スマートストレージ

サービスご利用の手引き

（オブジェクトストレージ版）

（第 1.3 版　2026 年　9 月　30 日）

We are the future.

Smart Connect

目次

1.はじめに 3

2.REST API ガイド 4

REST API を使用した接続の設定 4
1.ストレージアクセス用 IP アドレスの確認 4
2.ストレージアクセス時のポート番号の確認 4
REST API の設定及びテスト 6

3.各 S3 クライアントツールのインストール方法・使用方法 7

AWS CLI のインストール・使用方法(Linux) 7
インストール前の確認 7
Python3（および pip3）のインストール 7
pip3 を使用して AWS CLI をインストール 8
アクセスキーの設定 8
AWS CLI の使用 9
AWS SDK for Python のインストール・使用方法(Linux) 12
インストール前の確認 12
Python3（および pip3）のインストール 12
pip3 を使用して AWS SDK for Python(Boto3)のインストール 13
アクセスキーの設定 13
AWS SDK for Python の使用 14
S3 browser のインストール・使用方法(Windows) 19
S3 browser のダウンロード 19
S3 browser のインストール 19
S3 browser の起動とアカウントの追加 22
S3 browser の使用 23

4.アクセスログについて 27

アクセスログのご利用方法 27
アクセログの閲覧方法 27
アクセスログの形式 29

5.改訂履歴 30

1.はじめに

スマートストレージ マネージドオブジェクトプランはネットアップ社が提供する「ONTAP S3」を採用しています。
また本書に記載の、AWS(※1)、Amazon S3(※2)は、米国その他諸国における、Amazon.com,Inc またはその関連会社の商標です。

※1 アマゾン ウェブ サービス（AWS） : Amazon Web Services, Inc.が提供するクラウドコンピューティングサービスです。
※2 Amazon Simple Storage Service (Amazon S3) : Amazon Web Services, Inc.が提供するオブジェクトストレージサービスです。

2.REST API ガイド

REST API を使用した接続の設定

REST API を活用してオブジェクトストレージを利用する前に、接続の確認をする必要があります。

手順

1. ストレージアクセス用 IP アドレスの確認
2. ストレージアクセス時のポート番号の確認
3. 弊社提供のワイルドカード証明書を SSL 証明書認証するための設定（任意）
4. S3 REST API の設定及びテスト

1.ストレージアクセス用 IP アドレスの確認

ストレージアクセスポイントの IP アドレスは以下の通りです。

接続先	IP アドレス
ストレージアクセス用 IP アドレス	開通通知書に記載

2.ストレージアクセス時のポート番号の確認

ストレージにアクセスする際に使用するポート番号を確認しておく必要があります。S3 クライアントでは、ストレージへの接続に次のポートが使用されます。

用途	ポート番号
HTTPS アクセス	443
HTTP アクセス	80

3.弊社提供のワイルドカード証明書を SSL 証明書認証するための設定(任意)

ストレージに SSL 証明書認証して https アクセスする場合、設定シートに記載されたストレージアクセス用 IP アドレスに対して、smstr.jp をご利用いただけます。

例：Hosts ファイルに登録する場合

```
10.3.0.151      smstr.jp
```

注：本設定を行わず IP アドレスでアクセスする場合、各種接続クライアントツールの証明書確認（verify）設定は無効にする必要があります。（詳細は「各 S3 クライアントツールのインストール方法・使用方法」に後述）

仮想ホスト形式でバケット(またはオブジェクト)にアクセスする場合、設定シートに記載されたストレージアクセス用 IP アドレスに対して、[希望のバケット名].smstr.jp をご利用いただけます。

例：Hosts ファイルに登録する場合

```
10.3.0.151      [希望のバケット名].smstr.jp
```

注：Hosts ファイルに設定する場合、バケット毎にドメイン名を登録する必要があります。ワイルドカードに対応している DNS であれば、*.smstr.jp の登録のみで問題ありません。

REST API の設定及びテスト

以下のいずれかのツールを使用してシステムへの接続を確認することができます。リンクから「**各 S3 クライアントツールのインストール方法・使用方法**」をご確認ください。

- **Amazon Web Services Command Line Interface（AWS CLI）**

  注：Linux/Windows 対応。(マニュアルは Linux CentOS7 でのインストール方法・使用方法のみ)

- **Amazon Web Services Software Development Kit（AWS SDK）**

  注：Linux/Windows 対応。(マニュアルは Linux CentOS7 でのインストール方法・使用方法のみ)

- **S3 browser**

  注：Windows 対応。(マニュアルは Windows10 でのインストール方法・使用方法のみ)

- **s3fs**

  注：Linux/Windows 対応。(マニュアルは Linux CentOS7 でのインストール方法・使用方法のみ)

3.各 S3 クライアントツールのインストール方法･使用方法

本章では、各 S3 クライアントツールのインストール方法･使用方法についてご説明します。
尚、いずれのクライアントツールも弊社のサポート対象外となります。ご了承の上、お使いください。

AWS CLI のインストール･使用方法(Linux)

AWS CLI は CUI を使ってオブジェクトストレージにアクセスできるツールです。
このマニュアルでは Linux CentOS7 を使って AWS CLI のインストール方法から使用方法までを紹介します。
尚、AWS CLI は弊社のサポート対象外となります。ご了承の上、お使いください。

インストール前の確認

pip3 がすでに存在する場合は、**「pip3 を使用して AWS CLI をインストール」**に進みます。pip --version を実行して、Linux CentOS7 に既に pip3 がインストールされているかどうか確認します。

注：awscli パッケージは apt や yum など他のパッケージマネージャー用にリポジトリで利用可能ですが、pip から入手するか、バンドルインストーラを使用しない限り、最新バージョンであることは保証されません。

```
$ pip3 --version
```

pip3 がない場合は、インストールされている Python のバージョンを確認します。

```
$ python --version
```

または

```
$ python3 --version
```

まだ Python 3.4 以降を持っていない場合は、**「Python3（および pip3）のインストール」**に進みます。

Python3（および pip3）のインストール

Python 3.4 以降がインストールされていない場合は、yum を使用して Python3.4 以降をインストールします。尚、Python 3.4 以降がインストールされると pip3 も同時にインストールされ、使用できるようになります。

```
$ sudo yum install python3
```

インストール先および使用方法の詳細については、ご使用のシステムのパッケージマネージャーのドキュメント、および Python(https://www.python.org/doc/)のドキュメントを参照してください。

次のコマンドを実行して、Python が正しくインストールされたことを確認します。

```
$ python3 --version
Python 3.6.8
```

併せて、次のコマンドを実行して、pip3 が正しくインストールされたことも確認します。

```
$ pip3 --version
pip 9.0.3 from /usr/lib/python3.6/site-packages (python 3.6)
```

pip3 を使用して AWS CLI をインストール

pip3 を使用して AWS CLI をインストールします。

```
$ sudo pip3 install awscli --upgrade
```

AWS CLI が正しくインストールされたことを確認します。

```
$ aws --version
aws-cli/1.16.280 Python/3.6.8 Linux/3.10.0-1062.4.1.el7.x86_64 botocore/1.13.16
```

エラーが表示される場合は、「AWS CLI エラーのトラブルシューティング」を参照してください。

関連情報

- Amazon Web Services（AWS）ドキュメント：「AWS CLI エラーのトラブルシューティング」
  https://docs.aws.amazon.com/ja_jp/cli/latest/userguide/cli-chap-troubleshooting.html

アクセスキーの設定

AWS CLI 使用してアクセスキーの設定を行います。

```
$ aws configure
AWS Access Key ID [None]: [開通通知書に記載のアクセスキー]
AWS Secret Access Key [None]: [開通通知書に記載のシークレットアクセスキー]
Default region name [None]: [入力不要]
Default output format [None]: [入力不要]
```

