



# XCP を設定します

## XCP

NetApp  
January 22, 2026

# 目次

XCP を設定します	1
XCP NFS の INI ファイルを設定します	1
ルートユーザの INI ファイルを設定します	1
root 以外のユーザの INI ファイルを設定する	1
パフォーマンスの調整	2
環境変数	2
POSIX コネクタを設定します	3
サポートされている機能	3
パスの構文	3
POSIX コネクタを設定します	3
所有権（UID および GID）	4
オープンファイル記述子の最大数を増やします	5
HDFS コネクタを設定します	5
マルチノードスケールアウトを構成します	6
S3 Connectorを設定します	8
S3コネクタをセットアップします	8

# XCP を設定します

## XCP NFS の INI ファイルを設定します

XCP の INI ファイルを設定する手順。



XCP INI ファイルは XCP SMB では必要ありません。

### ルートユーザの INI ファイルを設定します

次の手順を使用して、XCP NFS root ユーザの INI ファイルを設定できます。

手順

1. vi エディタを使用して、ホスト構成ファイルに XCP サーバのカタログの場所を追加します。



カタログの場所は、「`xcp.ini` XCP」構成ファイルの詳細を変更する前にエクスポートする必要があります。カタログの場所（NFSv3）は XCP Linux ホストにマウント可能である必要がありますが、マウントされているとは限りません。

```
[root@localhost /]# vi /opt/NetApp/xFiles/xcp/xcp.ini
```

2. カタログの XCP Linux クライアントホスト構成ファイルのエントリが変更されたことを確認します。

```
[root@localhost /]# cat /opt/NetApp/xFiles/xcp/xcp.ini
# Sample xcp config
[xcp]
catalog = 10.61.82.210:/vol/xcpvol/
```

### root 以外のユーザの INI ファイルを設定する

root 以外のユーザとして、NFS ファイルシステムをマウントする権限がありません。最初にカタログボリュームをマウントし、XCP を実行している root 以外のユーザとして XCP を実行するには root ユーザが必要です。カタログボリュームに対する読み取り / 書き込み権限を持っている場合は、POSIX コネクタを使用してマウントされたカタログボリュームにアクセスできます。ボリュームがマウントされたら、カタログにパスを追加できます。

```
(t/10.237.170.53_catalog_vol - This is the path where catalog volume is  
mounted) as follows.
```

```
[user1@scspr2474004001 xcp]$ ls -ltr  
total 8  
drwxrwxr-x 2 user1 user1 21 Sep 20 02:04 xcplogs  
-rw-rw-r-- 1 user1 user1 71 Sep 20 02:04 xcp.ini  
-rwxr-xr-x 1 user1 user1 352 Sep 20 02:10 license  
[user1@scspr2474004001 xcp]$ cat /home/user1/NetApp/xFiles/xcp/xcp.ini  
  
Sample xcp config [xcp]  
catalog = file:///t/10.237.170.53_catalog_vol
```

## パフォーマンスの調整

XCP NFS の場合は、「How」コマンドと「Can」コマンドを使用して移行を計画した後で、データを移行できます。



root 以外のユーザとしてデータ移行を実行する場合は、root ユーザが次の手順を実行できます。

最適なパフォーマンスと信頼性を実現するために、XCP Linux クライアントホストの /etc/sysctl.conf に次の Linux カーネル TCP パフォーマンスパラメータを設定することを推奨します。「sysctl -p」または「reboot」コマンドを実行して変更をコミットします。

```
net.core.rmem_default = 1342177  
net.core.rmem_max = 16777216  
net.core.rmem_max = 16777216  
net.core.wmem_default = 1342177  
net.core.wmem_max = 16777216  
net.ipv4.tcp_rmem = 4096 1342177 16777216  
net.ipv4.tcp_wmem = 4096 1342177 16777216  
net.core.netdev_max_backlog = 300000  
net.ipv4.tcp_fin_timeout = 10
```



root 以外のユーザの場合は、root ユーザが設定を実行する必要があります。

## 環境変数

XCP NFS システムの環境変数のオプション設定。



root 以外のユーザは、次の変数も使用できます。

環境変数XCP\_CONFIG\_DIRは'デフォルトの場所である/opt/NetApp/xFiles/xcpを上書きしますこの値を設定する場合は、マウントされている NFS ディレクトリなど、OS ファイルシステムのパスを指定します。'XCP CONFIG\_DIR'変数が設定されている場合'ホスト名と同じ名前の新しいディレクトリがカスタム構成ディレクトリパス内に作成され'新しいログはこの場所に格納されます

```
[root@localhost /]# export XCP_CONFIG_DIR ='/tmp/xcp_config_dir_path'
```

