャに基づいています。ワーカーはトラフィック制限機能を提供しています。設定ファイル sys.properties の workerMaxThroughput (KB/s) (KB / s) 項目を変更することで、トラフィック制限を実装できます。この設定項目はすぐには有効になりません。設定項目を有効にするために、設定の変更後にサービスを再起動します。

分散展開モードで、各ワーカーの \$OSS_IMPORT_WORK_DIR/conf にある sys.properties を変更してからサービスを再起動します。

crontab を使用して sys.properties の時限変更を実装してから、サービスを再起動して時間別のトラフィック制限を実装します。

ワーカーを追加

ジョブを送信する前にワーカーリストを決定します。現在の OSS はワーカーの動的追加をサポートしていません。

移行が不要のデータ検証

OssImport は移行を伴わないデータ検証をサポートしています。ジョブ構成ファイル job.cfg または local_job.cfg では、構成項目 jobType は import ではなく audit です。その他の設定項目はデータ移行と同じです。

データ移行の増分モード

データ移行の増分モードとは、データ移行ジョブの開始後にまず完全移行を実行し、次に設定された間隔で自動的に増分データ移行操作を実行するプロセスを指します。最初のデータ移行ジョブは完全移行です。ジョブは送信直後に開始されます。それ以降のデータ移行ジョブは、設定された間隔ごとに 1 回開始されます。データ移行の増分モードは、データのバックアップとデータの同期に適用されます。

増分モードには 2 つの構成項目があります。

- job.cfg の `isIncremental` は、増分移行モードが有効になっているかどうかを示します。 `true` は増分モードが有効であることを、 `false` は、増分モードが無効になっていることを意味します。既定値は `false` です。
- job.cfg の `incrementalModelInterval` は、増分モードの同期間隔を秒単位で指定します。設定は、`isIncremental` が `true` に設定されている場合に使用されます。設定可能な最小値は 900 秒です。3600 秒よりも小さい値に設定することは推奨しません。大量のリクエストが無駄になり、追加のシステムオーバーヘッドが発生してしまうからです。

オブジェクト移行のためのフィルタリング条件の指定

指定したフィルタ条件を満たすオブジェクトのみが移行されます。OssImport は、`prefix` と `last modified time` の指定をサポートしています。

- job.cfg の `srcPrefix` 設定では、移行するオブジェクトのプレフィックスを指定します。既定では空です。
 - `srcType=local` の場合は、ローカルディレクトリをフルパスで入力し、入力値を `/` で区切り、入力値を `/` で終了させます `c:/example/` または `/data/example/` など)。
 - `srcType` が `oss`、`qiniu`、`bos`、`ks3`、`youpai`、または `s3` の場合は、`data/to/oss/` のように、バケット名を除いて、同期するオブジェクトのプレフィックスを入力します。すべてのオブジェクトの `srcPrefix` は空に設定する必要があります。
- job.cfg の `importSince` オプションでは、移行オブジェクトの最終更新日時を指定します。整数かつ秒単位で表されます。`importSince` 設定は、UNIX タイムスタンプ形式です。つまり、1970 年 1 月 1 日の 00:00 UTC からの秒数です。`date +%s` コマンドにより、値を取得できます。既定値は "0" で、すべてのデータを移行することを示します。増分モードのデータ移行は、最初の完全移行でのみ有効です。非増分モードは、移行ジョブ全体に対して有効です。
 - オブジェクトの `LastModified Time` が `importSince` 以前の場合、オブジェクトは移行されます。
 - オブジェクトの `LastModified Time` が `importSince` より後の場合は、オブジェクトは移行されません。

一般的なシナリオ

サードパーティーのストレージサービスから OSS へのシームレスな切り替え

次の手順に従って、他のストレージサービスから OSS にシームレスに切り替えることができます。

1. 完全移行。この時点では、ビジネスはまだサードパーティー製のストレージサービスで実行されています。データ移行 T1 の開始時間を記録します。時間は UNIX のタイムスタンプ形式、つまり 1970 年 1 月 1 日の UTC 00:00 からの秒数であることにご注意ください。`date +%s` コマンドにより値を取得できます。
2. OSS イメージのオリジン検索機能を開きます。データ移行が完了したら、OSS コンソールでバケットの **イメージのオリジン取得** 機能を設定します。
3. 読み取り / 書き込みを OSS に切り替えます。この時点で、T1 より前のデータは OSS から読み取られ、T1 より後のデータはイメージオリジン検索を使用してサードパーティーサービスから読み取られ、新しいデータは OSS に完全に書き込まれます。
4. 増分データ移行。構成ファイル (`job.cfg` または `local_job.cfg`) では、増分移行ジョブの設定項目は `importSince=T1` です。増分移行は T2 で完了します。

❓ 説明 増分データ移行は、データ移行の増分モードではありません。

5. サードパーティ製のストレージを削除します。T2以降は、業務の読み書きはすべてOSSで行われ、サードパーティーのストレージは履歴データのコピーにすぎません。サードパーティストレージを保持するか削除するかは、ユーザーの判断で決定できます。OssImportはデータの移行と検証を行う役割があります。データは削除されません。

ローカルデータのOSSへの移行

ローカルデータをOSSに移行するためのツール

- 30TB未満のローカルデータファイルを移行する場合、またはストレージサービスをローカルファイルシステムにマウントする場合は、ossutilの使用を推奨します。このツールは簡単で使いやすく便利です。OssUtilはオブジェクトレベルでの増分アップロードをサポートしており、`-u/--update` および `--snapshot-path` オプションを介して機能を実装します。詳細な説明については、`ossutil help cp` コマンドを実行してご確認ください。
- 大規模データの移行には、OssImportの分散バージョンを推奨します。

❓ 説明 ローカルデータの増分移行中、Windowsのcpとmv、Linuxの `-t` または `-a` オプション付きのmvとrsyncなど、ファイルシステムの一部の操作では、オブジェクトの最終更新日時は変更されません。これらの操作によるデータ変更は検出されず、OSSにも同期されません。

OSS間のデータ移行

- OssImportを使用する場合
 - 異なるリージョンのOSS間におけるデータ移行のためのクロスリージョンレプリケーション機能を追加する場合、コンソールで設定することができます。
 - セキュリティ上の理由から、あるリージョンがクロスリージョンレプリケーションをまだサポートしていない場合は、OssImportを使用することでデータを移行またはバックアップすることができます。
 - 同じリージョン内の異なるアカウントとバケット間でのデータ移行。
 - OSS内で直接データを移行するため、Alibaba Cloudイントラネットを推奨します。つまり、`internal` のECSまたはOSSドメイン名を使用します。
- OSS内での直接データ移行の料金
 - `internal` のドメイン名を使用する場合、トラフィック料金は発生しませんが、リクエストとストレージ料金を支払う必要があります。
 - `internal` でドメイン名を使用しなかった場合は、トラフィック料金が発生する可能性があります。
- 推奨されない使用例
 - クロスリージョンレプリケーションサービスがアクティブ化されているリージョン間でのデータ移行。
 - 増分モードによりOSS間でオブジェクトに対する変更を同期する場合、OssImportはオブジェクト変更の同期(`put/append/multipart`)のみをサポートし、読み取りおよび削除操作の同期はサポートしません。データの同期が特定のSLAによって適時に行われることは保証されていません。このオプションを選択するときは注意してください。アップロードコールバックの使用を推奨しています。

移行手順

ECS およびトラフィック

クラウド (非ローカル) から OSS にデータを移行する際、帯域幅リソースが不十分な場合は、移行用の従量課金の ECS インスタンスを購入することを推奨します。ECS の設定

- 課金方法に [従量課金] を選択します。
- OSS の該当するリージョンを選択します。
- 帯域幅ピークは100 MB を選択します。

移行ジョブ構成で、targetDomain を internal を含むイントラネットドメイン名に設定します。ソースエンドも OSS の場合は、srcDomain も internal を含むイントラネットドメイン名に設定します。これにより、OSS ソースドメイン名からのダウンロードにコストがかからず、OSS のアクセスに対してのみ料金が請求されます。

HTTP データの OSS への移行

HTTP データ移行ジョブで設定するパラメーター:

- job.cfg で、srcType を srcType=http に設定します。大文字と小文字が区別されます。
- job.cfg の httpListFilePath には、c:/example/http.list 、 /root/example/http.list などの HTTP アドレステーブルファイルへの絶対パスを指定します。完全な HTTP リンクは 127.0.0.1/aa/bb.jpg です。分割メソッドが異なっていると、アップロード後に OSS 上のオブジェクトパスが異なる可能性があります。

```
http://127.0.0.1/aa/ bb.jpg # The first line
http://127.0.0.1/ aa/bb.jpg # The second line
```

最初の行が OSS にインポートされた後のオブジェクト名は destPrefix + bb.jpg で、2 行目のオブジェクト名は destPrefix + aa/bb.jpg です。httpPrefixColumn パラメーターは、ドメイン名の列を指定します。前述の 127.0.0.1/aa/ または 127.0.0.1/ など、最初の列が既定で適用されます。

relativePathColumn は、前述の bb.jpg や aa/bb.jpg など、OSS 内のオブジェクト名を指定します。オブジェクトに複数の列がある場合、次のようになります。

```
http://127.0.0.1/aa/ bb/cc dd/ee ff.jpg
```

次のように設定を行う必要があります。: httpPrefixColumn=1 , relativePathColumn=4 。

- job.cfg 内の destAccessKey 、 destSecretKey 、 destDomain および destBucket の設定項目。
- taskObjectCountLimit : 各タスクの最大オブジェクト数。既定値は 10,000 です。
- taskObjectSizeLimit : 各タスクの最大データサイズ。マスターがタスクを分割している時、それぞれの HTTP オブジェクトがソースから取得したオブジェクトの大きさである場合、各オブジェクトには HTTP リクエストのオーバーヘッドが 1 つずつあるため、タスク割り当ての効率に悪影響を及ぼし、タスクの並行処理および移行の効率性が低下します。そのため、このパラメーターは HTTP データの移行には無効です。
- Domain name : httpListFilePath によって指定されたオブジェクトの最初の列。同じドメイン名を持つ連続ジョブは taskObjectCountLimit パラメーターに従って分割され、異なるドメイン名を持つ連続ジョブは異なるタスクに分割されることで、接続をより効率的に再利用できるようになります。例:

```
http://mingdi-hz.oss-cn-hangzhou.aliyuncs.com/ import/test1.txt
http://mingdi-hz.oss-cn-hangzhou.aliyuncs.com/ import/test2.txt
http://mingdi-bj.oss-cn-beijing.aliyuncs.com/ import/test3.txt
http://mingdi-bj.oss-cn-beijing.aliyuncs.com/ import/test4.txt
```

`taskObjectCountLimit` `taskObjectCountLimit` 値が 2 より大きい場合、ジョブは 2 つのタスクに分割されますが、以下の条件では、ジョブは 4 つのタスクに分割されます。

```
http://mingdi-hz.oss-cn-hangzhou.aliyuncs.com/ import/test1.txt
http://mingdi-bj.oss-cn-beijing.aliyuncs.com/ import/test3.txt
http://mingdi-hz.oss-cn-hangzhou.aliyuncs.com/ import/test2.txt
http://mingdi-bj.oss-cn-beijing.aliyuncs.com/ import/test4.txt
```

そのため、`httpListFilePath` で指定された HTTP アドレス一覧オブジェクトは、まずドメイン名でソートされます。

ネットワークトラフィックおよびパラメーターの設定

次のパラメーターの設定はネットワークトラフィックに関連しています。

- `sys.properties` では、`workerTaskThreadNum` パラメーターはワーカーによる並行処理のジョブ数を示します。ネットワーク品質が悪い場合、または同時実行性が高い場合は、多数のタイムアウトエラーが発生する可能性があります。この時点で、同時実行性を低くし、設定項目を変更してサービスを再開することを推奨します。
- `sys.properties` では、`workerMaxThroughput (KB / s)` パラメーターはワーカーのトラフィック上限を示します。ソース側の帯域幅調整やネットワークの制限外トラフィックの制限など、トラフィックを制限する場合は、このパラメーターの値をマシンで許可されている最大ネットワークトラフィックよりも小さくし、ビジネス要件に基づいて評価する必要があります。
- `job.cfg` の `taskObjectCountLimit` パラメーターは、各タスクのオブジェクトの最大数を示します。既定値は 10,000 です。このパラメーターは、タスクの数に影響します。タスクの数が少なすぎると、並行タスクの効率性が低下する可能性があります。
- `job.cfg` の `taskObjectSizeLimit` は、各タスクの最大データサイズを示します。既定値は 1 GB です。このパラメーターは、タスクの数に影響します。タスクの数が少なすぎると、並行タスクの効率が低下する可能性があります。

② 説明

- 移行を開始する前に、設定ファイルのパラメーターを決定することを推奨します。
- `sys.properties` 内のパラメーターに対する変更は、移行サーバーを再起動した後に有効になります。
- `job.cfg` ジョブが送信された後は、ジョブの設定パラメーターを変更することはできません。