AWS CLI の使用

AWS CLI を使用してオブジェクトストレージにアクセスします。下記の URL を参考にしつつ、コマンドを入力してください。また、本章でコマンド入力のいくつかの例を記載しておりますので、ご活用ください。

- **aws s3 コマンドリファレンス（英語）**
  https://docs.aws.amazon.com/cli/latest/reference/s3/index.html#cli-aws-s3

- **aws s3api コマンドリファレンス（英語）**
  https://docs.aws.amazon.com/cli/latest/reference/s3api/index.html#cli-aws-s3api

注：AWS CLI は、Amazon S3 の機能を基に開発されているため、S3 互換である本サービスでは一部使用できない機能がございます。各機能についてはお客様にて検証及びご確認ください。
以下は現状使用が確認できないコマンドになります。(2023/11 時点)

aws s3 コマンド	
website	
aws s3api コマンド	
create-bucket	get-public-access-block
delete-bucket	list-bucket-analytics-configurations
delete-bucket-analytics-configuration	list-bucket-inventory-configurations
delete-bucket-inventory-configuration	list-bucket-metrics-configurations
delete-bucket-metrics-configuration	put-bucket-accelerate-configuration
delete-bucket-website	put-bucket-acl
delete-public-access-block	put-bucket-analytics-configuration
get-bucket-accelerate-configuration	put-bucket-inventory-configuration
get-bucket-analytics-configuration	put-bucket-logging
get-bucket-inventory-configuration	put-bucket-metrics-configuration
get-bucket-logging	put-bucket-request-payment
get-bucket-metrics-configuration	put-bucket-website
get-bucket-policy-status	put-object-acl
get-bucket-request-payment	put-object-legal-hold
get-bucket-website	put-object-lock-configuration
get-object-legal-hold	put-object-retention
get-object-lock-configuration	put-public-access-block
get-object-retention	restore-object
get-object-torrent	select-object-content

注：エンドポイント URL に FQDN を指定する場合、SSL 証明書認証が必要になります。（詳細は「弊社提供のワイルドカード証明書を SSL 証明書認証するための設定（任意）」参照）

注：エンドポイント URL に IP アドレスを指定する場合、コマンド入力の際に「--no-verify-ssl」を追加する必要があります。

例

```
$ aws s3 ls s3://[バケット名] --endpoint-url https://[ストレージアクセス用 IP アドレス] --no-verify-ssl
```

バケット一覧の確認

```
$ aws s3 ls --endpoint-url https://smstr.jp
2019-10-24 11:33:30 test-bucket1
2019-10-25 13:33:48 test-bucket2
```

バケット使用容量の確認

```
$ aws s3 ls --recursive --summarize --human-readable s3://[バケット名] --endpoint-url https://smstr.jp
2019-10-26 10:25:55   74.0 KiB test1.txt
2019-10-26 11:28:56  109.3 KiB test2.txt

Total Objects: 2
   Total Size: 183.3 KiB
```

オブジェクトのアップロード

```
$ aws s3 cp /[アップロードファイルの格納先(例:root)]/[ファイル名.拡張子] s3://[バケット名] --endpoint-url https://smstr.jp
upload: ./test1.txt to s3://test-bucket1/test1.txt
```

オブジェクトのダウンロード

```
$ aws s3 cp s3://[バケット名]/[ファイル名.拡張子] /[ダウンロードファイルの格納先(例:root)]/ --endpoint-url https://smstr.jp
download: s3://test-bucket1/test1.txt to ./test1.txt
```

オブジェクトの削除

```
$ aws s3 rm s3://[バケット名]/[ファイル名.拡張子] --endpoint-url https://smstr.jp
delete: s3://test-bucket1/test1.txt
```

署名付き URL の作成

```
$ aws s3 presign s3://[バケット名]/[ファイル名.拡張子] --expires-in [有効期限(秒数)] --endpoint-url https://smstr.jp
https://smstr.jp/testbucket1/test1.txt?AWSAccessKeyId=GRG9T19N3OOW3MK9GQZH&Signature=2AVgIervLXPgBNdzwr4mt1q2GyY%3D&Expires=1574408095
```

AWS SDK for Python のインストール･使用方法(Linux)

AWS SDK for Python は Python を使ってオブジェクトストレージを操作できるツールです。
ここでは Linux CentOS7 を使って AWS SDK for Python のインストール方法から使用方法までを紹介します。
尚、AWS SDK for Python は弊社のサポート対象外となります。ご了承の上、お使いください。

インストール前の確認

AWS SDK for Python を使用するためには、Python 3.4 以降と pip3 が必要になります。Python 3.4 以降と pip3 がすでにインストールされている場合は、**「pip3 を使って AWS SDK for Python(Boto3)のインストール」**に進みます。

インストールされている Python のバージョンを確認します。

```
$ python --version
```

または

```
$ python3 --version
```

pip3 がインストールされているかどうか確認します。

```
$ pip3 --version
```

まだ Python 3.4 以降をインストールしていない場合は、**「Python3（および pip3）のインストール」**に進みます。

Python3（および pip3）のインストール

Python 3.4 以降がインストールされていない場合は、yum を使用して Python3.4 以降をインストールします。尚、Python3.4 以降がインストールされると pip3 も同時にインストールされ、使用できるようになります。

yum を使用してインストールを行います。

```
$ sudo yum install python3
```

インストール先および使用方法の詳細については、ご使用のシステムのパッケージマネージャーのドキュメント、および Python(https://www.python.org/doc/)のドキュメントを参照してください。

次のコマンドを実行して、Python が正しくインストールされたことを確認します。

```
$ python3 --version
Python 3.6.8
```

併せて、次のコマンドを実行して、pip3 が正しくインストールされたことも確認します。

```
$ pip3 --version
pip 9.0.3 from /usr/lib/python3.6/site-packages (python 3.6)
```

pip3 を使用して AWS SDK for Python(Boto3)のインストール

pip3 を使用して、AWS SDK for Python(Boto3)をインストールします。

```
$ sudo pip3 install boto3
```

アクセスキーの設定

~/.aws のディレクトリを作成します。

```
$ sudo mkdir ~/.aws
```

認証情報ファイルを作成し、下線部分の値を自分用の値に置き換えてから以下の行を保存します。

```
$ vi ~/.aws/credentials
[default]
aws_access_key_id = [開通通知書に記載のアクセスキー]
aws_secret_access_key = [開通通知書に記載のシークレットアクセスキー]
```

AWS SDK for Python の使用

AWS SDK for Python を使用してオブジェクトストレージを操作します。下記の URL を参考にしつつ、操作してください。次ページでは操作のいくつかの例を記載しておりますので、ご活用ください。

- **Boto3 S3 API リファレンス（英語）**
  https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/s3.html

注：AWS SDK for Python は、 Amazon S3 の機能を基に開発されているため、S3 互換である本サービスでは一部使用できない機能がございます。各機能についてはお客様にて検証及びご確認ください。

注：エンドポイント URL に FQDN を指定する場合、SSL 証明書認証が必要になります。（詳細は「弊社提供のワイルドカード証明書を SSL 証明書認証するための設定（任意）」参照）

注：エンドポイント URL に IP アドレスを指定する場合、入力の際に「verify=False」にする必要があります。

例

```
$ vi [希望のファイル名].py
import boto3
import boto3.session

session = boto3.session.Session(profile_name='default')

endpoint = 'https://[ストレージアクセス用 IP アドレス]'
s3 = session.resource(service_name='s3', endpoint_url=endpoint, verify=False)
client = s3.meta.client

for bucket in s3.buckets.all():
    print(bucket.name)
```