環境変数のXCPログが設定ディレクトリに格納されるデフォルトの場所は'XCPログディレクトリによって上書きされますこの値を設定する場合は、マウントされている NFS ディレクトリなど、OS ファイルシステムのパスを指定します。'XCP\_LOG\_DIR'変数を設定すると'ホスト名と同じ名前の新しいディレクトリがカスタム・ログ・ディレクトリ・パス内に作成され'新しいログはこの場所に格納されます

```
[root@localhost /]# export XCP_LOG_DIR='/tmp/xcp_log_dir_path'
```

環境変数XCPカタログパスは、xcp.inの設定よりも優先されます設定する場合、値は XCP パス形式である「server:export[:subdirectory]」でなければなりません。

```
[root@localhost /]# export XCP_CATALOG_PATH='10.61.82.210:/vol/xcpvol/'
```



root以外のユーザの場合は、エクスポートされたパスの「XCP カタログパス」を POSIX パスに置き換える必要があります。

## POSIX コネクタを設定します

XCP NFS では、POSIX コネクタを使用したデータ移行のソースパスとデスティネーションパスの提供がサポートされるようになりました。

### サポートされている機能

POSIX コネクタでは、次の機能がサポートされています。

- nanosecond`atim`、`imtime`、および `ctime` をサポートする POSIX ファイルシステムでは "can command" は完全な値（秒とナノ秒）を取得し 'copy' コマンドはそれらを設定します
- POSIX コネクタは、NFSv3 TCP ソケットを使用する XCP よりも安全です。

### パスの構文

POSIX コネクタのパス構文は 'file://<Linux 上のマウントパス>' です

## POSIX コネクタを設定します

POSIX コネクタを設定するには、次のタスクを実行する必要があります。

- ソースボリュームとデスティネーションボリュームをマウント

- ・デスティネーションパスにデータを書き込むために必要な権限があることを確認してください

次の例では、デスティネーションとカタログがマウントされています。

```
root@scspr2395903001 ~]# findmnt -t nfs4
TARGET SOURCE FSType OPTIONS
/t/10.237.170.39_src_vol 10.237.170.39:/source_vol nfs4
rw,relatime,vers=4.0,rsize=65536,wsize=65536,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clien
/t/10.237.170.53_dest_vol 10.237.170.53:/dest_vol nfs4
rw,relatime,vers=4.0,rsize=65536,wsize=65536,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clien
/t/10.237.170.53_catalog_vol 10.237.170.53:/xcp_catalog nfs4
rw,relatime,vers=4.0,rsize=65536,wsize=65536,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clien
[root@scspr2395903001 ~]#
```

POSIX コネクタは 'file://' という POSIX 構文を使用して 'ソースボリュームとデスティネーションボリュームにアクセスします' 上の例では、ソースパスは 「file:///t/10.237.170.39\_src\_vol」 で、デスティネーションパスは 「[file:///t/t](#)/10.237.170.53\_dest\_vol」 です。

root 以外のユーザが共有する XCP カタログの構成例を管理するには、XCP ユーザ用の Linux グループを作成します。root 以外のユーザの場合、Linux グループユーザが移行を実行するには、次の権限が必要です。

次の出力例では、「demo」は root 以外のユーザ、「/mnt/XCP-catalog」はカタログボリュームがマウントされているパスです。

```
sudo groupadd -g 7711 xcp_users
sudo usermod -G xcp_users -a demo
sudo chown -R :xcp_users /mnt/xcp-catalog
sudo chmod -R g+w /mnt/xcp-catalog
```

XCP カタログにはデータは保存されませんが、スキャンおよびコピーのファイル名、ディレクトリ名、およびその他のメタデータは保存されます。したがって、ユーザに格納されたメタデータの保護機能を付与するために、許可されたユーザにカタログファイルシステムの権限を設定することを推奨します。