4.5. よくある質問

- 1. `UnsupportedClassVersionError`

Exception Executing コマンド

```
Exception in thread "main" java.lang.UnsupportedClassVersionError: com/aliyun/ossimport2/OSSImport2 : Unsupported major.minor version 51.0
at java.lang.ClassLoader.defineClass1(Native Method)
at java.lang.ClassLoader.defineClassCond(ClassLoader.java:631)
at java.lang.ClassLoader.defineClass(ClassLoader.java:615)
at com.simontuffs.onejar.JarClassLoader.defineClass(JarClassLoader.java:693)
at com.simontuffs.onejar.JarClassLoader.findClass(JarClassLoader.java:599)
at java.lang.ClassLoader.loadClass(ClassLoader.java:306)
at java.lang.ClassLoader.loadClass(ClassLoader.java:247)
at com.simontuffs.onejar.Boot.run(Boot.java:300)
at com.simontuffs.onejar.Boot.main(Boot.java:159)
```

原因: Java のバージョンが古すぎるため、1.7 以降に更新できません。

- 2. InvocationTargetException

submit コマンドを使用して、タスク報告の例外を送信します。

```
Exception in thread "main" java.lang.reflect.InvocationTargetException
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:497)
at com.simontuffs.onejar.Boot.run(Boot.java:306)
at com.simontuffs.onejar.Boot.main(Boot.java:159)
Caused by: java.lang.NullPointerException
at com.aliyun.ossimport2.config.JobConfig.load(JobConfig.java:44)
at com.aliyun.ossimport2.OSSImport2.doSubmitJob(OSSImport2.java:289)
at com.aliyun.ossimport2.OSSImport2.main(OSSImport2.java:120)
... 6 more
```

理由: 設定ファイル内の項目が削除またはコメントアウトされているかどうかを確認し、等号の後に、設定する必要がない項目、および削除する必要がない項目を入力します。

- 3. 開かれるファイルが多すぎます。

理由: `ulimit -n` 表示システムハンドル。

- 値が 10,000 未満の場合は、`ulimit -n 65536` を使用してプロセスを再開できます。
- 比較的大規模に設定されている場合は、`sudo losf -n` を使用して、どのプロセスがハンドルを開いたかをトラブルシューティングします。

- 4. Windows が起動後、Windows が数秒後に戻る

原因: ほとんどの場合、Java がインストールされていないか、バージョンが 1.7 未満であるか、構成ファイルのエラーが原因です。

- 5. ジョブが実行しない、もしくはジョブが終了しません。

submit コマンドがタスクを完了したら、stat を使用します。タスク状態は常に表示されます。

```
bash console.sh stat
[WARN] List files dir not exist : /home/<user>/ossimport/workdir/master/jobs/
no jobs is running or finished.
```

理由:

- ジョブが実行依頼されたばかりです。タスクが実際に生成および配信されておらず、ログの出力が正常な場合は、マスターは最初にファイルの一覧をスキャンする必要があります。
- 長時間経過しても、通常は start コマンドがなくエラーが出力されたままになります。プロセスを開始するか、プロセスの開始後に予期せず終了するように指示します。サービスを開始させない場合は、start を使用するだけです。それ以外の場合は、logs/ossimport.log を調べて例外の原因を見つけ、それを解決してからサービスプロセスを開始します。
- 6. STAT コマンドは常に "scanfinished: false" を表示します。

タスクの総数が増えているかどうかを確認します。

- まだ処理中のものがある場合、ジョブのファイルの一覧は完全ではありません。一覧には新しいファイルもあります。
- ジョブが増分モードに設定されている場合、常に変更されないため、scanfinishedは "true" にはなりません。ユーザーが設定した間隔に応じてファイルの一覧を定期的にスキャンするには、新規または変更されたドキュメントを調べます。
- 増分モードではない場合、タスク数は増えず、ログで例外がチェックされます。
- 7. サービスプロセスは削除されましたが、ログに例外は出力されませんでした。

理由: マシンの使用可能なメモリが 2 GB 未満の場合、考えられる可能性は、削除に必要なだけのメモリが不足しているからです。dmsg を確認してください。削除するためのメモリが足りないという記録があるかどうか、記録を取ってください。

- 8. プロセスがハングアップまたは強制終了された後、サービスを再開するために必要なことは何ですか。

start コマンドを直接呼び出します。このコマンドはサービスを開始します。送信されたジョブは、clean コマンドを呼び出さない限り、再送信する必要はありません。実行依頼されたすべてのジョブには、実行された作業をやり直さないブレイクポイントレコードがあります。

- 9. OSS コンソールがソースより少ない量のデータを表示するタスクを完了します。

ジョブがすべて正常にアップロードされた後、またはローカルで使用された後は、OSS コンソール内のバケットのサイズは変わりません。du 統計のサイズは大きく変わります。原因: OSS コンソールのバケットデータの量が更新は 1 時間遅れます。du コマンドは、実際のファイルよりも大きいブロックサイズをカウントします。次のコマンドを参照することで、ローカルディレクトリの実際のサイズをカウントできます。ls -lR <directory absolute path> | grep "\-rw" | awk '{sum+=\$5}END{print sum}'

- 10. stat コマンドで示される失敗したタスクはどのように処理するのですか。

通常は、retry コマンドを使用して再試行します。

- 11. タスクが失敗した後、繰り返しの再試行が成功しないファイルがあります。

理由: ファイル「`$work_dir/master/jobs/$jobName/failed_tasks/$taskName/error.list`」をご参照ください。失敗したファイルの相対パスを取得し、そのファイルにアクセス許可があるかどうか、削除されているか、柔軟性があるか、ファイル名が文字化けしていないかなどを確認します。

- 12. どのようにしたら OSS に間違ったファイル名のファイルをアップロードすることができますか。

ファイル名を確認した後、まず、`export LANG="<your file name encode>"` と `ls` を使用する必要があります。元のジョブを消去して `submit` コマンドでジョブを再実行依頼するためのコマンド。

- 13. `java.nio.file.AccessDeniedException`

例外報告: `ava.nio.file.AccessDeniedException` 原因: 構成ファイルのディレクトリにアクセス許可がありません。

- 14. タスク状態は 0 を表示しますが、ジョブは完了を表示します。

タスク状態は 0 を表示しますが、ジョブは次のように完了を表示します。

```
[2015-12-28 16:12:35] [INFO] JobName:dir_data
[2015-12-28 16:12:35] [INFO] Pending Task Count:0
[2015-12-28 16:12:35] [INFO] Dispatched Task Count:0
[2015-12-28 16:12:35] [INFO] Succeed Task Count:0
[2015-12-28 16:12:35] [INFO] Failed Task Count:0
[2015-12-28 16:12:35] [INFO] Is Scan Finished:true
[2015-12-28 16:12:35] [INFO] JobState:SUCCEED
```

理由:

- `srcPrefix` はエラーに対処し、`List` がファイルから出てこないようにします。
 - ディレクトリ概念は OSS によってシミュレートされるため、`srcPrefix` にはディレクトリのみが存在し、ファイルは存在しません。ファイルが本当にアップロードされることはありません。
- 15. アクセスしようとしているバケットは、指定されたエンドポイントを使用して対処する必要があります。

ログの例外報告:

```
Exception:com.aliyun.oss.OSSException: The bucket you are attempting to access must be addressed using the specified endpoint. Please send all future requests to this endpoint.
```

```
<Error>
```

```
<Code>AccessDenied</Code>
```

```
<Message>The bucket you are attempting to access must be addressed using the specified endpoint. Please send all future requests to this endpoint.</Message>
```

```
<RequestId>56EA98DE815804**21B23EE6</RequestId>
```

```
<HostId>my-oss-bucket.oss-cn-qingdao.aliyuncs.com</HostId>
```

```
<Bucket>my-oss-bucket</Bucket>
```

```
<Endpoint>oss-cn-hangzhou.aliyuncs.com</Endpoint>
```

```
</Error>
```

理由: バケットの `srcDomain` または `destDomain` がエラーです。ドメイン名の一覧に従ってください。正しいドメイン名を入力します。

- 16. 計算したリクエストの署名が、指定した署名と一致しません。

ログの例外報告:

```
Exception:com.aliyun.oss.OSSException: The request signature we calculated does not match the signature you provided. Check your key and signing method.
[ErrorCode]: SignatureDoesNotMatch
[RequestId]: xxxxxxxx
[HostId]: xxx.oss-cn-shanghai.aliyuncs.com
```

理由: `destAccessKey` と `destSecretKey` およびスキナが間違っていないかを確認します。Access control をご参照ください。

- 17. InvocationTargetException

送信コマンドのタスク時間送信の例外:

```
submit job:/disk2/ossimport2/local_job.cfg
Exception in thread "main" java.lang.reflect.InvocationTargetException
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:606)
at com.simontuffs.onejar.Boot.run(Boot.java:306)
at com.simontuffs.onejar.Boot.main(Boot.java:159)
Caused by: java.lang.NullPointerException
at com.aliyun.ossimport2.OSSImport2.doSubmitJob(OSSImport2.java:289)
at com.aliyun.ossimport2.OSSImport2.main(OSSImport2.java:120)
... 6 more
```

理由: `conf/sys.properties` 内の "Configuration item workingdir" を確認します。設定するかどうか、正しく設定するかどうか、および設定ファイルパスが正しいパスであることを検証します。

- 18. エージェントの設定をサポートしていますか。

この機能はサポート対象外です。

- 19. OSS を OSS に移行させるのにはなぜ費用がかかるのでしょうか。

「[エンドポイント](#)」をご参照ください。ヘルプ内のドメイン名は、内部ネットワークドメイン名を設定した後、トラフィック料金は請求されませんが、アクセスに対するコストが発生しています。

- 20. 同期プロセスがソースファイルが存在しないことを示しています。

理由: マスターは最初にファイルリストを一覧表示し、次にファイル一覧に従ってデータを移動させます。一覧表示が終了すると、ソース側の特定のファイルが削除され、ソースファイルが存在しないことが確認できます。このタイプのファイルはスキップされ、エラーリストに出力されます。

- 21. 増分モードがオンになっていると、OSS はローカルで削除された後に削除されますか。

ローカルでの削除後に OSS が削除されると、増分モードがオンになり、削除操作は同期されません。

- 22. 増分モードがオンになっていると、新しいドキュメントに同期されないものがあります。

理由: 増分モードでは、コントラストファイルの最後の変更を使用して、ファイルが増分かどうかを判断します。ファイルシステムの操作によっては、Windows では cp や mv、Linux では -t や -a オプションを付けた mv や rsync など、オブジェクトの最終更新日時を変更できないものがあります。これらの操作によるデータ変更は検出されず、OSS にも同期されません。

- 23. 移行したタスクの数が常に 0 を示しています。

理由: 主に 2 つの状況に分けられます。

```
[2016-07-21 10:21:46] [INFO] [name=YoupaiList, totalRequest=1729925, avgLatency=38, recentLatency=300000]
```

recentLatency=30000 であれば、このログは通常、正常です。一覧表示および一覧表示を何度も行う際の表示スピードは遅く、通常タイムアウトの 30 秒がかかります。タスクがゆっくりと一覧表示される場合など、いくつかのファイルを一覧表示するのに 30 秒ほど時間がかかる場合があります (これは正常です)。

- recentLatency は非常に低いので、一般的なケースではアカウントのパスワードが間違っていることなどが考えられます。これは、SDK の別のエラーでは null だけが返されるためです。エラーの結果は返されないため、パッケージを取得することによって返される別のエラーコードを取得することしかできません。
- 24. 移行中に srcAccessKey と srcSecretKey が再度入力される内容は?
オペレーターのアカウント番号とパスワードを入力します。
- 25. 別のショットの移行中、HTTP がいつもエラー 429 を表示します。
SDK のアクセス間隔を制限するためにショットしたのも、アクセスが少しでも速い場合は速度が制限されます。カスタマーサービスリリースの制限については、チケットを起票し、サポートセンターへお問い合わせください Ossimport がこの状況を再試行します。
- 26. 不明なコマンド "java"、不明なコマンド "nohup" などの実行。
理由: 使用されているコマンドがインストールされていません。yum、apt-get、または zypper のいずれかを使用します。対応するコマンドがインストールされるまで待機します。
- 27. タスクが構成ファイルと一致しません。
ジョブ構成ファイルは正しいように見えますが、実際の実行は、ジョブプロファイル構成とはかなり異なります。sys.properties プロパティのみ変更を適用し、有効にするために再起動します。ジョブの設定ファイルが送信されても変更は有効になりません。元のジョブを削除し、新しい設定ファイルを再送信します。
- 28. バケット名 "xxx / xx" が無効です。

ログの例外報告:

```
java.lang.IllegalArgumentException: The bucket name "xxx/xx" is invalid. A bucket name must: 1) be comprised of lower-case characters, numbers or dash(-); 2) start with lower case or numbers; 3) be between 3-63 characters long.
```

理由: `destBucket` の設定項目が正しく入力されていること、およびバケットが "/" を使用していないこと、他のパスを使っていないことを確認します。

- 29. `com.aliyun.oss.ClientException: Unknown`

ログの例外報告:

```
com.aliyun.oss.ClientException: Unknown
[ErrorCode]: NonRepeatableRequest
[RequestId]: Cannot retry request with a non-repeatable request entity. The cause lists the reason the original request failed.
```