バケット一覧の確認

vi コマンドで py ファイルを作成し、以下の内容を記入します。

```
$ vi [希望のファイル名].py
import boto3
import boto3.session

session = boto3.session.Session(profile_name='default')

endpoint = 'https://smstr.jp'
s3 = session.resource(service_name='s3', endpoint_url=endpoint, verify=True)
client = s3.meta.client

for bucket in s3.buckets.all():
    print(bucket.name)
```

python3 コマンドで作成したファイルを実行します。

```
$ python3 [希望のファイル名].py
testbucket1
testbucket2
```

バケット使用容量の確認

vi コマンドで py ファイルを作成後、以下の内容を記入し、保存します。

```
$ vi [希望のファイル名].py
import boto3
import boto3.session

session = boto3.session.Session(profile_name='default')

endpoint = 'https://smstr.jp'
s3 = session.resource(service_name='s3', endpoint_url=endpoint, verify=True)
client = s3.meta.client

bucket = s3.Bucket('バケット名')
bucket_size = 0
for object in bucket.objects.all():
    bucket_size += object.size

print('Bucket Size: {} bytes'.format(bucket_size))
```

python3 コマンドで作成したファイルを実行します。

```
$ python3 [希望のファイル名].py
Bucket Size: 187731 bytes
```

オブジェクトのアップロード

vi コマンドで py ファイルを作成後、以下の内容を記入し、保存します。

```
$ vi [希望のファイル名].py
import boto3
import boto3.session

session = boto3.session.Session(profile_name='default')

endpoint = 'https://smstr.jp'
s3 = session.resource(service_name='s3', endpoint_url=endpoint, verify=True)
client = s3.meta.client

obj = s3.Object('アップロード先のバケット名', 'アップロード先でのファイル名.拡張子')
obj.upload_file('/ファイルのパス(例：root)/ファイル名.拡張子')
```

python3 コマンドで作成したファイルを実行します。

```
$ python3 [希望のファイル名].py
```

オブジェクトのダウンロード

vi コマンドで py ファイルを作成後、以下の内容を記入し、保存します。

```
$ vi [希望のファイル名].py
import boto3
import boto3.session

session = boto3.session.Session(profile_name='default')

endpoint = 'https://smstr.jp'
s3 = session.resource(service_name='s3', endpoint_url=endpoint, verify=True)
client = s3.meta.client

obj = s3.Object('ダウンロード先のバケット名', 'ファイル名.拡張子')
obj.download_file('希望のファイル名.拡張子')
```

python3 コマンドで作成したファイルを実行します。

```
$ python3 [希望のファイル名].py
```

オブジェクトの削除

vi コマンドで py ファイルを作成後、以下の内容を記入し、保存します。

```
$ vi [希望のファイル名].py
import boto3
import boto3.session

session = boto3.session.Session(profile_name='default')

endpoint = 'https://smstr.jp'
s3 = session.resource(service_name='s3', endpoint_url=endpoint, verify=True)
client = s3.meta.client

obj = s3.Object('バケット名', 'ファイル名.拡張子').delete()
```

python3 コマンドで作成したファイルを実行します。

```
$ python3 [希望のファイル名].py
```

署名付き URL の作成

vi コマンドで py ファイルを作成後、以下の内容を記入し、保存します。

```
$ vi [希望のファイル名].py
import boto3
import boto3.session

session = boto3.session.Session(profile_name='default')

endpoint = 'https://smstr.jp'
s3 = session.resource(service_name='s3', endpoint_url=endpoint, verify=True)
client = s3.meta.client

url = client.generate_presigned_url('get_object',
                                    {'Bucket': '[バケット名]',
                                     'Key': '[オブジェクト名.拡張子]'},
                                      ExpiresIn=[有効期限(秒数)])
print("Pre-signed URL: %s" % (url))
```

python3 コマンドで作成したファイルを実行します。

```
$ python3 [希望のファイル名].py
Pre-signed URL: https://smstr.jp/testbucket1/test1.txt?AWSAccessKeyId=GRG9T19N3OO
W3MK9GQZH&Signature=KYvXOQ2M%2FJPkeA7T6vuYSIBcIO4%3D&Expires=1574404750
```

S3 browser のインストール・使用方法(Windows)

S3browser はオブジェクトストレージを GUI で操作できるツールです。
このマニュアルでは Windows10 を使って S3 browser のインストール方法から使用方法までを紹介します。
尚、S3 browser は弊社のサポート対象外となります。ご了承の上、お使いください。