## 所有権（UID および GID）

通常のユーザとして設定されている場合、デフォルトでは POSIX または NFS3 宛先への「copy」コマンドは所有権（ユーザ ID（UID）およびグループ ID（GID）の設定を試みません。所有権の設定は、通常は管理者が行います。ユーザ A がユーザ B からファイルをコピーする場合、ユーザ A は宛先を所有することを想定します。ただし、root ユーザがファイルをコピーする場合は、このようなことはありません。root ユーザがファイルをコピーすると '-down' オプションによって動作が変更され '-chown' を指定した root 以外の copy コマンドが UID と GID の設定を試みます

## オープンファイル記述子の最大数を増やします

最適なパフォーマンスと信頼性を実現するために、すべてのノードで XCP ユーザのオープンファイル記述子の最大数を増やすことができます。

### 手順

1. ファイルを開くには `vi /etc/security/limits.conf` コマンドを使用します
2. ファイルに次の行を追加します :`<username>-nofile 999999`
  - 例 \*

```
root - nofile 999999
```

を参照してください "Red Hat ソリューション" を参照してください。

## HDFS コネクタを設定します

XCP NFS の場合、 Hadoop Distributed File System ( HDFS ) Connector ( HDFS : // ) は、 XCP に、異なるベンダーが提供するすべての HDFS ファイルシステムへのアクセスを許可します。

### サポートされている機能

HDFS から NFS への「copy」コマンド操作は、 HDFS コネクタでサポートされています。

### パスの構文

HDFS コネクタのパス構文は、「 hdfs://[user@host : port]/full-path 」です。



ユーザ、ホスト、およびポートを指定しない場合、 XCP はホストを「デフォルト」に設定し、ポートを「 0 」に設定した「 hdfsConnect 」を呼び出します。

### HDFSコネクタをセットアップする

hdfs`copy` コマンドを実行するには、 Linux システム上に HDFS クライアントを設定する必要があります。また、 Hadoop ベンダーに基づいて、インターネット上で利用可能なセットアップ構成に従います。たとえば、 MapR クラスタのクライアントを設定するには、「 <https://docs.datafabric.hpe.com/60/AdvancedInstallation/SettingUptheClient-redhat.html> 」を使用します。

HDFS クライアントのセットアップが完了したら、クライアントの設定を完了する必要があります。 XCP コマンドで HDFS パスを使用するには、次の環境変数が必要です。

- Nhdfs\_LIBhdfs\_path です
- Nhdfs\_libjvm\_path

次の例では、設定は、 CentOS 上の MapR および Java -1.8.0-openjdk-devel と連携しています。

```
export JAVA_HOME=$(dirname $(dirname $(readlink $(readlink $(which javac)))))  
export NHDFS_LIBJVM_PATH=`find $JAVA_HOME -name "libjvm.so"` export  
NHDFS_LIBHDFS_PATH=/opt/mapr/lib/libMapRClient.so
```

```
[demo@mapr0 ~]$ hadoop fs -ls Found 3 items  
drwxr-xr-x - demo mapr 0 2021-01-14 00:02 d1  
drwxr-xr-x - demo mapr 0 2021-01-14 00:02 d2  
drwxr-xr-x - demo mapr 0 2021-01-14 00:02 d3
```

## マルチノードスケールアウトを構成します

XCP NFS の場合、1つのノードのパフォーマンス制限を克服するには、単一の「copy」（または「CAN-MD5」）コマンドを使用して、複数の Linux システムまたはクラスタノード上でワーカーを実行します。

サポートされている機能

マルチノードスケールアウトは、次のような場合など、単一システムのパフォーマンスでは不十分な環境に役立ちます。

- ・ペタバイトのデータをコピーするのにノードが1つの場合、数ヶ月かかるとします
- ・クラウドオブジェクトストレージへの高レイテンシ接続が原因で、個々のノードの速度が低下している場合
- ・大量の I/O 処理を実行する大規模な HDFS クラスタファームでは

パスの構文

マルチノードスケールアウトのパス構文は '--nodes worker1' 'worker2' 'worker3' です

マルチノードのスケールアウトをセットアップします

同様の CPU と RAM 構成の 4 台の Linux ホストを使用したセットアップを考えてみてください。移行には 4 つのホストすべてを使用できます。XCP はすべてのホストノードでコピー処理を調整できるためです。スケールアウト環境でこれらのノードを利用するには、4つのノードのいずれかをマスターノード、他のノードをワーカーノードとして識別する必要があります。たとえば、Linux の 4 ノードセットアップの場合、ノードに「master」、「worker1」、「worker2」、および「worker3」という名前を付け、マスターノードで構成をセットアップします。