通常、ネットワークが一杯になったときに、ossimport は再試行します。再試行が失敗した場合は、タスクの完了後に再試行することができます。retry コマンドを再試行します。

- 30. `xxx.oss-cn-beijing-internal.aliyuncs.com` に接続して 80 タイムアウトになりました。

ログの例外報告:

```
Unable to execute HTTP request: Connect to xxx.oss-cn-beijing-internal.aliyuncs.com:80 timed out
[ErrorCode]: ConnectionTimeout
[RequestId]: Unknown
```

理由: ECS 以外のマシンは内部ドメイン名を使用できません。

- 31. 指定されたバケットは無効です。

ログの例外報告:

```
com.aliyun.oss.OSSEException: The specified bucket is not valid.
[ErrorCode]: InvalidBucketName
[RequestId]: 57906B4DD0EBAB0FF553D661
[HostId]: you-bucket.you-bucketoss-cn-hangzhou-internal.aliyuncs.com
```

理由: 設定ファイル `destDomian` から設定したドメイン名では、バケット名を付けることができません。

- 32. 設定ファイルの `srcPrefix` でファイルを個別に指定できますか。

いいえ。 `srcPrefix` はディレクトリまたはプレフィックスレベルのみをサポートします。1 つのファイルのアップロードであれば、他のよりシンプルなツールにて実行できます。

- 33. HTTP リクエストを実行できません: "... " 間の違いが大きすぎます。

ログの例外報告:

```
Unable to execute HTTP request: The Difference between the request time and the current time is too large.
```

```
[ErrorCode]: RequestTimeTooSkewed
```

```
[RequestId]: xxxxxxxx
```

理由:

- 大抵、ローカルマシンタイムはサーバータイムと 15 分以上相違しており、好ましくありません。
- 特に CPU 使用率が高い場合、同時実行性が高すぎるため、同時実行中にアップロードが遅くなる可能性があります。

- 34. No route to host

No route to host のエラーがログに表示されます。ローカルのファイアウォールや iptables によるネットワークの中断が原因の可能性があります。

- 35. 不明な http 一覧ファイルの形式

指定された HTTP 一覧ファイルの形式が正しくないため、http モードのログを使用してエラーが表示されます。

- その理由の 1 つとして、ファイルが別のシステムからコピーされている可能性が挙げられます。 `mac2unix` または `doc2unix` コマンドを使用することで、ファイル形式を変換することができます。
- ファイル内に、2 列未満の行など、規則を満たさない行があります。

- 36. boject キー "/xxxxx.jpg" が無効です。

ログの例外報告:

```
Exception:java.lang.IllegalArgumentException: The boject key "/xxxxx.jpg" is invalid. An object name should be between 1 - 1023 bytes long when encoded as UTF-8 and cannot contain LF or CR or unsupported chars in XML1.0, and cannot begin with "/" or "\".
```

理由:

- `srcPrefix` がディレクトリであることと、オブジェクトの名前の末尾に ";" が付いていないことを確認します。
- `destPrefix` が "/" または "\" で始まっていることを確認します。

5.RAM Policy Editor

アドレス

RAM Policy Editor

使用方法

RAM 権限ポリシーは、複数のルールで構成されています。RAM Policy Editor を使用して、ルールを 1 つずつインターフェイスに追加または削除した後、ポリシーの JSON ファイルが自動的に生成されます。すべてのポリシールールを追加した後、JSON ファイルをコピーし、アクセス制御コンソールで、作成済み権限ポリシーの内容ボックスに貼り付けます。

詳細な操作については、「[権限ポリシーの作成](#)」をご参照ください。

RAM Policy Editor では、ルールごとに Effect、Actions、Resources、Conditions の各フィールドを設定する必要があります。

- Effect

このルールへのアクセスを許可するか拒否するかを指定します。

- Actions

リソースに対するアクションを指定します。1 つ以上のアクションを選択できます。通常、ワイルドカードアクションを使用すれば十分です。

- `oss:*` : すべての操作を許可
- `oss:Get*` : すべての読み取り操作を許可
- `oss:Put*` : すべての書き込み操作を許可

詳しくは、「[RAM Policy Editor README](#)」をご参照ください。

- Resources

アクセスを許可する OSS のリソースを指定します。複数のリソースを指定することができ、それぞれは次の形式で表されます。

- バケット: `my-bucket` (バケット内のオブジェクトに対する権限なし)
- バケット内のすべてのオブジェクト: `my-bucket/*` (ListObjects など、バケット自体に対する権限なし)
- バケット内のディレクトリ: `my-bucket/dir` (dir/ 下のオブジェクトに対する権限なし)
- バケット内のディレクトリにあるすべてのオブジェクト: `my-bucket/dir/*` (ListObjects など、dir に対する権限なし)
- 完全なリソースパス: `acs:oss:*:1234:my-bucket/dir`、`1234` はユーザー ID です (コンソールに表示されます)。

EnablePath

ディレクトリに対する権限を付与したい場合は、通常、その上位レベルのディレクトリに対するリスト権限を付与する必要があります。たとえば、`my-bucket/users/dir/*` に読み取りおよび書き込み権限を付与したい場合、このディレクトリをコンソール (または他のツール) で表示するため、次の権限も付与する必要があります。

```
ListObjects my-bucket
ListObjects my-bucket/users
ListObjects my-bucket/users/dir
```

EnablePath オプションが選択されていると、前述の権限が自動的に追加されます。

- Conditions

アクセスを許可する場合に満たす必要のある条件を指定します。複数の条件を指定できます。

詳しくは、「[RAM Policy Editor README](#)」をご参照ください。

例

`my-bucket` と `my-bucket` のファイルにすべての権限を付与します。

RAM Policy Editor v1.1.0

[Chinese](#)[Star](#)

2

Add Rules

Effect

Actions

Resources

- After entering a resource name , **press Enter** for confirmation
- Example: my-bucket, my-bucket/dir/*
- [More...](#)

EnablePath ☐ Automatically add the parent directory permissions ?

Conditions (Optional)

[Generate auth policy](#)

Authorization Policy

```
{
  "Version": "1",
  "Statement": []
}
```

List of Rules

Effect	Actions	Resources	Conditions
--------	---------	-----------	------------

他の例については、「[RAM Policy Editor README](#)」をご参照ください。

6.ossftp

6.1. OSS FTP のクイックインストール

はじめに

OSS FTP は、一般的な FTP 要求を受信したときにファイルやフォルダに対する操作を OSS インスタンスにマッピングする、特殊な FTP サーバーです。このユーティリティを使用することで、FTP プロトコルを使用し、OSS インスタンスに保存されているファイルを管理できます。

❓ 説明 OSS SDK は本番環境向けに設計されており、OSS FTP は主に個人ユーザー向けです。

- 主な機能
 - クロスプラットフォーム: Windows、Linux、Mac の 32 ビットおよび 64 ビットの各オペレーティングシステムに対応しており、グラフィックとコマンドラインの両方のインターフェイスで使用できます。
 - インストール不要: このユーティリティは展開後にそのまま実行できます。
 - 設定不要: さらに設定を行うことなく、ユーティリティを実行できます。
 - 透明性: この FTP ユーティリティは Python で記述されているので、すべてのソースコードを見ることができます。近日中に、GitHub でオープンソースを公開する予定です。
- 主な機能
 - ファイルやフォルダのアップロード、ダウンロード、削除などの操作をサポート
 - 大きなファイルのマルチパートアップロードをサポート
 - ほとんどの FTP コマンドをサポートし、日常のニーズを満たします

❓ 説明

- 現在、インストールと展開を容易にするために、OSS FTP V1.0 は TLS 暗号化をサポートしていません。FTP プロトコルはプレーンテキスト転送を実装しています。パスワードの漏洩を防ぐために、FTP サーバーとクライアントを同じマシン上で実行し、127.0.0.1:por を使用してアクセスすることを推奨します。
- このユーティリティは、名前の変更および移動操作をサポートしていません。
- インストールパッケージの extract-to パスには漢字を含めないでください。
- FTP サーバーの管理コントロールページは、初期の IE ブラウザーでは開けない可能性があります。
- サポートされている Python のバージョンは Python 2.6 と Python 2.7 です。

ダウンロード

- Windows: [ossftp-1.0.3-win.zip](#)

既定では、Python 2.7 は Windows にインストールされていません。インストールパッケージに含まれているので、インストールや設定の手間がかかることなく、展開後に使用できます。

- Linux/Mac: [ossftp-1.0.3-linux-mac.zip](#)

既定では、Python 2.7 または Python 2.6 は Linux および Mac システムにインストールされています。そのため、Linux および Mac 用のインストールパッケージには実行可能な Python プログラムは含まれておらず、関連する依存ライブラリのみが含まれています。

実行

まず、ダウンロードしたファイルを展開します。次に、環境条件に基づき、適切な実行モードを選択します。

- Windows: start.vbs をダブルクリックして実行します。
- Linux: ターミナルを起動し、実行します。

```
$ bash start.sh
```

- Mac: start.command をダブルクリックするか、またはターミナルで実行します。

```
$ bash start.command
```

既定では、上記のプロセスは、127.0.0.1 のポート 2048 をリッスンする FTP サーバーを開始します。また、FTP サーバーのステータスを簡単に制御できるように、プログラムは Web サーバーをアクティブにします。Web サーバーは 127.0.0.1 でポート 8192 をリッスンします。システムにグラフィックインターフェイスがある場合は、コントロールページが自動的に開きます。

FTP サーバーへの接続

FileZilla クライアントを使用して FTP サーバーに接続することを推奨します。ダウンロードしてインストールしたら、次のようにして FTP サーバーに接続します。

- ホスト: 127.0.0.1
- ログオンの種類: 通常
- ユーザー: access_key_id / bucket_name
- パスワード: access_key_secret

? 説明

- スラッシュ記号 (/) は、両方の項目が必須ではないことを意味します。たとえば、ユーザーは、`tSxyiUM3NKswPMEp/test-hz-jh-002` でもよいです。
- access_key_id および access_key_secret についての詳細は、「**OSS アクセス制御**」をご参照ください。

高度な使用方法

- コンソールページから ftpserver を管理
 - リスニングアドレスの変更

ネットワーク経由で ftpserver にアクセスする場合は、既定のアドレス 127.0.0.1 ではローカルアクセスしか許可されないため、リスニングアドレスを変更する必要があります。リスニングアドレスはイントラネット IP またはインターネット IP に変更できます。

- リスニングポートの変更

ftpserver のリスニングポートを変更します。1024 未満のポートには管理者のアクセス許可が必要なので、1024 を超えるポートを使用することを推奨します。

○ ログレベルの変更

ftpserver のログレベルを設定します。FTP サーバーのログは、`data/ossftp/` ディレクトリに出力されます。表示するには、コンソールページの [ログ] ボタンをクリックします。既定のログレベルは INFO になっており、ログにはほとんど情報が記録されません。より詳細なログ情報が必要な場合は、レベルを DEBUG に変更します。ログ出力を減らしたい場合は、ログレベルを WARNING または ERROR に設定します。

○ バケットエンドポイントの設定

既定では、ftpserver はバケットの位置情報を検索するので、後に続くリクエストに対応するリージョン (`oss-cn-hangzhou.aliyuncs.com` や `oss-cn-hangzhou.aliyuncs.com` など) に送信することができます。まず、ftpserver はイントラネットを介し、OSS インスタンスへのアクセスを試みます。たとえば、`test-bucket-a.oss-cn-hangzhou.aliyuncs.com` というバケットエンドポイントを設定した場合、`test-bucket-a` にアクセスすると、`oss-cn-hangzhou.aliyuncs.com` という名前のドメインに接続されます。

○ 表示言語の設定

`cn / en` を設定することで、FTP コントロールページの表示言語を中国語 / 英語に変更できます。

② 説明

- 変更を有効にするには、システムを再起動する必要があります。
- 上記の変更はすべて、実際には ftp ディレクトリの `config.json` ファイルに対する変更です。したがって、このファイルを直接変更することもできます。

● ftpserver を直接起動する (Linux / Mac)。

`web_server` のオーバーヘッドを回避するため、ossftp ディレクトリでのみ、`ftpserver.py` ファイルを実行できます。

```
$ python ossftp/ftpserver.py &
```

設定の変更方法は、上記の方法と同様です。

考えられる問題

- FTP サーバーへの接続時にエラーが発生した場合
- FTP サーバーを Linux 上で実行している場合、FileZilla を使用してサーバーに接続すると、以下のエラーが発生する可能性があります。

```
501 はパスをデコードできません (サーバーファイルシステムのエンコードは ANSI_X3.4-1968です)。
```

これは通常、ローカルの中国語のコードでエラーが発生したときに生成されます。`start.sh` を実行するターミナルに次のコマンドを入力します。その後、プログラムを再起動します。

```
$ export LC_ALL=en_US.UTF-8; export LANG="en_US.UTF-8"; locale
```