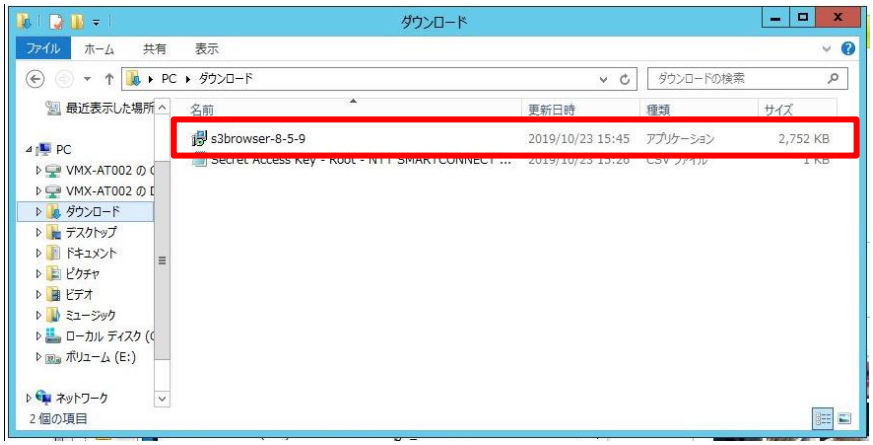
S3 browser のダウンロード

S3 browser を WEB サイトからダウンロードします。
https://s3browser.com/

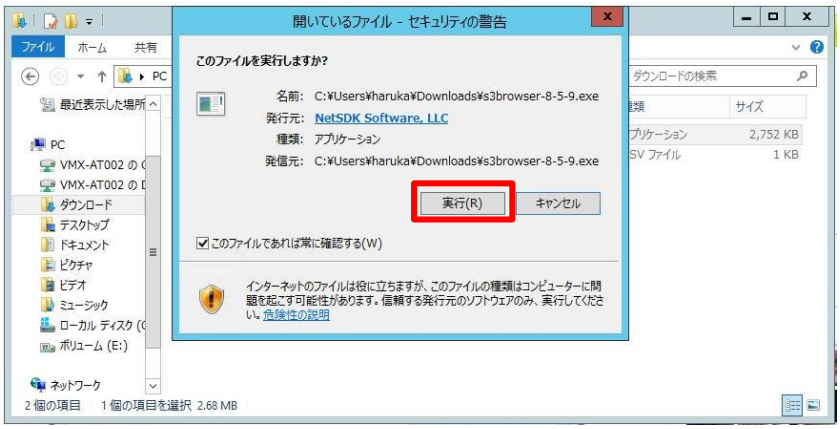
S3 browser のインストール

[S3browser-x-x-x]をダブルクリックします。

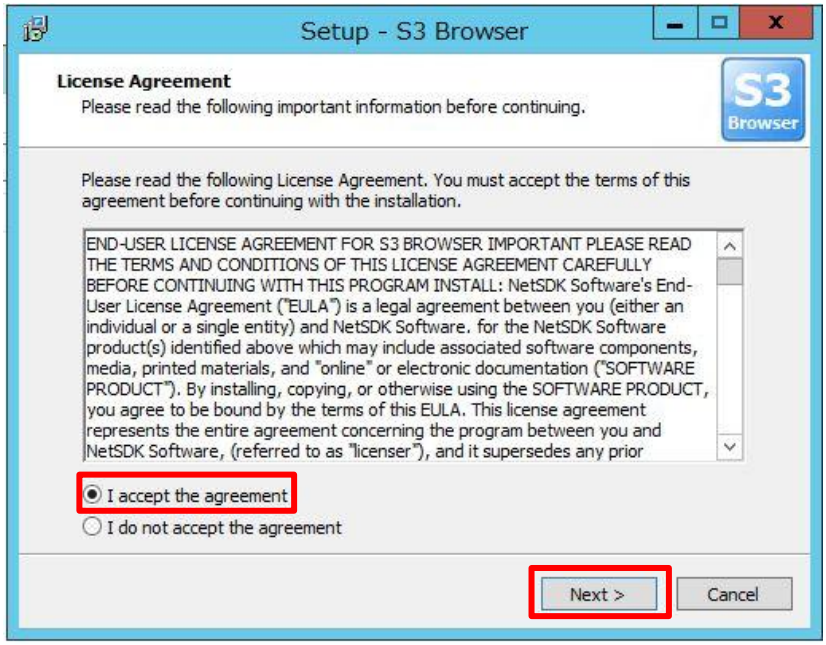
注：バージョンはダウンロードした時期によって異なります。



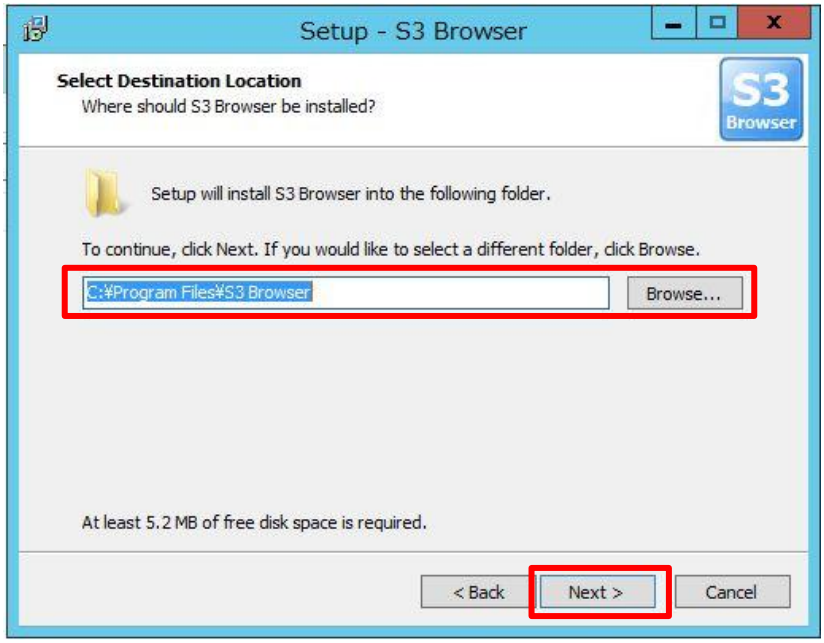
[実行]をクリックします。



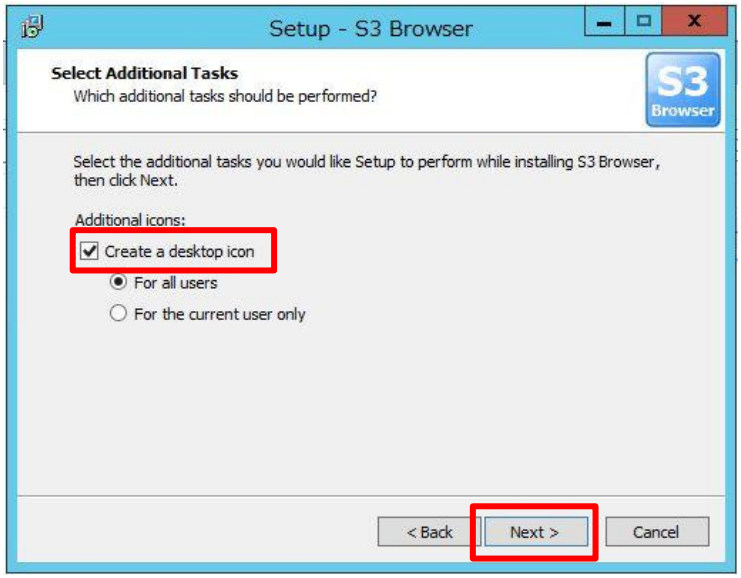
[I accept the agreement]を選択し、**[Next]**をクリックします。



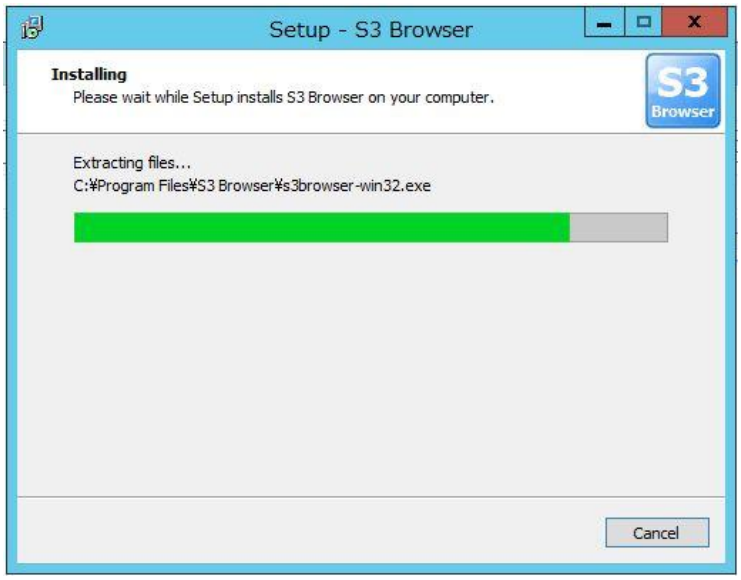
S3browser のインストール先を選択し、**[Next]**をクリックします。



[Create a desktop icon(デスクトップアイコンを作成する)]にチェックを入れ、**[Next]**をクリックします。



(インストール中)