1. ホームディレクトリに XCP をコピーします。
2. XCP ライセンスをインストールしてアクティブにします。
3. xcp.ini` ファイルを変更し 'カタログ・パスを追加します
4. マスターノードからワーカーノードにパスワードなしの Secure Shell (SSH) を設定します。
  - a. マスターノードでキーを生成します。

「ssh-keygen -b 2048 -t rsa-f/root/.ssh/id\_rsa-q-N」

- b. すべてのワーカーノードにキーをコピーします。

```
「ssh-copy-id -i /root/.ssh/id_rsa.pub root@worker1」
```

XCP マスターノードは、SSH を使用して他のノードでワーカーを実行します。マスターノードで XCP を実行しているユーザに対してパスワードを使用しない SSH アクセスを有効にするようにワーカーノードを設定する必要があります。たとえば、マスターノードで XCP ワーカーノードとしてノード「worker1」を使用するためのユーザデモを有効にするには、マスターノードからホームディレクトリ内のすべてのワーカーノードに XCP バイナリをコピーする必要があります。

#### MaxStartupsのことです

複数の XCP ワーカーを同時に起動してエラーを回避するには、次の例に示すように、各ワーカーノードで「SHD MaxStartups」パラメータを増やす必要があります。

```
echo "MaxStartups 100" | sudo tee -a /etc/ssh/sshd_config  
sudo systemctl restart sshd
```

#### 「nodes.ini」ファイル

クラスタノードで XCP を実行すると、ワーカープロセスはマスターノードのメイン XCP プロセスから環境変数を継承します。特定のノード環境をカスタマイズするには'マスターノード上の構成ディレクトリ内の nodes.ini` ファイルに変数を設定する必要があります (ワーカーノードには構成ディレクトリまたはカタログがありません) たとえば、Wave ( CentOS ) などのマスターノードとは別の場所に「libjvm.so」を置いている Ubuntu サーバ MARS の場合、MARS 上のワーカーが HDFS コネクタを使用できるようにするには、設定ディレクトリが必要です。このセットアップの例を次に示します。

```
[schay@wave ~]$ cat /opt/NetApp/xFiles/xcp/nodes.ini [mars]  
NHDFS_LIBJVM_PATH=/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/  
amd64/server/libjvm.so
```

POSIX ファイルパスと HDFS ファイルパスを使用してマルチセッションを実行する場合は、ファイルシステムとソースおよびデスティネーションのエクスポート済みファイルシステムをマスターノードとすべてのワーカーノードにマウントする必要があります。

ワーカーノードで XCP を実行する場合、ワーカーノードにはローカル構成はありません（ライセンス、ログファイル、またはカタログはありません）。XCP バイナリが必要なのは、ホームディレクトリ内のシステムのみです。たとえば 'copy' コマンドを実行するには'マスター・ノードとすべてのワーカー・ノードがソースとデスティネーションにアクセスする必要があります' xCP copy — ノード linux1、linux2 hdfs://user/ceme/test/file:/mnt/ontap `、linux1` および `linux2` hosts には HDFS クライアントソフトウェアが設定されており、/mnt/ontap に NFS エクスポートがマウントされている必要があります。前述のように、XCP バイナリのコピーがホームディレクトリに格納されている必要があります。

**POSIXコネクタとHDFSコネクタ、マルチノードのスケールアウト、セキュリティ機能を組み合わせて使用できます**

POSIX コネクタと HDFS コネクタ、マルチノードスケールアウト、セキュリティ機能を組み合わせて使用できます。たとえば、次の「copy」コマンドと「verify」コマンドは、POSIX コネクタと HDFS コネクタをセキュリティおよびスケールアウト機能と組み合わせたものです。

- 「copy」コマンドの例：

```
./xcp copy hdfs:///user/demo/d1 file:///mnt/nfs-server0/d3  
./xcp copy -match "'USER1 in name'" file:///mnt/nfs-server0/d3  
hdfs:///user/demo/d1  
./xcp copy -node worker1,worker2,worker3 hdfs:///user/demo/d1  
file:///mnt/nfs-server0/d3
```