6.2. Discuz で OSS インスタンスにリモート添付ファイルを格納する方法

はじめに

Web サイトのリモート添付ファイル機能とは、アップロードされた添付ファイルを FTP 経由でリモートストレージサーバー (通常はリモート FTP サーバー) に直接保存することを指します。

現在、Discuz フォーラム、PHPWind フォーラム、および WordPress Web サイトにて、リモート添付ファイル機能をサポートしています。

このドキュメントでは、Discuz ベースのフォーラムからのリモート添付ファイルの保存について説明します。

準備

OSS アカウントを申請し、"public-read" のバケットを作成します。匿名アクセスを許可する必要があるため、許可を "public-read" に設定する必要があります。

手順

ここで使用している Discuz のバージョンは Discuz! X3.1 です。詳細な設定プロセスは以下の通りです。

1. Discuz Web サイトにログインし、管理インターフェイスに移動します。[グローバル]、[アップロード設定] の順にクリックします。
2. [リモート添付ファイル] をクリックし、機能を設定します。
3. 検証

機能をテストするため、フォーラムにファイルを投稿します。任意の掲示板で投稿内容を作成し、投稿する添付ファイルとしてイメージファイルをアップロードします。

[イメージ] を右クリックし、[新しいタブでイメージを開く] をクリックします。

6.3. PHPWind でリモート添付ファイルを OSS インスタンスに格納する方法

はじめに

Web サイトのリモート添付ファイル機能とは、アップロードされた添付ファイルを FTP 経由でリモートストレージサーバー (通常はリモート FTP サーバー) に直接保存することを指します。

現在、Discuz フォーラム、PHPWind フォーラム、および WordPress Web サイトにて、リモート添付ファイル機能をサポートしています。

このドキュメントは、PHPWind ベースのフォーラムからのリモート添付ファイルの保存方法を説明します。

準備

OSS アカウントを申請して、"public-read" のバケットを作成します。匿名アクセスを許可する必要があるため、許可を "public-read" に設定する必要があります。

手順

使用する PHPWind は PHPWind 8.7 です。設定プロセスは以下の通りです。

1. Web サイトにログインします。

管理インターフェイスに移動し、[グローバルアップロード設定のリモート添付] を選択します。 >>

2. 機能の設定

3. 検証

PHPWind では、ユーザーがテストボタンをクリックし、直接、機能をテストすることはできません。そのため、機能を検証するためには、イメージファイルを投稿する必要があります。

イメージを右クリックし、[新しいタブで画像を開く] をクリックします。すると、イメージが新しいタブに表示されます。

イメージの URL は、そのイメージが OSS のバケット "test-hz-jh-002" にアップロードされたことを示します。

6.4. WordPress でリモート添付ファイルを OSS インスタンスに格納する方法

はじめに

Web サイトのリモート添付ファイル機能とは、アップロードされた添付ファイルを FTP 経由でリモートストレージサーバー (通常はリモート FTP サーバー) に直接保存することを指します。

現在、Discuz フォーラム、PHPWind フォーラム、および WordPress Web サイトにて、リモート添付ファイル機能をサポートしています。

このドキュメントは、WordPress ベースのフォーラムからのリモート添付ファイルの保存方法を説明します。

準備

OSS アカウントを申請して、"public-read" のバケットを作成します。匿名アクセスを許可するため、許可を "public-read" に設定する必要があります。

手順

WordPress は固有のリモート添付ファイル機能をサポートしていませんが、サードパーティーのプラグインを使用することで、リモート添付を実装しています。使用する WordPress は WordPress 4.3.1 で、プラグインは Hacklog Remote Attachment です。具体的な設定プロセスは次のとおりです。

1. WordPress の Web サイトにログインし、[プラグインのインストール] をクリックします。"FTP" をキーワード検索し、*Hacklog Remote Attachment* のインストールを選択します。
2. 設定
 - OSS-FTP を実行する FTP サーバーのアドレスを設定します。通常、127.0.0.1 です。
 - "FTP サービスポート番号" を既定の "2048" に設定します。
 - "FTP アカウント" を AccessKeyID/BucketName の形式で設定します。ここで、"/" は "または" を意味しません。
 - "FTP パスワード" を AccessKeySecret に設定します。

② 説明 AccessKeyID と AccessKeySecret を取得するため、Alibaba Cloud コンソールにログインし、「アクセスキーの 管理」ページに移動します。

- FTP タイムアウトを既定値の 30 秒に設定します。
- "Remote Basic URL" を `http://BucketName.Endpoint/wp` に設定します。ここで、杭州リージョンのバケット "test-hz-jh-002" をテストします。したがって、`http://test-hz-jh-002.oss-cn-hangzhou.aliyuncs.com/wp` と入力します。
- FTP リモートパスを設定します。すべての添付ファイルをバケットの "wp" ディレクトリに保存するため、"wp" と入力します。このフィールドは "Remote Basic URL" フィールドに関連していません。
- "HTTP リモートパス" を "." に設定します。

詳しくは、下の図をご参照ください。

3. 検証

設定完了後、[保存] をクリックするとテストが自動的に開始されます。テスト結果はページ上部に表示されます。

4. 新しい記事を投稿してイメージを挿入します。

これで、新しい記事を書いてリモート添付機能をテストすることができます。記事を作成したら、[メディアの追加] をクリックし、添付ファイルをアップロードします。

5. 添付ファイルがアップロードされたら、[投稿] をクリックして記事を表示します。

6.5. ファイル共有のために RAM を統合する方法

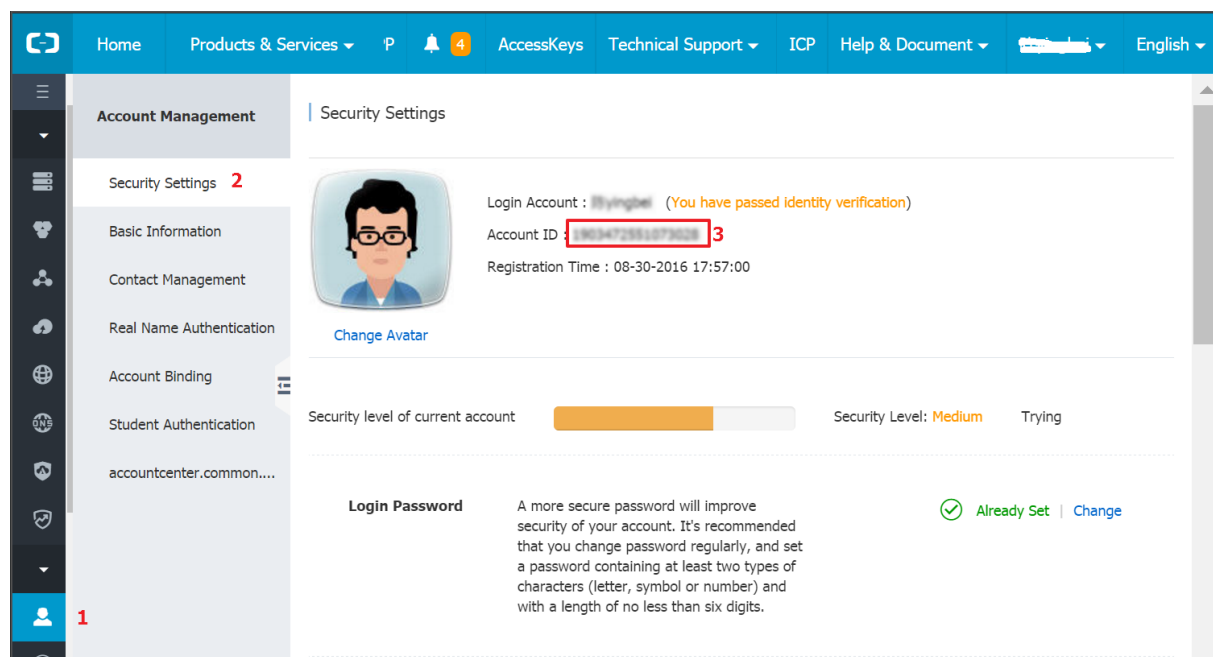
はじめに

このドキュメントでは、ユーザーバケット内のファイルとフォルダを共有するため、RAM サービスを統合する方法を説明します。バケット所有者はオブジェクトを編集できますが、その他のユーザーは読み取り専用権限のためオブジェクトの編集はできません。

プロセス: RAM の有効化 -> 読み取り専用権限付与ポリシーの作成 -> サブアカウントの作成 -> サブアカウントへの権限付与 -> FTP ログインの検証

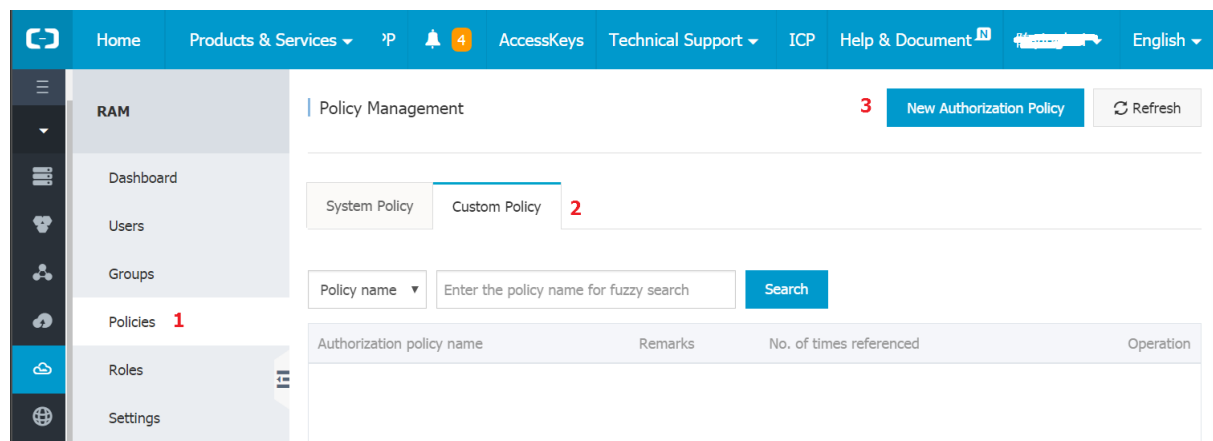
アカウント ID の取得

次の図に示すように、アカウント ID を取得します。



RAM をアクティブにする

Resource Access Management (RAM) は、リソースアクセスを制御するために設計された Alibaba Cloud のサービスです。ポリシーを作成することで、共有の読み取りアカウントを作成できます。ユーザーはこのアカウントを使用することで、FTP ツールにログインし、ファイルを読み取ることができます。



権限付与ポリシーの作成

RAM をアクティブにした後、RAM コンソールに移動し、左側の [ポリシー] をクリックします。次の図に示す手順に従って、新しい権限付与ポリシーを作成します。

次のように権限付与ポリシーを入力します。

Create Authorization Policy

STEP 1: Select an authorization policy

STEP 2: Edit permissions and submit

STEP 3: Policy created

* Authorization policy name :

1 read-list-test-hz-john-001

The name must be 1-128 characters long and can contain English letters, numbers, and "-"

Remarks :

2 read-list-test-hz-john-001

Policy content :

3

```
1 {
2   "Version": "1",
3   "Statement": [
4     {
5       "Action": [
6         "oss:GetObject",
7         "oss:HeadObject"
8       ],
9       "Resource": [
10        "acs:oss:*:1884083416482000:test-hz-john-001/*"
11      ],
12       "Effect": "Allow"
13     },
14     {
15       "Action": [
```

Authorization policy format definition

Authorization policy FAQs

4

Prev

New Authorization Policy

Cancel

ポリシー名と、必要に応じて補足 (フィールド 1 と 2) を指定します。フィールド 3 の [ポリシーの内容] により、ポリシーが決まります。

> Document Version:20200819

149

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "oss:GetObject",
        "oss:HeadObject"
      ],
      "Resource": [
        "acs:oss:*.*****:test-hz-john-001/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "oss:ListObjects",
        "oss:GetBucketAcl",
        "oss:GetBucketLocation"
      ],
      "Resource": [
        "acs:oss:*.*****:test-hz-john-001"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "oss:ListBuckets"
      ],
      "Resource": [
        "acs:oss:*.*****.*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

上記の例では、`*****` を自分のアカウント ID に、`test-hz-john-001` を自分のバケット名にそれぞれ置き換えます。次に、すべての内容をコピーし、[ポリシーの内容] に貼り付けます。最後に、[権限付与ポリシーを作成] をクリックします。

アカウントの作成

上記の権限付与ポリシーから、読み取り専用ポリシーが作成されます。次にアカウントを作成し、そのアカウントにこのポリシーを適用させます。次の手順に従ってアカウントを作成します。

❓ 説明 新しいアカウントの access_key は忘れないようにしてください。

アカウントの承認

新しいポリシーをアカウントに付与します。

サブアカウントでログインする

サブアカウントの access_key と権限付与ポリシーのバケットを使用し、ログインします。これで、ファイルとフォルダをダウンロードすることができます。(アップロード操作は失敗します)

6.6. よくある質問

- アクセス許可の問題:

- バケットを一覧表示できません。ログインできません。

大抵、使用されている accesskeyid に原因があります。また、accesskeysecret はサブアカウントに属しており、サブアカウントにはバケットを一覧表示する権限がありません。

バケットを一覧表示する権限がない場合、Endpoint 列により、バケットのアクセสดメイン名が設定されます。ossftp を介してバケットにアクセスすると ftpserver はサービスを介し、バケットのリージョンの取得を試みます。この時、ユーザーアカウントがリストにない場合、バケット権限によりログインが失敗します。

解決策はバケット内にあります。エンドポイントで設定されている第 3 レベルアクセสดメイン名等。

- ログインが成功した後、リストファイルが間違っ報告されます。

大抵、使用されている accesskeyid に原因があります。accesskeysecret はサブアカウントに属しており、サブアカウントには List Object (get bucket と同等) 権限がありません。

- その他の質問

- リストファイルタイムアウトにより、ログインが成功した後に接続が切断されます。

通常、バケットのルートディレクトリ内のファイル、またはフォルダが多すぎることが原因です。FTP にログイン後、ftpserver はバケットのルートディレクトリ内のすべてのファイルとフォルダを一覧表示しようとします。ルートディレクトリ内のファイルは一度に 1000 のファイルとフォルダを一覧表示することができます。しかし、ルートディレクトリ内に 100 万を超えるファイルとフォルダがあると、1000 を超える HTTP 要求が発生してしまい、タイムアウトしやすくなります。

- ftpserver を実行しているマシンがポート制限のため、データ転送に失敗します。

FTP プロトコルの制御ポートはデータポートとは異なるため、ftpserver がパッシブモードで動作している場合、データを転送する必要があるときはいつでも ftpserver はランダムなポートを 1 つ開き、クライアントが接続するのを待機します。そのため、ftpserver のマシンにポート制限があると、データが正しく転送されない可能性があります。

回避策は、ftpserver を実行しているときにあります。ローカルポートの開始範囲と終了範囲を設定するため、"-- Passive_ports_start" および "-- Passive_ports_end" パラメーターを指定し、その範囲のポートを開きます。

- クライアントと ftpserver 間の接続が頻繁に切断されます。

通常、各 FTP クライアントにはタイムアウト設定があり、タイムアウトしないよう設定することができます。たとえば、[設定] -> [接続] で filezilla ツールを使うことにより、タイムアウトを 0 に設定することができます。

7.ossfs

7.1. クイックインストール

ossfs を使用すると、Alibaba Cloud OSS バケットを Linux システムのローカルファイルにマウントできます。バケットをマウントしたシステムでは、ローカルファイルシステムを使用して OSS オブジェクトに対する操作をすばやく実行し、データの共有を実現できます。

主な機能

ossfs は S3FS に基づいて構築されており、すべての S3FS の機能が組み込まれています。主な機能は次のとおりです。

- ファイルの読み書き、ディレクトリ、リンク操作、パーミッション、UID / GID、拡張属性など、POSIX ファイルシステムのほとんどの機能をサポートします。
- OSS マルチパート機能を使用した、大きなファイルのアップロード
- MD5 チェックサムによるデータの完全性の保証

制限事項

ローカルファイルシステムと比較すると、ossfs が提供する機能とパフォーマンスには一定の制限があります。制限事項は次のとおりです。

- ランダム書き込みまたは追加操作によって、ファイル全体の書き換えが発生します。
- システムが OSS サーバーにリモートからアクセスする必要があるため、ディレクトリの一覧表示などのメタデータ操作のパフォーマンスが低下します。
- ファイルとフォルダーの名前変更操作はアトミックではありません。
- 複数のクライアントが 1 つの OSS バケットに接続されている場合は、各クライアントの動作を手動で調整する必要があります。たとえば、複数のクライアントが同じファイルに書き込まないように調整する必要があります。
- ハードリンクはサポートされていません。
- ossfs を使用したシステムでは、システム負荷が大幅に増加するため、高並列の読み書きシナリオには適していません。

インストールと使用

- インストールパッケージのダウンロード

Linux のバージョン	ダウンロードリンク
Ubuntu 16.04 (x64)	ossfs_1.80.5_ubuntu16.04_amd64.deb
Ubuntu 14.04 (x64)	ossfs_1.80.5_ubuntu14.04_amd64.deb
CentOS 7.0 (x64)	ossfs_1.80.5_centos7.0_x86_64.rpm
CentOS 6.5 (x64)	ossfs_1.80.5_centos6.5_x86_64.rpm

Linux ディストリビューションのバージョンが低いため、カーネルのバージョンは比較的低くなっています。実行中に ossfs が切断されたり、その他の問題が発生しやすい傾向があります。そのため、オペレーティングシステムを CentOS 7.0 または Ubuntu 14.04 以降にアップグレードすることを推奨します。

- インストール方法

- 次のコマンドを実行して、Ubuntu の ossfs をインストールします。

```
sudo apt-get update
sudo apt-get install gdebi-core
sudo gdebi your_ossfs_package
```

- 次のコマンドを実行して、CentOS 6.5 以降の ossfs をインストールします。

```
sudo yum localinstall your_ossfs_package
```

- 次のコマンドを実行して、CentOS 5 の ossfs をインストールします。

```
sudo yum localinstall your_ossfs_package --nogpgcheck
```

- 使用方法

バケット名と AccessKeyId/Secret を設定し、/etc/passwd-ossfs ファイルに保存します。このファイルへのアクセス許可は正しく設定されている必要がありますのでご注意ください。640 に設定することを推奨します。

```
echo my-bucket:my-access-key-id:my-access-key-secret > /etc/passwd-ossfs
chmod 640 /etc/passwd-ossfs
```

指定したディレクトリに OSS バケットをマウントします。

```
ossfs my-bucket my-mount-point -ourl=my-oss-endpoint
```

例

my-bucket バケットを /tmp/ossfs にマウントします。AccessKeyId は faint で、AccessKeySecret は 123 、OSS エンドポイントは http://oss-cn-hangzhou.aliyuncs.com です。

```
echo my-bucket:faint:123 > /etc/passwd-ossfs
chmod 640 /etc/passwd-ossfs
mkdir /tmp/ossfs
ossfs my-bucket /tmp/ossfs -ourl=http://oss-cn-hangzhou.aliyuncs.com
```

 **注意** Alibaba Cloud ECS インスタンスを使用して、ossfs サービスを提供する場合は、イントラネットエンドポイントを使用します。この例では、OSS エンドポイントを `oss-cn-hangzhou-internal.aliyuncs.com` に置き換えることで、帯域幅のコストを節約できます。イントラネットエンドポイントの詳細については、「[リージョンとエンドポイント](#)」をご参照ください。

バケットのマウントを解除するには以下のコマンドを実行します。

```
fusermount -u /tmp/ossfs
```

詳しくは、『[GitHub ossfs](#)』をご参照ください。

リリースログ

詳しくは、『[GitHub ChangeLog](#)』をご参照ください。

7.2. よくある質問

- 質問: ossfs はどのプログラムに適していますか。
 - ossfs は OSS バケットをローカルにマウントします。OSS をサポートしていないプログラムでデータを OSS に自動的に同期させるには、ossfs が最適な選択肢です。
- 質問: ossfs にはどのような制限がありますか。
 - データはネットワークを介してクラウドに同期させる必要があるため、ossfs のパフォーマンスと機能はローカルファイルシステムのものとは異なる場合があります。マウントされた ossfs ディスクで頻繁に I/O 操作を行ってデータベースまたは他のアプリケーションを実行したい場合は、これを慎重に検討する必要があります。ossfs は、次の点でローカルファイルシステムと異なります。
 - システムは OSS サーバーにリモートからアクセスする必要があるため、リストディレクトリなどのメタデータ操作のパフォーマンスは低下します。
 - ファイル / フォルダの名前変更操作は、アトミックではありません。
 - 複数のクライアントが 1 つの OSS バケットに接続されている場合は、各クライアントの動作を手動で調整する必要があります。たとえば、複数のクライアントが同じファイルに書き込まないようにする必要があります。
 - ハードリンクはサポートされていません。
- 質問: ossfs に Alibaba Cloud ホストを使用する必要はありますか。
 - ossfs は Alibaba Cloud イン트라ネットと組み合わせて使用する必要はありません。外部のインターネットホストでも使用できます。
- 質問: ossfs は同時に複数の OSS バケットをマウントできますか。
 - はい、passwd-ossfs ファイルに複数の OSS 構成情報エントリを書き込みます。異なる OSS アカウントからのバケットがサポートされています。
- 質問: ossfs を yum/apt-get にインストールしたところ、"conflicts with file from package fuse-devel" という内容のエラーが発生しました。
 - ユーザーのシステムには旧バージョンの fuse があります。関連するパッケージマネージャーを使用して ossfs をアンインストールした後、再インストールしてください。
- 質問: ossfs が正しく動作していません。どのようにデバッグすればいいでしょうか。

- マウント時に `-d -o f2` パラメーターを使用できます。ossfs はログの内容をシステムログに書き込みます。centos システムでは、`in/var/log/messages` にあります。
- マウント時に `-f -d -o f2` パラメーターを使用することもでき、ossfs はログを画面に出力します。
- 質問: バケットをマウントしようとすると、"`ossfs: unable to access MOUNTPOINT /tmp/ossfs: Transport endpoint is not connected`" というエラーが表示されるのはなぜですか。
 - まず、対応するディレクトリに対して `umount` コマンドを実行します。
 - ossfs でマウントするときは、入力した URL パラメーターが正しいこと、およびバケット、AccessKey ID、および AccessKey secret が一致していることを確認します。
 - URL にはバケット名を含めないでください。たとえば、OSS コンソールでバケットドメイン名が `ossfs-test-1.oss-cn-hangzhou.aliyuncs.com` の場合は、URL を `http://oss-cn-hangzhou.aliyuncs.com` に設定します。
- 質問: ossfs に "`ossfs: unable to access MOUNTPOINT /tmp/odat: No such file or directory`" と表示されるのはなぜですか。
 - このエラーは、ディレクトリがまだ作成されていない場合に発生します。マウントする前にディレクトリを作成する必要があります。
- 質問: バケットをローカルにマウントし、そのディレクトリに対して `ls` コマンドを実行した後に "`operation not permitted`" エラーが発生するのはなぜですか。
 - バケット内に、ディレクトリ名が見えない文字を含む OSS オブジェクトを含んでいないかどうか確認してください。ファイルシステムには、ファイル / ディレクトリ名に対する厳しい制限があります。ディレクトリ名が制限を満たさない場合、このエラーが発生します。別のツールを使用してこれらのオブジェクトの名前を変更して `ls` コマンドを実行すると、ディレクトリの内容が正しく表示されるようになります。
- 質問: 1つのディレクトリにたくさんのファイルがあります。なぜこれほど遅いのでしょうか。
 - ディレクトリ内に `n` 個のファイルがあると仮定すると、少なくともこのディレクトリの `ls` コマンドには、最低限の `n` 個の oss http リクエストが必要です。多数のファイルがある場合、重大なパフォーマンス上の問題を引き起こす可能性があります。
 - 2つの方法で最適化することができます。
 - "`-omax_stat_cache_size = xxx`" パラメーターを使用して統計キャッシュのサイズを増やしてください。これにより、ファイルのメタデータがローカルキャッシュにあるため、最初の `ls` コマンドは遅くなりますが、その後の `ls` コマンドは速くなります。既定値は 1000 で、約 4 MB のメモリが必要です。ユーザーのマシンのメモリのサイズに応じて適切な値に調整します。
 - `ls -f` コマンドを使用して、OSS による `n` 個の HTTP リクエストを排除します。
- 質問: ossfs のマウント中にアクセス権を設定するにはどうすればよいですか。
 - 他のユーザーにマウントされたフォルダーへのアクセスを許可する場合は、ossfs の実行時に次のように `allow_other` パラメーターを指定します。


```
ossfs your_bucket your_mount_point -ourl=your_endpoint -o allow_other
```
 - "allow_other" パラメーターがファイルにアクセスできないのはなぜですか。
 - 注記: "allow_other" は、マウントディレクトリ内の他のユーザーに付与されている許可であり、内部のファイルではありません。フォルダー内のファイルを変更したい場合は、`chmod` コマンドを使用します。

- "allow_other" は既定では、マウントディレクトリ 777 のアクセス権を与えますが、マウントディレクトリのアクセス権 770 が欲しい場合はどうすればよいですか。
 - umask によって設定することができます。
- 質問: 他のユーザーのフォルダー (/tmp/ossfs) のマウントを許可したいのですが、どうすればよいですか。
 - 方法 1: 他のユーザーに属するフォルダー (/tmp/ossfs) のマウントを許可したい場合は、ユーザーとしてマウントフォルダーを作成し、ossfs を使用する必要があります。
 - `sudo -u user mkdir /tmp/ossfs`
 - `sudo -u user ossfs bucket-name /tmp/ossfs`
 - 方法 2: 最初に、id コマンドによって指定されたユーザーの uid/gid 情報を取得します。たとえば、www ユーザーの uid / gid 情報を取得するには、次のように入力します。次にマウント時に uid / gid パラメーターを指定します。

```
ossfs your_bucket your_mountpoint -ourl=your_url -oid=your_uid  
-ogid=your_gid
```