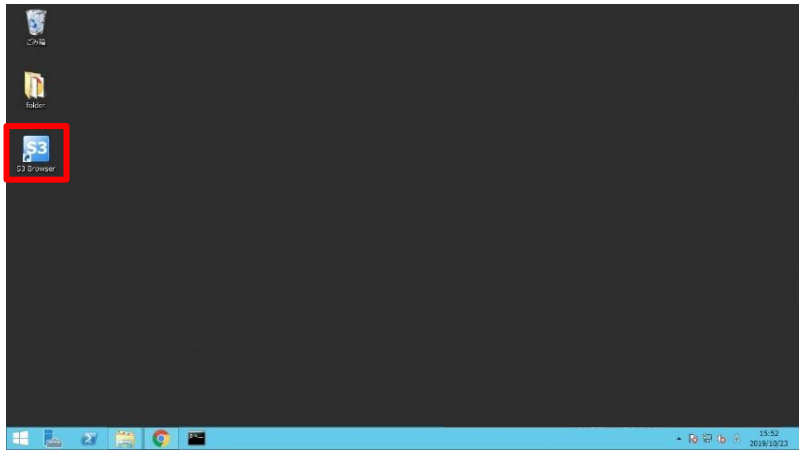


[Finish]をクリックし、インストール完了です。（そのまま S3browser を起動したい場合は、チェックを入れる。）

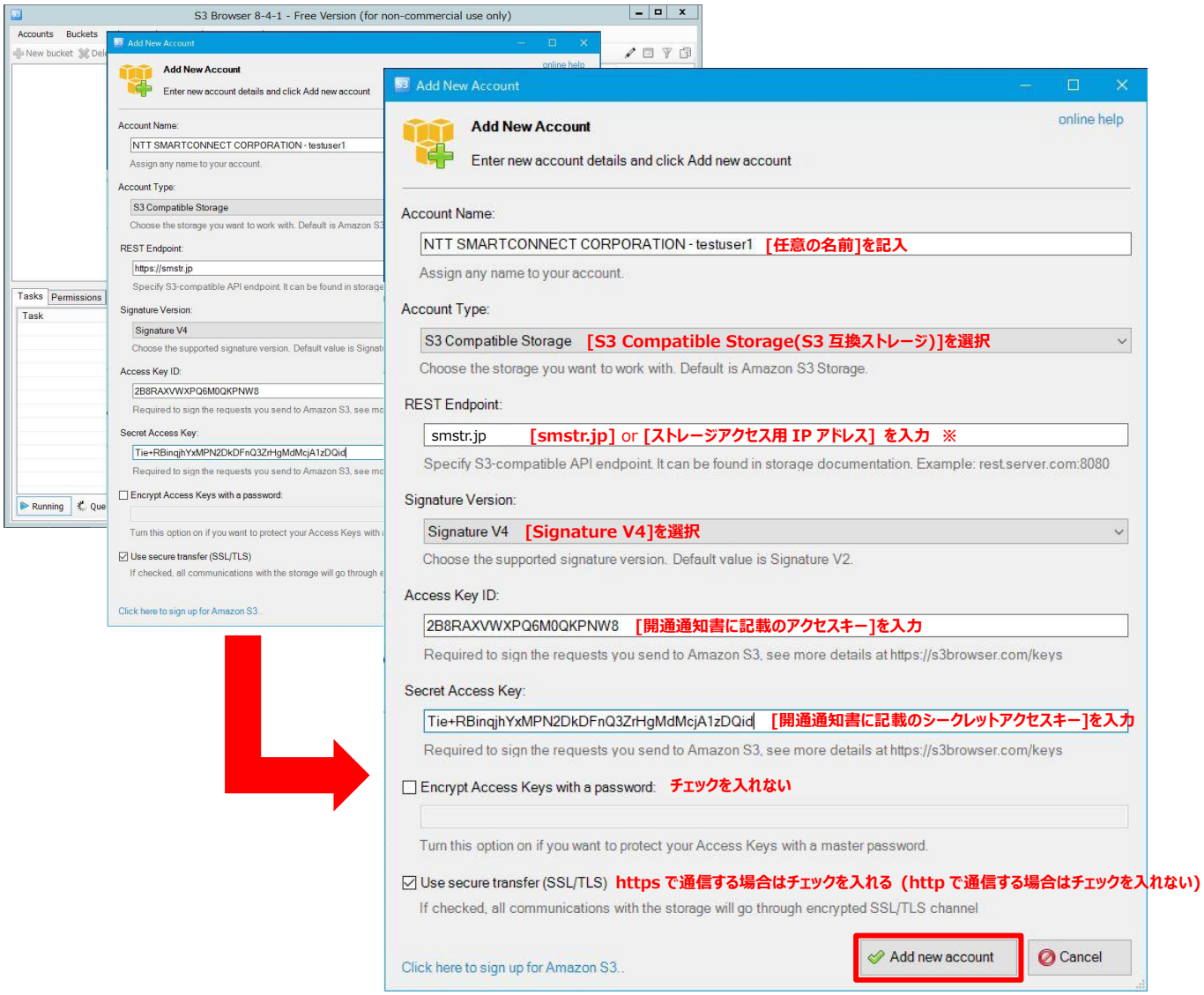


S3 browser の起動とアカウントの追加

デスクトップから S3 browser のアイコンをクリックし、起動します。



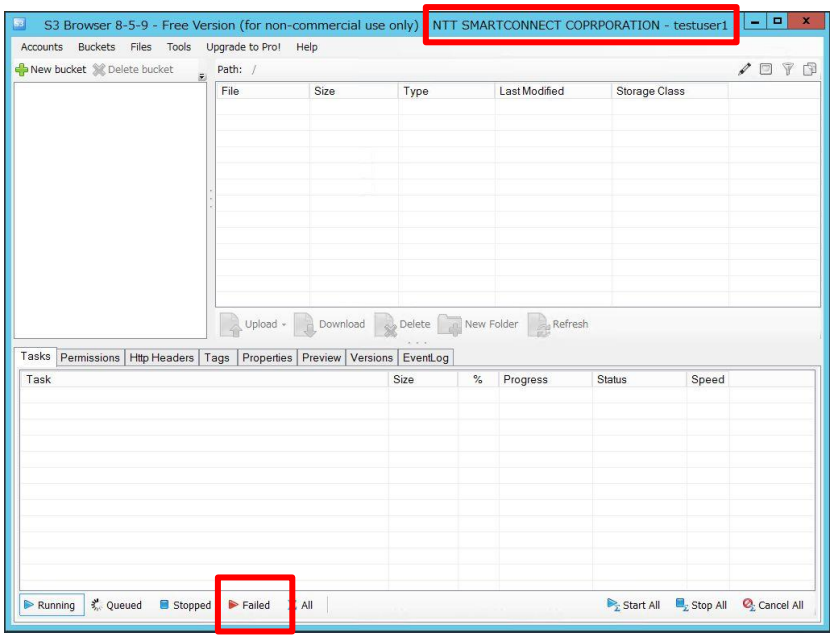
起動すると以下のような画面が表示されますので、以下の通りに入力を行い、**[Add new account]**をクリックします。



※ [smstr.jp] の前提条件として [smstr.jp] 及び [バケット名].smstr.jp が[ストレージアクセス用 IP アドレス]に名前解決可能である必要があります。

※ HTTPS で通信をする場合は [ストレージアクセス用 IP アドレス:443] を入力する必要があります。

ウインドウの枠に名前が表示され、下部にエラーが表示されていなければ、アカウントの追加成功です。



S3 browser の使用

S3 browser 使用してオブジェクトストレージを操作します。下記の URL を参考にしつつ、使用してみましょう。次ページでは操作のいくつかの例を記載しておりますので、ご活用ください。

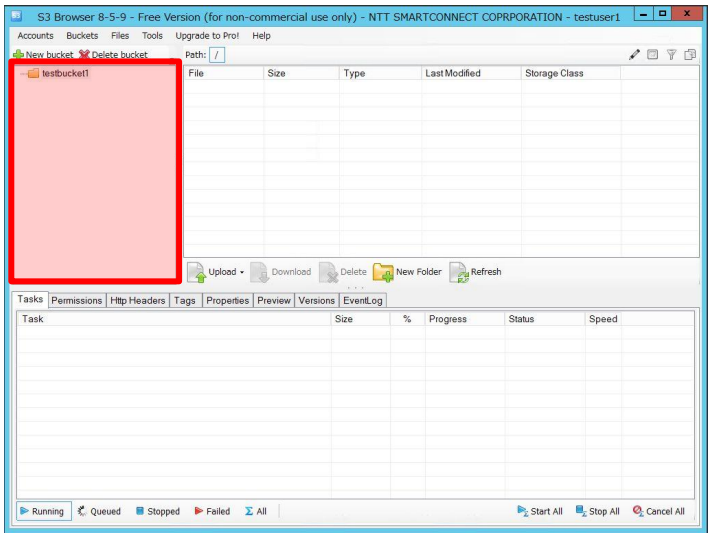
- **S3 browser 操作方法一覧（英語）**

  https://s3browser.com/help.aspx

注：S3browser は、Amazon S3 の機能を基に開発されているため、S3 互換である本サービスでは一部使用できない機能がございます。各機能についてはお客様にて検証及びご確認ください。