- 「`verify`」コマンドの例：

```
./xcp verify hdfs:///user/demo/d2 file:///mnt/nfs-server0/d3
```

## S3 Connectorを設定します

XCP 1.9.2以降では、Simple Storage Service (S3) コネクタによってHadoop Distributed File System (HDFS) ファイルシステムからS3オブジェクトストレージへのデータ移行が可能になり、XCPデータ移行の範囲が拡張されました。

サポートされる移行のユースケース

S3コネクタの移行のユースケースは次のとおりです。

- HDFSからNetApp StorageGRIDへの移行
- HDFSからAmazon S3への移行
- HDFSからNetApp ONTAP S3への移行



現在、MapRはHDFSでのみ認定され、サポートされています。

サポートされている機能

のサポート `scan`、`copy`、`verify`、`resume` および `delete` S3コネクタ用のコマンドを使用できます。

サポートされていない機能

のサポート `sync` コマンドはS3コネクタに対しては使用できません。

パスの構文

S3コネクタのパス構文は、です `s3://<bucket in S3>`。

- を使用して、XCPコマンドに特定のS3プロファイルを指定できます `-s3.profile` オプション
- を使用できます `s3.endpoint` S3と通信するためにエンドポイントの値を変更するオプション



エンドポイントの使用は、StorageGRIDおよびONTAP S3では必須です。

## S3コネクタをセットアップします

手順

- S3コネクタを使用してXCPコマンドを実行するには、各プラットフォームのオンラインドキュメントに従ってS3にバケットを作成します。
  - "ONTAP S3オブジェクトストレージの管理"
  - "StorageGRID：テナントアカウントの概要を使用します"



続行する前に、が必要です access key、secret key、認証局（CA）証明書バンドル、および endpoint url 情報。XCPは、処理を開始する前に、これらのパラメータを使用してS3バケットを識別して接続します。

- Amazon Web Services (AWS) CLIパッケージをインストールし、AWS CLIコマンドを実行してS3アカウントのキーとSecure Sockets Layer (SSL) 証明書を設定します。
  - を参照してください "AWS CLIの最新バージョンをインストールまたは更新します" をクリックしてAWSパッケージをインストールします。
  - を参照してください "AWS CLI Command Referenceを参照してください" を参照してください。
- を使用します aws configure コマンドを使用してクレデンシャルファイルを設定します。デフォルトでは、ファイルの場所はです /root/.aws/credentials。クレデンシャルファイルにアクセスキーとシークレットアクセスキーを指定する必要があります。
- を使用します aws configure set コマンドを使用してCA証明書バンドルを指定します。これは、のファイルです .pem SSL証明書の検証時に使用される拡張子。デフォルトでは、ファイルの場所はです /root/.aws/config。

◦ 例：\*

```
[root@client1 ~]# aws configure
AWS Access Key ID [None]: <access_key>
AWS Secret Access Key [None]: <secret_key>
Default region name [None]:
Default output format [None]:
[root@client1 ~]# cat /root/.aws/credentials
[default]
aws_access_key_id = <access_key>
aws_secret_access_key = <secret_key>
[root@client1 ~]#
[root@client1 ~]# aws configure set default.ca_bundle
/u/xxxx/s3/ca/aws_cacert.pem
[root@client1 ~]# cat /root/.aws/config
[default]
ca_bundle = /u/xxxx/s3/ca/aws_cacert.pem
```

- 必要なセットアップ設定が完了したら、XCPコマンドを実行する前に、AWS CLIコマンドがLinuxクライアントからS3バケットにアクセスできることを確認します。

```
aws s3 ls --endpoint-url <endpoint_url> s3://bucket-name/
```

```
aws s3 ls --profile <profile> --endpoint-url <endpoint_url> s3://bucket-name/
```

◦ 例：\*

```
[root@client1 linux]# aws s3 ls --profile <profile> --endpoint
<endpoint_url>  s3://<bucket-name>
                  PRE 1G/
                  PRE aws_files/
                  PRE copied_folders/
                  PRE d1/
                  PRE d2/
                  PRE giant_size_dirs/
                  PRE medium_size_dirs/
                  PRE small_size_dirs/
[root@client1 l
```

## 著作権に関する情報

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を隨時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5225.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用権を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用権については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

## 商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。