注記: uid / gid は数字です。

- 質問: root ユーザーではない場合、アンマウントの ossfs はどのようにディレクトリをマウントできますか。
 - `fusermount -u your_mountpoint`
- 質問: デバイス起動時に ossfs を自動的にマウントするにはどうすればよいですか。
 - ステップ1: バケット名、AccessKeyId / Secret、およびその他の情報を /etc/passwd-ossfs に書き込み、このファイルのアクセス許可を "640" に変更します。

```
echo your_bucket_name:your_access_key_id:your_access_key_secret >  
/etc/passwd-ossfs
```

- `chmod 640 /etc/passwd-ossfs`

- ステップ 2: 適切な設定を行います (設定方法は各システムのバージョンにより異なります)。

- ステップ 2A: fstab メソッドを使用して、ossfs を自動的にマウントします (Ubuntu 14.04 と CentOS 6.5 に適用されます)。

- /etc/fstab に次のコマンドを追加します。

```
ossfs#your_bucket_name your_mount_point fuse _netdev,url=your_url,allow_other 0
0
```

- 上記のコマンドで、"your_xxx" を実際のバケット名、およびその他の情報に置き換えます。
- /etc/fstab ファイルを保存します。 `mount -a` コマンドを実行します。エラーが報告されなければ、正しく設定されています。
- これで、Ubuntu 14.04 は自動的に ossfs をマウントすることができます。CentOS 6.5 の場合は、次のコマンドも実行します。
- `chkconfig netfs on`

- ステップ 2B: ブートスクリプトを使用して、ossfs をマウントします (CentOS 7.0 以降に適用されます)。

- /etc/init.d/ ディレクトリにファイル ossfs を作成します。「**テンプレートファイル**」の内容を新しいファイルにコピーします。ここで、"your_xxx" を自分の情報に置き換えます。
- コマンド `chmod a+x /etc/init.d/ossfs` を実行します。
- 上記のコマンドは、新しい ossfs スクリプトに実行権限を付与します。これでこのスクリプトを実行できます。スクリプトの内容にエラーが発生しなければ、OSS バケットは指定されたディレクトリにマウントされています。
- コマンド `chkconfig ossfs on` を実行します。
- 上記のコマンドは ossfs ブートスクリプトを別のサービスとして設定するため、デバイスの起動時に自動的に起動されます。
- これで、ossfs は起動時に自動的にマウントされるようになりました。まとめると、Ubuntu 14.04 または CentOS 6.5 を使用している場合は、ステップ 1 と 2A を実行します。CentOS 7.0 を使用している場合は、ステップ 1 および 2B を実行します。

- 質問: fusermount: failed to open current directory: Permission denied エラーを解決するには、どうすればよいですか。

- これはヒューズのバグです。現在のユーザーには、現在のディレクトリ (マウント解除されたディレクトリ) に対する読み取り権限が必要です。この問題を解決するには、cd コマンドを実行して読み取り権限を持つディレクトリに変更した後、再度 ossfs コマンドを実行します。

- 質問: www ユーザーを使用して ossfs をマウントする必要があります。この場合、どのように自動マウントを設定すればよいですか。

- 前述の質問に対する回答をご参照ください。前述のステップ 1 を実行します。/etc/init.d/ossfs ファイル内のコマンドを次のように変更して、ステップ 2B を実行します。

```
sudo -u www ossfs your_bucket your_mountpoint -ourl=your_url
```

- sudo を使用して、/etc/sudoers を編集できるようにブートスクリプトを設定します。 `Defaults requiretty` 行を、 `#Defaults requiretty` に変更します (この行をコメントアウトします)。

- 質問: fusermount: failed to open current directory: Permission denied エラーを解決するには、どうすればよいですか。

- これは**ヒューズのバグ**です。現在のユーザーには、現在のディレクトリ (マウント解除されたディレクトリ) に対する読み取り権限が必要です。この問題を解決するには、cd コマンドを実行して読み取り権限を持つディレクトリに変更した後、もう一度 ossfs コマンドを実行します。
- 質問: ECS を使って ossfs をマウントすることでファイルをスキャンするコストをどのように回避できますか。
 - プログラムは ossfs によってマウントされたディレクトリをスキャンし、OSS へのリクエストに変換します。リクエストの数が多い場合、コストが発生します (1 セント / 1 万倍)。updatedb の場合は、/etc/updatedb.conf を変更することによりスキップできます。具体的な方法は次のとおりです。
 - a. PRUNEFS = に fuse.ossfs を追加します。
 - b. マウントされたディレクトリを PRUNEPATHS = に追加します。
 - どのプロセスがカタログを一掃したのかは、どのように判断できますか。
 - a. 最初に auditd をインストールします。: sudo apt-get install auditd
 - b. auditd を起動します。: sudo service auditd を起動します。
 - c. モニターマウントディレクトリを設定します。: auditctl -w /mnt/ossfs
 - d. auditorium ログで、どのディレクトリがこのディレクトリにアクセスしたかを確認することができます。: ausearch -i | grep /mnt/ossfs
- 質問: ossfs を使用して、OSS all "application/ocdet-stream にアップロードする content-type ファイルは何ですか。
 - ossfs は、/etc/mime.types の内容を照会して、ファイルの Content-Type を決定します。ファイルが存在することを確認してください。存在しない場合は、追加する必要があります。
 - a. Ubuntu の場合は、udo apt-get install mime-support を使って追加します。
 - b. Centos の場合は、sudo Yum install mailcap を使って追加します。
 - c. フォームごとに 1 行ずつ手動で追加することもできます。それぞれの形式は次のとおりです。: Application/JavaScript JS
- 質問: どのようにして、スーパーバイザーを使用して ossfs を起動することができますか。
 1. スーパーバイザーをインストールするには、Ubuntu で sudo apt-Get install supervisor を実行します。
 2. ディレクトリを作成し、ossfs STARTUP スクリプトを編集します。

```
mkdir /root/ossfs_scripts
vi /root/ossfs_scripts/start_ossfs.sh
```

以下のデータを記述します。:

```
# Unload
fusermount -u /mnt/ossfs

# Re-mounted, you must add-F parameter to run ossfs, let ossfs run at the front desk
exec ossfs my-bucket my-mount-point -ourl=my-oss-endpoint -f
```

3. 次の段落を最後に追加するため、/etc/Supervisor/supervisord.conf を編集します。


```
[program:ossfs]
command=bash /root/ossfs_scripts/start_ossfs.sh
logfile=/var/log/ossfs.log
log_stdout=true
log_stderr=true
logfile_maxbytes=1MB
logfile_backups=10
```

4. スーパーバイザーを実行します。:

```
supervisord
```

supervisord

5. すべて問題ないことを確認します。

```
ps aux | grep supervisor # should be able to see the supervisor Process
ps aux | grep ossfs # should be able to see ossfs Process
kill -9 ossfs # Kill ossfs process, the supervisor must restart it, do not use killall, because killall sends sigterm, the process Exits normally, and the Supervisor no longer reruns ossfs.
ps aux | grep ossfs # should be able to see ossfs Process
```

エラーが発生した場合は、/var/log/supervisor/supervisord.log および /var/log/ossfs.log を確認します。

- 質問: fuse の警告が表示されます(エラー内容: "Library too old, some operations may not work")。

これは、ossfs がコンパイル時に使用する libfuse のバージョンが原因で発生します。バージョンが実行時にリンクされる libfuse のバージョンよりも高い場合にこのエラーが発生します。これは、libfuse をユーザーご自身でインストールしたことが原因である場合が多いです。libfuse を再度インストールせずに、提供する RPM パッケージを使用して ossfs をインストールします。

ボックスで提供する RPM バグとボックスは、そのボックスを含んでいます。実行環境にチェーンがあり、ossfs が旧バージョンの fuse にリンクされている場合、上記の警告が表示されます。

1. どのようにしたら ossfs ランタイムリンクの fuse バージョンを確認することができますか。

- LDD \$ (which ossfs) | grep Fuse を実行します。
- たとえば、結果は "/lib64/libfuse. So. 2" です。その後、LS-L/lib64/libfuse を介して fuse のバージョンを確認できます。

2. ossfs を正しいバージョンにリンクするにはどうすればよいですか。

- まず、rpm-QL ossfs | grep fuse を使用して libfuse のディレクトリを見つけます。
- たとえば、結果は "/usr/lib/libfuse. So. 2" になります。fig =/usr/lib ossfs... を実行し、ossfs を実行します。

3. この警告を無視してもいいですか。

- このバグは確認しない方がよいでしょう。

- 質問: ossfs でファイル情報 (サイズなど) が他のツールで見られる情報と一致しないのはなぜですか。

ossfs は既定では、ファイルのメタ情報 (サイズ / パーミッションなどを含む) をキャッシュするので、ファイル情報は、スピードアップのために OSS に送られる ls リクエストを毎回必要とはしません。ユーザーが他のプログラム (SDK / Webサイトコンソール / osscmd など) を渡した場合、ファイルは変更されているため ossfs でファイル情報を確認できますが、適時には更新されません。

ossfs のキャッシュを無効にしたい場合は、次のパラメーターを追加します。 "-

```
omax_stat_cache_size=0"
```

8. osscmd

8.1. クイックインストール

概要

osscmd は Python 2.x をベースにしたコマンドラインツールで、バケット管理およびファイル管理をサポートしています。

❓ 説明 osscmd の代わりに **ossutil** を使用することを推奨します。

- アプリケーションシナリオ
 - API の開発とデバッグ。たとえば、特定の形式で要求を送信し、段階的にマルチパートアップロードを実行します。
 - logging / website / lifecycle など、コンソールが利用できない場合の設定
- 制限事項
 - osscmd は Python 2.5 / 2.6 / 2.7 をサポートしていますが、Python 3.x はサポートしていません。
 - osscmd は Python SDK V0.x に基づいて開発されていますが、現在、保守は行われていません。現在は、**Python SDK V2.x.x** がサポートされています。
 - osscmd は、低頻度アクセスのストレージ、アーカイブストレージ、リージョン間レプリケーション、イメージのオリジン検索などの新しい機能をサポートしていません。デバッグのみをサポートしています。

osscmd の代わりに ossutil を使用することを推奨します。ossutil には以下の利点があります。

- Windows / Linux / Mac をサポートしています。
- 簡単なインストールと優れたパフォーマンスを特徴とする Go SDK に基づいて実装されています。
- 簡単なコマンドと豊富なヘルプ情報を提供し、中国語と英語をサポートしています。

環境要件

osscmd は python SDK 0.X でリリースされています。[ダウンロードはこちらをクリックしてください](#)。
python SDK 2.x はまだ対応するバージョンの osscmd を提供していませんのでご注意ください。

Python SDK には Python 対応の環境が必要です。Python バージョン: バージョン2.5 からバージョン2.7。SDK は Windows と Linux に適用されますが、Python3.0 は SDK バージョン2.x と完全に互換性がないため、SDK は Python3.0 以降をサポートしていません。

Python のインストール後:

- Linux シェルに "python" と入力して Enter キーを押すと、Python のバージョンが次のように表示されます。

```
Python 2.5.4 (r254:67916, Mar 10 2010, 22:43:17)
[GCC 4.1.2 20080704 (Red Hat 4.1.2-46)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
```

- 次のように、Windows cmd に "python" と入力して Enter キーを押し、Python のバージョンを表示します。

```
C:\Documents and Settings\Administrator>python
Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v. 1500 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license" for more information.
```

上記のコードは Python が正常にインストールされたことを示しています。

- 例外: Windows の cmd に "python" と入力して Enter キーを押すと、システムにより "内部コマンドまたは外部コマンドではありません" というプロンプトが表示されます。そのような場合は、環境変数 > パス の設定を確認し、Python のインストールパスを追加します。

Python がインストールされていない場合は、『[Python 公式 Web サイト](#)』からインストーラ を入手します。この Web サイトには、Python のインストールと使用に関する詳細な手順とガイダンスがあります。

インストールと使用方法

ダウンロードした Python SDK を osscommand のディレクトリに解凍し、`python osscommand + operation` を実行します。一例として、オブジェクトをバケットにアップロードします。

```
python osscommand put myfile.txt oss://mybucket
```

osscommand では、`oss://bucket` または `oss://bucket/object` がバケットまたはオブジェクトを示すために使用されていることに注意してください。`oss://` は単にリソースを示すための方法であり、他の意味を持っていません。

詳細なコマンドリストが必要な場合は、"`python osscommand`" と入力します。

詳細なパラメーターリストの説明が必要な場合は、"`python osscommand help`" と入力します。

8.2. 例

osscommand のインストールと設定

Linux または Windows で SDK インストーラをダウンロードした後、ダウンロードしたパッケージを解凍して osscommand の使用を開始します。

使用方法を取得するため、`python osscommand` を直接実行します。すべてのコマンドには 2 つの実行モードがあります。たとえば、ユーザーが作成したバケットをクエリするとします。`gs` コマンド ("`get service`" の略) が実行されます。