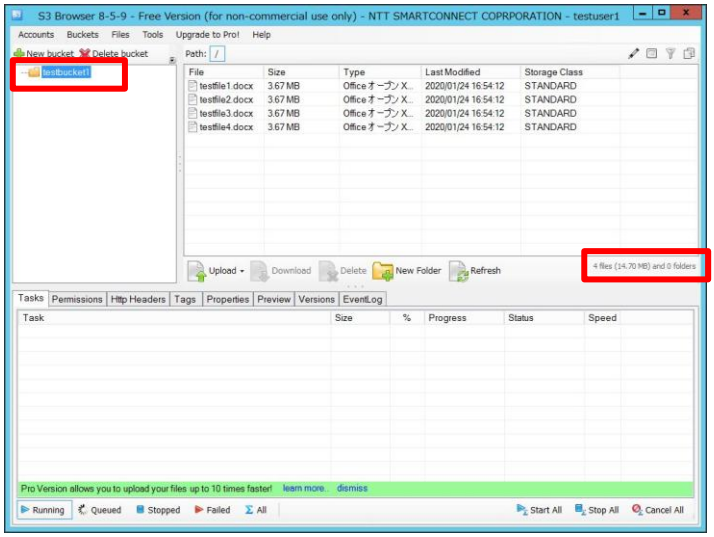
バケット一覧の確認

画面左側にバケット一覧が確認できる。



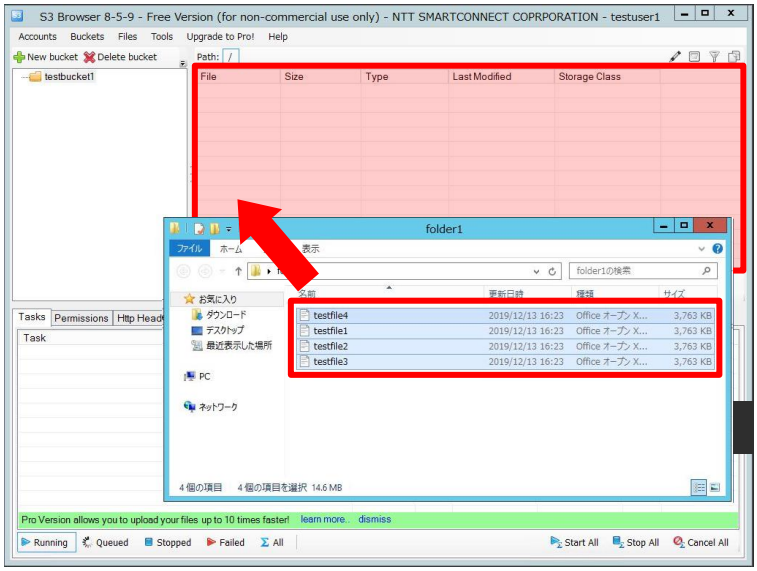
バケット使用容量の確認

バケットを選択し、画面右側中部から確認できる。

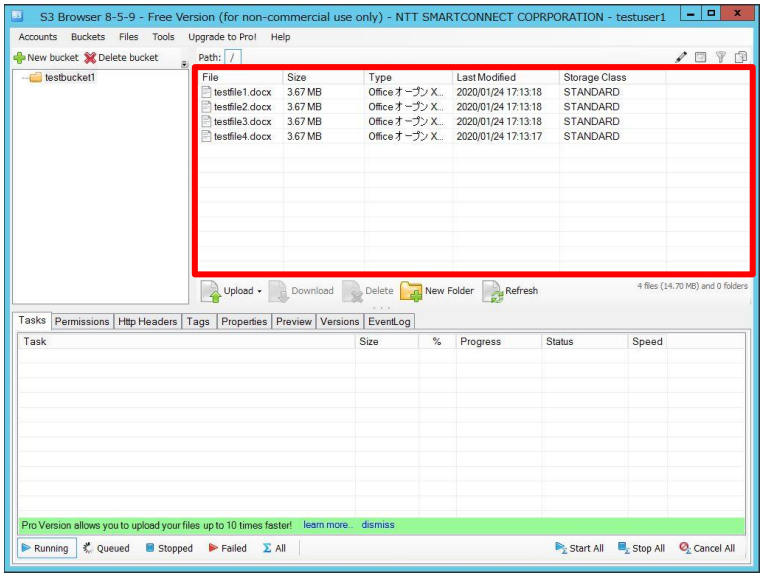


オブジェクトのアップロード

アップロードしたいオブジェクトを選択し、アップロードします。

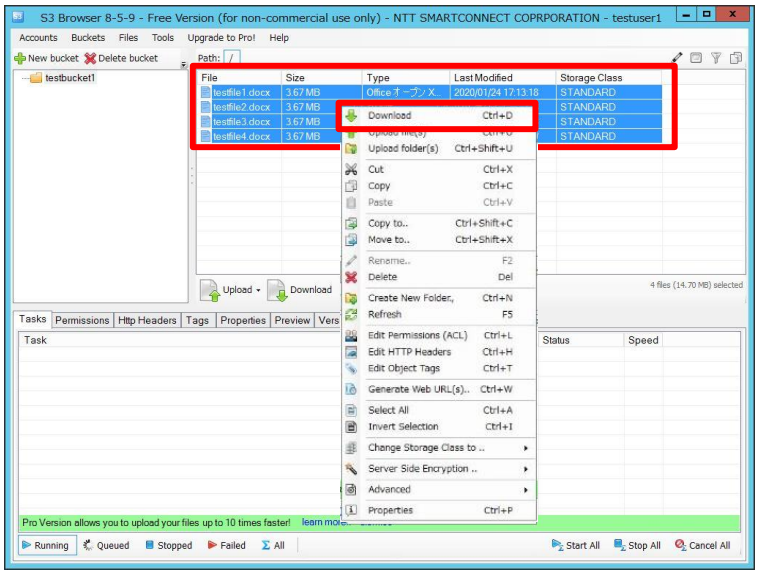


オブジェクトが赤枠内に入っていれば、完了です。

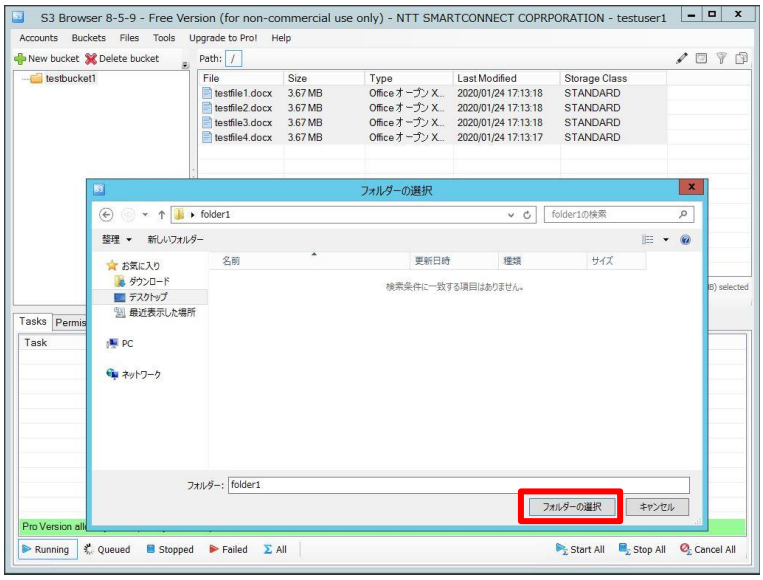


オブジェクトのダウンロード

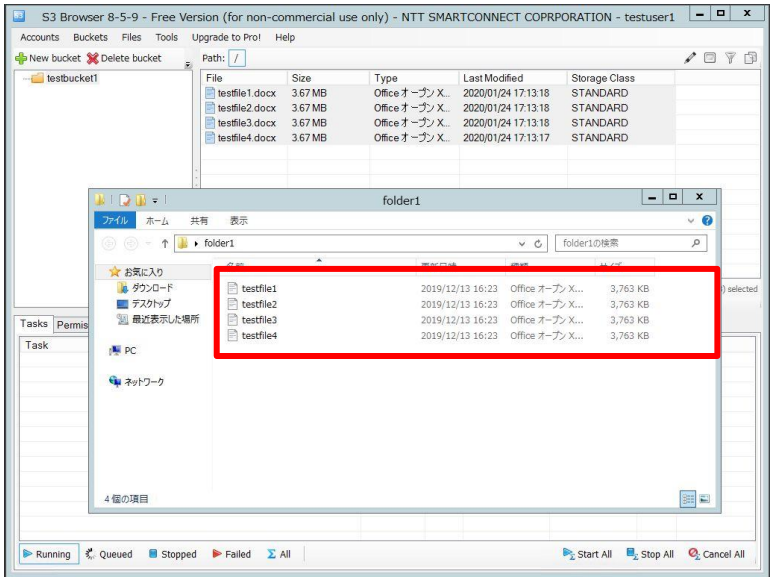
オブジェクトを選択し、右クリックをします。**[Download]**をクリックします。



保存先を選択します。

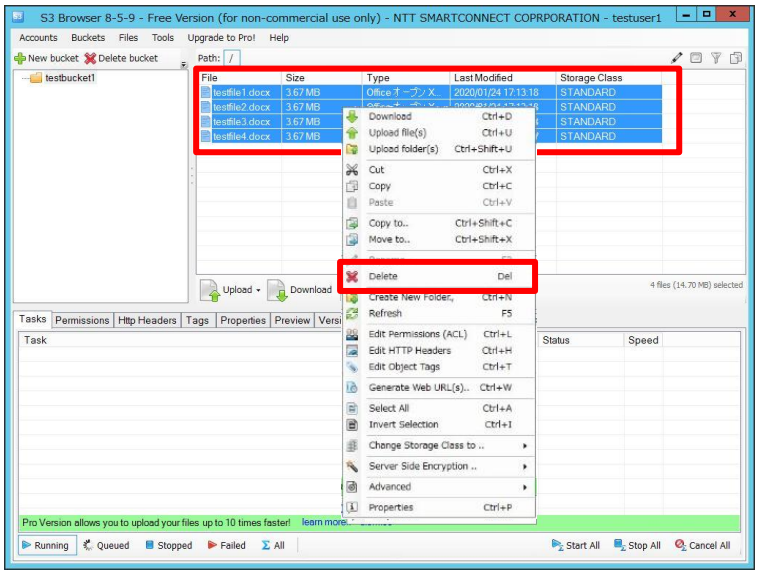