- 方法 1: ID およびキーを指定しない場合、osscommand は既定のファイルから ID およびキーを読み取ります。

```
$ python osscommand gs
can't get accessid/accesskey, setup use : config --id=accessid --key=accesskey
```

❓ 説明 このようなプロンプトが表示された場合は、ID およびキーが正しく設定されていないことを示しています。ステップ 2 の設定コマンドをご参照ください。

ID およびキーが正しく設定され有効になったら、コマンドを実行します。

```
$ python osscommand gs
2013-07-19 08:11 test-oss-sample
Bucket Number is: 1
```

- 方法2: コマンドに ID およびキーを指定すると、osscommand はコマンドラインから ID およびキーを読み取ります。ID およびキーが有効の場合、コマンドを実行すると以下の結果が表示されます。

```
$ python osscommand gs --id=your_id --key=your_key --host=your_endpoint
2013-07-19 08:11 test-oss-sample
Bucket Number is: 1
```

ユーザーの ID およびキーを既定のファイルに設定するには、次のコマンドを実行します。既定の oss ホストは oss.aliyuncs.com です。

```
$python osscommand config --id=your_id --key=your_key --host=your_endpoint
```

"設定は ~ に保存されます" などのプロンプトが表示された場合、ID およびキーが正常に保存されたことを示しています。

基本操作

- 作成したバケットの一覧表示

```
$python osscommand getallbucket
```

OSS ユーザーがバケットを作成しなかった場合、出力は空になります。

- バケットの作成

"mybucketname" という名前のバケットを作成します。

```
$python osscommand createbucket mybucketname
```

OSS のバケットの名前はグローバルに一意です。他の誰かがこの名前のバケットを作成した可能性があるため、"mybucketname" という名前のバケットの作成に失敗する可能性があります。バケットの作成に失敗した場合、プロジェクト名を変更する必要があります。たとえば、バケット名に特定の日付を追加します。

- バケットが正常に作成されたかどうかの確認

```
$ python osscommand getallbucket
```

失敗した場合は、返されるエラーメッセージを確認します。

- オブジェクトの表示

バケットの作成が成功したら、バケット内のオブジェクトを確認します。

```
$python osscommand list oss://mybucketname/
```

オブジェクトはバケットに含まれていないため、出力は空です。

- オブジェクトのアップロード

バケットにオブジェクトを 1 つアップロードします。ローカルファイルの名前が "local_existed_file" の場合、MD5 値は次のようになります。

```
$ md5sum local_existed_file 7625e1adc3a4b129763d580ca0a78e44 local_existed_file
$ python osscommand put local_existed_file oss://mybucketname/test_object
```

 説明 md5sum コマンドは、Windows ではなく Linux で使用されます。

- オブジェクトの再表示

バケットが正常に作成されたら、バケット内でオブジェクトをもう一度確認します。

```
$python osscommand list oss://mybucketname/
```

- オブジェクトのダウンロード

オブジェクトをバケットからローカルにダウンロードし、ダウンロードしたファイルの md5 値を比較します。

```
$ python osscommand get oss://mybucketname/test_object download_file
$ md5sum download_file
7625e1adc3a4b129763d580ca0a78e44 download_file
```

 説明 md5sum コマンドは、Windows ではなく Linux で使用されます。

- オブジェクトの削除

```
$ python osscommand delete oss://mybucketname/test_object
```

- バケットの削除

 説明 バケットにオブジェクトが含まれている場合、バケットは削除できません。

```
$ python osscommand deletebucket mybucketname
```

ライフサイクルの使用

- ライフサイクル用 XML テキストファイルの設定

```
<LifecycleConfiguration>
<Rule>
<ID>1125</ID>
<Prefix>log_backup/</Prefix>
<Status>Active</Status>
<Expiration>
<Days>2</Days>
</Expiration>
</Rule>
</LifecycleConfiguration>
```

これは、現在時刻から 2 日以上経過したバケット内の "log_backup/" というプレフィックスを持つオブジェクトを削除することを示します。詳細な規則の設定については、「[API リファレンス](#)」をご参照ください。

- ライフサイクルの記述

```
python osscommand putlifecycle oss://mybucket lifecycle.xml
0.150(s) elapsed
```

- ライフサイクルの読み取り

```
python osscommand getlifecycle oss://mybucket
<? xml version="1.0" encoding="UTF-8"? >
<LifecycleConfiguration>
<Rule>
<ID>1125</ID>
<Prefix>log_backup/</Prefix>
<Status>Active</Status>
<Expiration>
<Days>2</Days>
</Expiration>
</Rule>
</LifecycleConfiguration>
0.027(s) elapsed
```

- ライフサイクルの削除

```
python osscommand deletelifecycle oss://mybucket
0.139(s) elapsed
```

- ライフサイクルの読み取り

```
python osscmd getlifecycle oss://mybucket
Error Headers:
[('content-length', '288'), ('server', 'AliyunOSS'), ('connection', 'close'), ('x-oss-request-id', '54C74FEE5D7F6B24E5042630'), ('date', 'Tue, 27 Jan 2015 08:44:30 GMT'), ('content-type', 'application/xml')]
Error Body:
<? xml version="1.0" encoding="UTF-8"? >
<Error>
<BucketName>mybucket</BucketName>
<Code>NoSuchLifecycle</Code>
<Message>No Row found in Lifecycle Table.</Message>
<RequestId>54C74FEE5D7F6B24E5042630</RequestId>
<HostId>mybucket.oss-maque-hz-a.alibaba.net</HostId>
</Error>
Error Status:
404
getlifecycle Failed!
```

アンチリーチ 設定

- 空のリファラーのアクセスを許可

```
$osscmd putreferer oss://test --allow_empty_referer=true
0.004(s) elapsed
```

- 設定済みのリファラーの取得

```
$osscmd getreferer oss://test
<? xml version="1.0" encoding="UTF-8"? >
<RefererConfiguration>
<AllowEmptyReferer>true</AllowEmptyReferer>
<RefererList />
</RefererConfiguration>
```

- 空のリファラーは許可せず、テストリファラー要求のみを許可

```
$osscmd putreferer oss://test --allow_empty_referer=false --referer='www.test.com'
0.092(s) elapsed
```

- 設定済みのリファラーの取得

```
$osscommand getreferer oss://test
<? xml version="1.0" encoding="UTF-8"? >
<RefererConfiguration>
<AllowEmptyReferer>>false</AllowEmptyReferer>
<RefererList>
<Referer>www.test.com</Referer>
</RefererList>
</RefererConfiguration>
```

- 空のリファラーを許可せず、test および test1 リファラー要求のみを許可

```
$osscommand putreferer oss://test --allow_empty_referer=false --referer='www.test.com,www.test1.com'
```

- 設定済みのリファラーの取得

```
$osscommand getreferer oss://test
<? xml version="1.0" encoding="UTF-8"? >
<RefererConfiguration>
<AllowEmptyReferer>>false</AllowEmptyReferer>
<RefererList>
<Referer>www.test.com</Referer>
<Referer>www.test1.com</Referer>
</RefererList>
</RefererConfiguration>
```

ログの使用

- ログの設定

```
$osscommand putlogging oss://mybucket oss://myloggingbucket/mb
```

- ログの取得

```
$osscommand getlogging oss://mybucket
```

8.3. バケットコマンド

config

コマンドの説明:

```
config --id=[accessid] --key=[accesskey] --host=[host] --sts_token=[sts_token]
```

例:

- `python osscmd config --id=your_id --key=your_key`
- `python osscmd config --id=your_id --key=your_key --host=oss-internal.aliyuncs.com`

getallbucket(gs)

コマンドの説明:

```
getallbucket(gs)
```

ユーザーが作成したバケットを表示します。gs は get サービスの短縮形です。gs は getallbucket と同じ効果が得られます。

例:

- `python osscmd getallbucket`
- `python osscmd gs`

createbucket(cb,mb,pb)

コマンドの説明:

```
createbucket(cb,mb,pb) oss://bucket --acl=[acl]
```

バケットを作成するコマンドです。cb は create bucket の略で、mb は make bucket の略、pb は put bucket の略で、oss://bucket はバケットを示します。acl パラメーターを含めることができますが、必須ではありません。同じ効果が得られるコマンドがいくつかあります。

例:

- `python osscmd createbucket oss://mybucket`
- `python osscmd cb oss://myfirstbucket --acl=public-read`
- `python osscmd mb oss://mysecondbucket --acl=private`
- `python osscmd pb oss://mythirdbucket`

deletebucket(db)

コマンドの説明:

```
deletebucket(db) oss://bucket
```

バケットを削除するコマンドです。db は delete bucket の略です。Deletebucket は db と同じ効果が得られます。

例:

- `python osscmd deletebucket oss://mybucket`
- `python osscmd db oss://myfirstbucket`

deletewholebucket

❓ 説明 このコマンドはすべてのデータを消去し、消去されたデータは復元できないため、非常に危険です。このコマンドは慎重にご使用ください。

コマンドの説明:

```
deletewholebucket oss://bucket
```

バケット、そのオブジェクト、およびマルチパートコンテンツを削除します。

例:

- `python osscommand deletewholebucket oss://mybucket`

getacl

コマンドの説明:

```
getacl oss://bucket
```

バケットへのアクセス権と制御権を取得します。

例:

- `python osscommand getacl oss://mybucket`

setacl

コマンドの説明:

```
setacl oss://bucket --acl=[acl]
```

バケットへのアクセス権限、および制御権限を変更します。aclには、private、public-read、またはpublic-read-writeの3つの制御権限の中から、いずれか1つのみを設定します。

例:

- `python osscommand setacl oss://mybucket --acl=private`

putlifecycle

コマンドの説明:

```
putlifecycle oss://mybucket lifecycle.xml
```

ライフサイクルルールを設定します。lifecycle.xmlはライフサイクルの設定ファイルです。詳細なルール設定については、「[API リファレンス](#)」をご参照ください。

例:

- `python osscommand putlifecycle oss://mybucket lifecycle.xml`

lifecycle.xmlにはライフサイクルの設定ルールが記述されています。例:

```
<LifecycleConfiguration>
<Rule>
<ID>1125</ID>
<Prefix>log_backup/</Prefix>
<Status>Enabled</Status>
<Expiration>
<Days>2</Days>
</Expiration>
</Rule>
</LifecycleConfiguration>
```

getlifecycle

コマンドの説明:

```
osscommand getlifecycle oss://bucket
```

バケットのライフサイクルルールを取得します。

例:

- `python osscommand getlifecycle oss://mybucket`

deletelifecycle

コマンドの説明:

```
osscommand deletelifecycle oss://bucket
```

バケット内のすべてのライフサイクルルールを削除します。

例:

- `python osscommand deletelifecycle oss://mybucket`

putreferer

コマンドの説明:

```
osscommand putreferer oss://bucket --allow_empty_referer=[true|false] --referer=[referer]
```

アンチリーチ ルールを設定します。 `allow_empty_referer` パラメーターは必須であり、null を許可するかどうかを設定するために使用されます。 `referer` パラメーターは、アクセスが許可されているホワイトリストを設定するために使用されます。たとえば、" www.test1.com,www.test2.com" のように指定します ("," は区切り文字)。詳細なルール設定については、「[プロダクト説明書](#)」をご参照ください。

例:

- `python osscommand putreferer oss://mybucket --allow_empty_referer=true --referer="www.test1.com,ww
w.test2.com"`

getreferer

コマンドの説明:

```
osscommand getreferred oss://bucket
```

バケットの アンチリーチ ルールを取得します。

例:

- ```
python osscommand getreferred oss://mybucket
```

## putlogging

コマンドの説明:

```
osscommand putlogging oss://source_bucket oss://target_bucket/[prefix]
```

source\_bucket はログを行うバケットを示し、target\_bucket はログの保存先を示します。 カテゴリ別のクエリを簡単にするため、ソースバケットで生成されるログファイルにプレフィックスを設定できます。

例:

- ```
python osscommand getlogging oss://mybucket
```

getlogging

コマンドの説明:

```
osscommand getlogging oss://bucket
```

バケットのログを取得し、XML ファイルを返します。

例:

- ```
python osscommand getlogging oss://mybucket
```

# 8.4. オブジェクトコマンド

## ls(list)

コマンドの説明:

```
ls(list) oss://bucket/[prefix] [marker] [delimiter] [maxkeys]
```

バケット内のオブジェクトを一覧表示します。

例:

- ```
python osscommand ls oss://mybucket/folder1/folder2
```
- ```
python osscommand ls oss://mybucket/folder1/folder2 marker1
```
- ```
python osscommand ls oss://mybucket/folder1/folder2 marker1 /
```
- ```
python osscommand ls oss://mybucket/
```
- ```
python osscommand list oss://mybucket/ "" "" 100
```

コマンドの説明:

```
ls(list) oss://bucket/[prefix] --marker=xxx --delimiter=xxx --maxkeys=xxx
```

バケット内のオブジェクトを一覧表示します。name_type は、転送に使用されるエンコード方式を指定します。URL エンコードが指定されている場合、制御文字を用いたオブジェクトの表示がサポートされています。

例:

- `python osscmd ls oss://mybucket/folder1/folder2 --delimiter=/`
- `python osscmd ls oss://mybucket/folder1/folder2 --marker=a`
- `python osscmd ls oss://mybucket/folder1/folder2 --maxkeys=10`

mkdir

コマンドの説明:

```
mkdir oss://bucket/dirname
```

"/ " で終わる、サイズが 0 のオブジェクトを作成します。

例:

- `python osscmd mkdir oss://mybucket/folder`

listallobject

コマンドの説明:

```
listallobject oss://bucket/[prefix]
```

バケット内のすべてのオブジェクトを表示し、プレフィックスを指定します。

例:

- `python osscmd listallobject oss://mybucket`
- `python osscmd listallobject oss://mybucket/testfolder/`

deleteallobject

コマンドの説明:

```
deleteallobject oss://bucket/[prefix]
```

バケット内のすべてのオブジェクトを削除し、プレフィックスを指定します。

例:

- `python osscmd deleteallobject oss://mybucket`
- `python osscmd deleteallobject oss://mybucket/testfolder/`

downloadallobject

コマンドの説明:

```
downloadallobject oss://bucket/[prefix] localdir --replace=false --thread_num=5
```

ディレクトリ構造を変更せずに、バケット内のオブジェクトをローカルディレクトリにダウンロードします。プレフィックスを指定してダウンロードすることができます。—replace = false は、同じ名前のローカルファイルがすでに存在する場合、ダウンロード中に置き換えられないことを示します。—replace = true とすると、同じ名前のローカルファイルは置き換え可能であることを示します。thread_num を使用してダウンロードスレッドを設定することもできます。

例:

- `python osscmd downloadallobject oss://mybucket /tmp/folder`
- `python osscmd downloadallobject oss://mybucket /tmp/folder --replace=false`
- `python osscmd downloadallobject oss://mybucket /tmp/folder --replace=true --thread_num=5`

downloadtodir

コマンドの説明:

```
downloadtodir oss://bucket/[prefix] localdir --replace=false
```

ディレクトリ構造を変更せずに、バケット内のオブジェクトをローカルディレクトリにダウンロードします。プレフィックスを指定してダウンロードすることができます。—replace = false は、同じ名前のローカルファイルがすでに存在する場合、ダウンロード中に置き換えられないことを示します。—replace = true とすると、同じ名前のローカルファイルは置き換え可能であることを示します。このコマンドの実行は、downloadallobject と同じ効果が得られます。

例:

- `python osscmd downloadtodir oss://mybucket /tmp/folder`
- `python osscmd downloadtodir oss://mybucket /tmp/folder --replace=false`
- `python osscmd downloadtodir oss://mybucket /tmp/folder --replace=true`

uploadfromdir

コマンドの説明:

```
uploadfromdir localdir oss://bucket/[prefix] --check_point=check_point_file --replace=false --check_md5=false --thread_num=5
```

ローカルファイルをバケットにアップロードします。たとえば、localdirは /tmp/ です。

次の3つのファイル a/b 、 a/c 、および a がある場合、これらを OSS にアップロードすると、それぞれ oss://bucket/a/b 、 oss://bucket/a/c 、および oss://bucket/a となります。prefix を mytest として指定した場合、OSS にアップロードしたファイルは oss://bucket/mytest/a/b 、 oss://bucket/mytest/a/c 、および oss://bucket/mytest/a となります。

`--check_point=check_point_file` は指定されたファイルです。ファイルが指定された後、osscommand はアップロードされたローカルファイルをタイムスタンプとして `check_point_file` に格納されます。`uploadfromdir` コマンドはアップロードされるファイルのタイムスタンプと、`check_point_file` に記録されたファイルのタイムスタンプとを比較します。各ファイルに変更があると、各ファイルは再アップロードされます。それ以外の場合、ファイルはスキップされます。`check_point_file` は既定では存在しません。`--replace=false` は、同じ名前のローカルファイルがすでに存在する場合、ダウンロード中に置き換えられないことを示します。`--replace=true` は、同じ名前のローカルファイルが置き換え可能であることを示します。`--check_md5=false` は、ファイルがアップロードされているときに、Content-MD5 要求ヘッダーの検証が行われないことを示します。`true` は、Content-MD5 リクエストヘッダーの検証が行われることを示します。

❓ 説明 `check_point_file` 内のログには、アップロードされたすべてのファイルのログが記録されます。アップロードするファイルが多すぎると、`check_point_file` のサイズは相当に大きくなります。

例:

- `python osscommand uploadfromdir /mytemp/folder oss://mybucket`
- `python osscommand uploadfromdir /mytemp/folder oss://mybucket --check_point_file=/tmp/mytemp_record.txt`
- `python osscommand uploadfromdir C:\Documents and Settings\User\My Documents\Downloads oss://mybucket --check_point_file=C:\cp.txt`

put

コマンドの説明:

```
put localfile oss://bucket/object --content-type=[content_type] --headers="key1:value1#key2:value2" --check_md5=false
```

ローカルファイルをバケットにアップロードするときは、オブジェクトの `content-type` を指定するか、またはカスタマイズしたヘッダーを指定します。`--check_md5=false` は、ファイルがアップロードされているときに、Content-MD5 要求ヘッダーの検証が行われないことを示します。`true` は、Content-MD5 リクエストヘッダーの検証が行われることを示します。

例:

- `python osscommand put myfile.txt oss://mybucket`
- `python osscommand put myfile.txt oss://mybucket/myobject.txt`
- `python osscommand put myfile.txt oss://mybucket/test.txt --content-type=plain/text --headers="x-oss-meta-des:test#x-oss-meta-location:CN"`
- `python osscommand put myfile.txt oss://mybucket/test.txt --content-type=plain/text`

Upload

コマンドの説明:

```
upload localfile oss://bucket/object --content-type=[content_type] --check_md5=false
```

ローカルファイルをオブジェクトグループにアップロードします。推奨されていません。 --

`check_md5=false` は、ファイルがアップロードされているときに、Content-MD5 要求ヘッダーの検証が行われないことを示します。true は、Content-MD5 リクエストヘッダーの検証が行われることを示します。

例:

- `python osscommand upload myfile.txt oss://mybucket/test.txt --content-type=plain/text`

get

コマンド説明:

```
get oss://bucket/object localfile
```

オブジェクトをローカルにダウンロードします。

例:

- `python osscommand get oss://mybucket/myobject /tmp/localfile`

multiget(multi_get)

コマンドの説明:

```
multiget(multi_get) oss://bucket/object localfile --thread_num=5
```

マルチスレッドでオブジェクトをローカルにダウンロードします。スレッド数を設定することができます。

例:

- `python osscommand multiget oss://mybucket/myobject /tmp/localfile`
- `python osscommand multi_get oss://mybucket/myobject /tmp/localfile`

cat

コマンドの説明

```
cat oss://bucket/object
```

オブジェクトのメタ情報を読み取り、それを出力します。オブジェクトのサイズが大きい場合は使用しないでください。

例:

- `python osscommand cat oss://mybucket/myobject`

meta

コマンドの説明:

```
meta oss://bucket/object
```

オブジェクトのメタ情報を読み取り、それを出力します。メタ情報には、コンテンツタイプ、ファイル長、カスタムメタなどが含まれます。

例:

- `python osscommand meta oss://mybucket/myobject`

copy

コマンドの説明:

```
copy oss://source_bucket/source_object oss://target_bucket/target_object --  
headers="key1:value1#key2:value2"
```

ソースバケットのソースオブジェクトを宛先バケットの宛先オブジェクトにコピーします。

例:

- `python osscmd copy oss://bucket1/object1 oss://bucket2/object2`

rm(delete,del)

コマンドの説明:

```
rm(delete,del) oss://bucket/object --encoding_type=url
```

オブジェクトを削除します。encoding-type がURL エンコードの場合、削除する文字列も URL エンコードである必要があります。

例:

- `python osscmd rm oss://mybucket/myobject`
- `python osscmd delete oss://mybucket/myobject`
- `python osscmd del oss://mybucket/myobject`
- `python osscmd del oss://mybucket/my%01object --encoding_type=url`

signurl(sign)

コマンドの説明

```
signurl(sign) oss://bucket/object --timeout=[timeout_seconds]
```

署名を含む URL を生成し、タイムアウト値を指定します。このコマンドは、プライベートバケット内の指定されたオブジェクトに他のユーザーがアクセスできる場合、実行することができます。

例:

- `python osscmd sign oss://mybucket/myobject`
- `python osscmd signurl oss://mybucket/myobject`

8.5. マルチパートコマンド

init

コマンドの説明:

```
init oss://bucket/object
```

アップロード ID の生成を開始します。アップロード ID は multiupload コマンドと組み合わせて使用します。

例:

- `python osscmd init oss://mybucket/myobject`

listpart

コマンドの説明:

```
listpart oss://bucket/object --upload_id=xxx
```

指定されたオブジェクトのアップロード ID について、アップロードされたパーツを表示します。関連する概念については、「OSS API リファレンス」をご参照ください。アップロード ID は必ず指定する必要があります。

コマンドの説明:

- `python osscmd listpart oss://mybucket/myobject --upload_id= 75835E389EA648C0B93571B6A46023F3`

listparts

コマンドの説明:

```
listparts oss://bucket
```

バケット内の未完了のマルチパートアップロード ID とオブジェクトを表示します。バケットを削除しようとした際、バケットが空ではないというシステムプロンプトが表示された場合、このコマンドを使用することで、マルチパートコンテンツが含まれているかどうかを確認できます。

例:

- `python osscmd listparts oss://mybucket`

getallpartsize

コマンドの説明:

```
getallpartsize oss://bucket
```

バケット内の既存のアップロード ID のパーツ合計サイズを表示します。

例:

- `python osscmd getallpartsize oss://mybucket`

cancel

コマンドの説明:

```
cancel oss://bucket/object --upload_id=xxx
```

アップロード ID のマルチパートアップロードイベントを終了します。

例:

- `python osscmd cancel oss://mybucket/myobject --upload_id= D9D278DB6F8845E9AFE797DD235DC576`

Multiupload (multi_upload, MP)

コマンドの説明:

```
multiupload(multi_upload,mp) localfile oss://bucket/object --check_md5=false --thread_num=10
```

ローカルファイルをマルチパートで OSS にアップロードします。

例:

- `python osscmd multiupload /tmp/localfile.txt oss://mybucket/object`
- `python osscmd multiup_load /tmp/localfile.txt oss://mybucket/object`
- `python osscmd mp /tmp/localfile.txt oss://mybucket/object`

コマンドの説明:

```
multiupload(multi_upload,mp) localfile oss://bucket/object --upload_id=xxx --thread_num=10 --max_part_num=1000 --check_md5=false
```

ローカルファイルをマルチパートで OSS にアップロードします。ローカルファイルのパーツ数は max_part_num で定義されています。このコマンドを実行すると、まずアップロード ID の対応するパーツの ETag がローカルファイルの MD5 値と一致しているかどうかを判断します。一致していれば、アップロードはスキップされます。そのため、使用前に生成されたアップロード ID だった場合、アップロード ID はパラメーターに含まれます。アップロードが失敗した場合でも、multiupload コマンドを繰り返すことでアップロードを再開できます。--check_md5=false は、ファイルがアップロードされているときに、Content-MD5 要求ヘッダーの検証が行われないことを示します。true は、Content-MD5 要求ヘッダーの検証が行われることを示します。

例:

- `python osscmd multiupload /tmp/localfile.txt oss://mybucket/object --upload_id= D9D278DB6F8845E9AFE797DD235DC576`
- `python osscmd multiup_load /tmp/localfile.txt oss://mybucket/object --thread_num=5`
- `python osscmd mp /tmp/localfile.txt oss://mybucket/object --max_part_num=100`

copylargefile

コマンドの説明:

```
copylargefile oss://source_bucket/source_object oss://target_bucket/target_object --part_size=10*1024*1024 --upload_id=xxx
```

1GB を超える大きなファイルをコピーする場合、オブジェクトはマルチパートを介して、コピー先にコピーします (コピー元バケットとコピー先バケットは同じリージョン内にある必要があります)。upload_id はオプションのパラメーターです。マルチパートコピーイベントの転送を再開したい場合は、upload_id を含めることができます。part_size は、パーツサイズを定義するために使用します。1 つのパーツの最小サイズは 100 KB で、最大 10,000 個のパーツがサポートされています。part_size の設定値が OSS の制限値と競合する場合、アプリケーションは自動的にパーツサイズを調整します。

例:

- `python osscmd copylargefile oss://source_bucket/source_object oss://target_bucket/target_object --part_size=10*1024*1024`

uploadpartfromfile (upff)

コマンドの説明:

```
uploadpartfromfile (upff) localfile oss://bucket/object --upload_id=xxx --part_number=xxx
```

このコマンドは主に、テスト用として使用されます。実際の使用は推奨されません。

uploadpartfromstring(upfs)

コマンドの説明:

```
uploadpartfromstring(upfs) oss://bucket/object --upload_id=xxx --part_number=xxx --data=xxx
```

このコマンドは主に、テスト用として使用されます。実際の使用は推奨されません。