オブジェクトが保管先にあれば完了です。

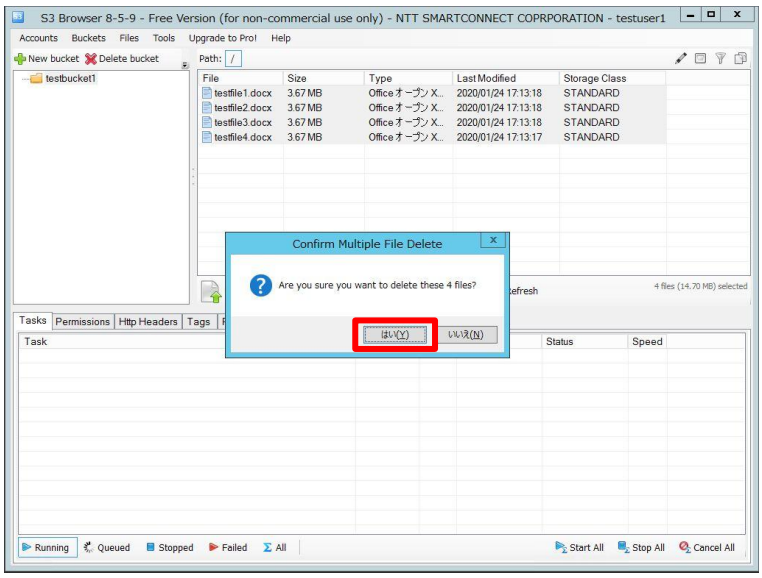


オブジェクトの削除

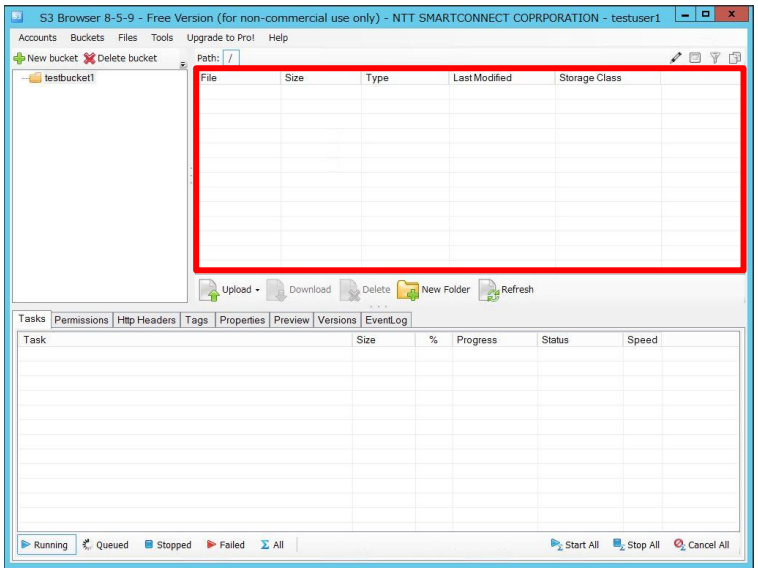
オブジェクトを選択し、右クリックをします。**[Delete]**をクリックします。



[はい(Y)]をクリックします。



オブジェクトが赤枠からなくなれば完了です。



4.アクセスログについて

本章ではアクセスログのご利用方法についてご説明します。

アクセスログのご利用方法

アクセスログは専用のボリュームに保存されております。この専用ボリュームにアクセスすることでアクセスログを読み出すことが可能となります。専用ボリュームへのアクセスは『 サービスご利用の手引き（CIFS 版　Windows 用）』をご参照ください。

注：アクセスには開通通知書のアクセスログ情報に記載されているユーザ名/パスワードが必要となります。

（本章の説明は、Windows10 の端末からのご利用方法について記載いたしますが、他の OS につきましても、説明している内容に沿って実施いただくことで可能となります。）

ログに記録されるイベントは次のとおりです。

- ListBucket
- GetObject
- PutObject
- DeleteObject

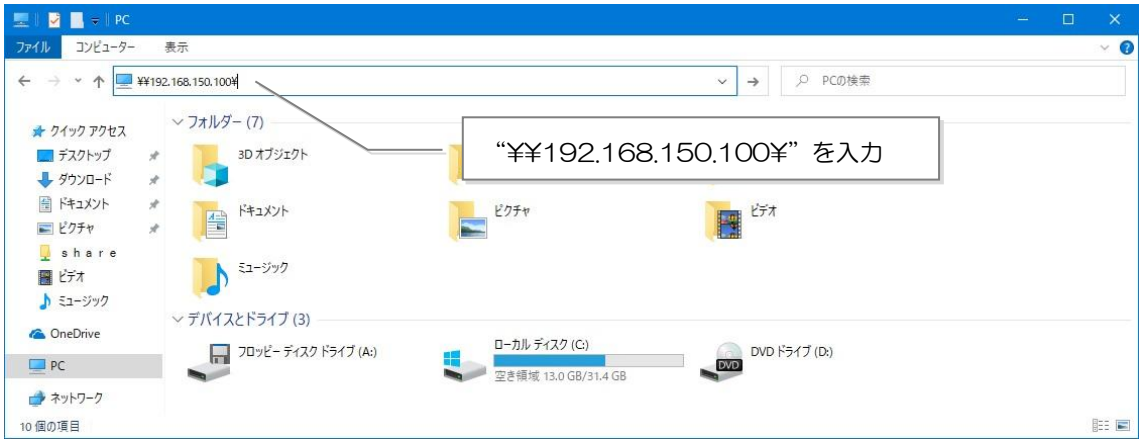
アクセログの閲覧方法

アクセスログはイベント発生に応じて専用のネットワークフォルダー内に出力されます。アクセスログファイルの閲覧には「テキストエディタ」または「テキストビューア」が必要です。また、json 形式を整形して閲覧するには、専用ビューアやソフトウェアをご利用ください。

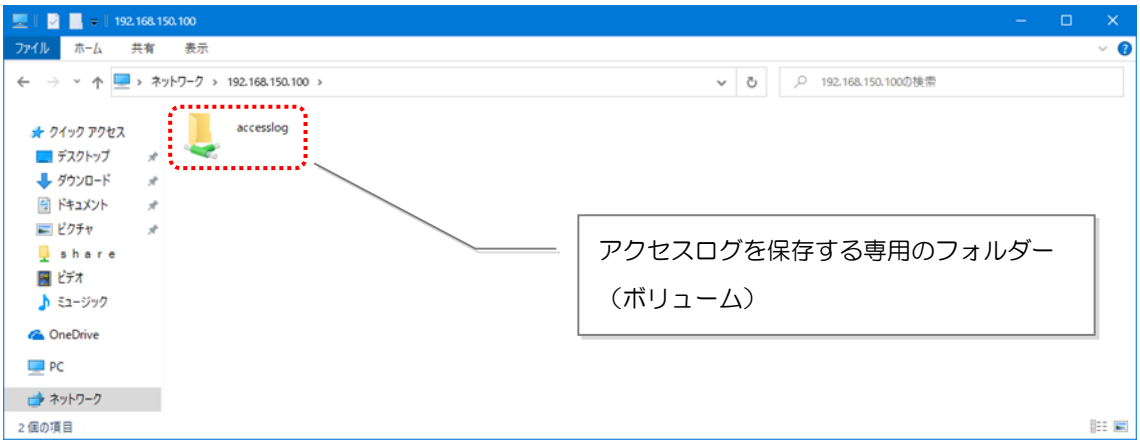
１）アクセスログファイルを閲覧するため、アクセスログが保存されている専用ボリュームに接続します。

専用ボリュームはお客さまがお使いのオブジェクトストレージの中に専用のネットワークフォルダーとして構成されております。

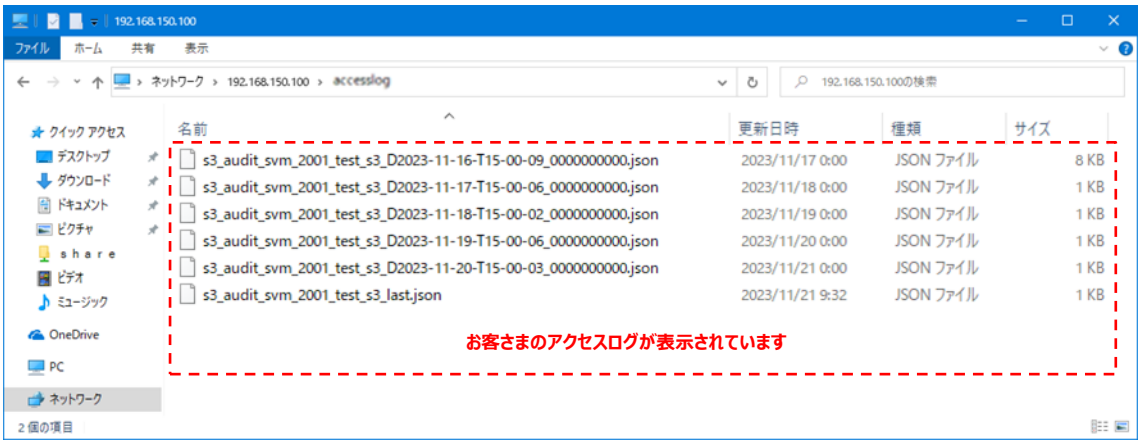
下図は、ストレージアクセス用 IP アドレスが「192.168.150.100」の場合のアクセス方法を示しています。



２）正しくアクセスできると、「accesslog」という名前のネットワークフォルダー（ボリューム）があり、このネットワークフォルダー内にアクセスログが保存されています。



３）「accesslog」フォルダー内に一旦蓄積されたアクセスログファイル(~last.json)と、24 時間毎に出力されたアクセスログファイル(~00000.json)が表示されます。「accesslog」フォルダーは読み取り専用となっております。
アクセスログ「.json ファイル」を任意の「テキストエディタ」や「テキストビューア」などで閲覧ください。または任意のフォルダー等にコピーした後に閲覧ください。



アクセスログの形式

アクセスログのファイル形式は JSON フォーマットのテキスト形式です。 以下はアクセスログの例です。


```
{
        "eventVersion": "1",
        "userIdentity": {
                "type": "Local User",
                "userName": "user001"
        },
        "eventTime": "2024-03-06T01:13:29.383744000Z",
        "eventSource": "svm_1234_test_s3",
        "eventName": "PutObject",
        "sourceIPAddress": "192.0.2.1",
        "eventID": "befdd886-db56-11ee-964a-d039ea075f1b",
        "readOnly": "False",
        "managementEvent": "False",
        "eventCategory": "Data",
        "requestID": "467273291",
        "requestParameters": {
                "bucketName": "bucket001",
                "key": "object.txt"
        },
        "ONTAP": {
                "Result": "Audit Success",
                "VolumeUUID": "ab_cde_1696467415;14529eda-631a-11ee-9067-d039ea075f1b"
        }
}
```


以下は代表的なキーと説明です。

キー		説明
userIdentity		
	userName	アクセス元のユーザ名
eventTime		アクセス日時(UTC)
eventName		オブジェクトアクセスイベント名※
sourceIPAddress		アクセス元の IP アドレス
requestParameters		
	bucketName	アクセスしたバケット名
	key	アクセスしたオブジェクト名

※ログに記録されるイベント（eventName キーの値）は以下のとおりです。

イベント	説明
ListBucket	バケット閲覧
PutObject	オブジェクトのアップロード
GetObject	オブジェクトのダウンロード
DeleteObject	オブジェクトの削除

5.改訂履歴

版数	日付	変更内容
第 1.0 版	2023 年 12 月 1 日	初版リリース
第 1.1 版	2024 年 3 月 8 日	アクセスログを追加
第 1.2 版	2024 年 8 月 7 日	軽微な修正
第 1.3 版	2026 年 9 月 30 日	ドキュメント名称を変更