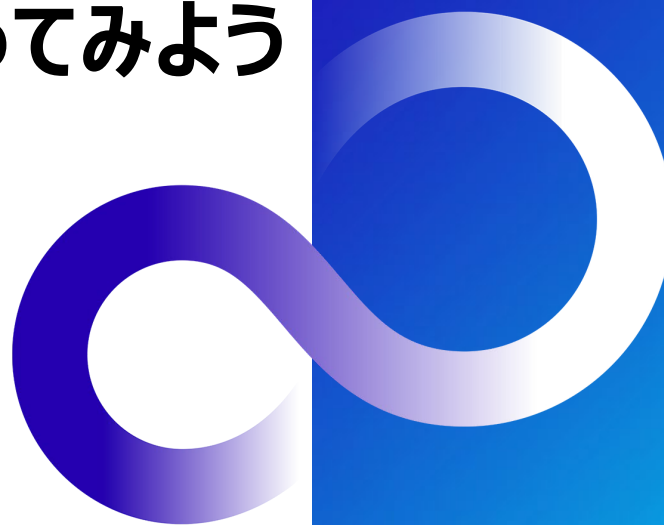


Oracle Solaris 11 ZFSを試してみよう (構築・運用ガイド)

2019年6月 (第5.0版)

富士通株式会社



■ 目的

- Oracle Solaris 11の標準ファイルシステムであるZFS（Zettabyte（ゼタバイト） File System）の構築・運用方法を紹介します。

■ 対象読者

- Oracle SolarisとRAIDの基礎知識を有している方
- 『Oracle Solaris 11 ZFSを使ってみよう（概要・設計ガイド）』を参照された方

■ 留意事項

- 本資料では、Solaris環境をOracle VM Server for SPARC上に構築した場合の手順を記載しています。一部、Oracle VM Server for SPARCに依存した内容がありますのでご了承ください。
例：ディスクやネットワークインターフェースなどのデバイス名がOracle VM Server for SPARC固有の値となっており、物理サーバ環境とは異なります。
- 本資料は、Oracle Solaris 11.4に基づいています。
Oracle Solaris 11 に関する最新情報は、Oracle社のマニュアルをご参照ください。
 - Oracle Solaris Documentation
<https://docs.oracle.com/en/operating-systems/solaris.html>

■ ドキュメントの位置づけ

- Oracle Solaris ZFSを使ってみよう
<https://www.fujitsu.com/jp/sparc-technical/document/solaris/#solaris11zfs>



※ 本書は、『Oracle Solaris 11 ZFSを使ってみよう（構築・運用手順書）』とあわせてお読みください。

■ 本書での表記

- コマンドのセクション番号は省略しています。

例：

- ls(1) ⇒ lsコマンド
- shutdown(1M) ⇒ shutdownコマンド

- 以下の用語は略称を用いて表記する場合があります。

略称	正式名称
Solaris	Oracle Solaris
Solarisゾーン	Oracle Solarisゾーン
GDS	PRIMECLUSTER GD

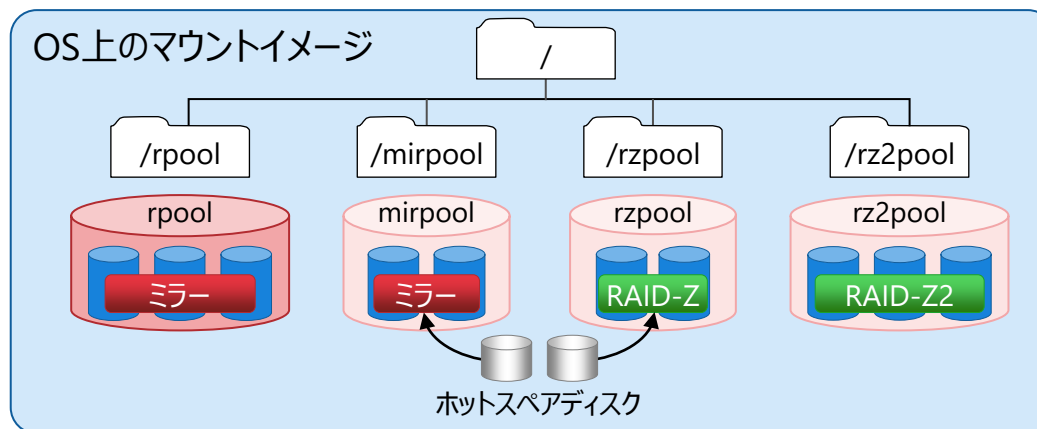
1. ストレージプールの構築
 2. ファイルシステムの構築
 3. スナップショットとクローンの作成
 4. バックアップ／リストア
 5. ミラーディスクの分割
- 参考情報
 - 付録

1.ストレージプールの構築

ストレージプールの構成変更方法や作成方法などを説明します。

■ 作成するストレージプールの環境

本章では、ストレージプールの構成の変更方法や作成方法などを説明します。
構築するストレージプールの構成は、以下のとおりです。



■ ストレージプールの操作

- ルートプール（システム領域）の構成変更
ルートプール「rpool」にディスクを追加し、ミラー構成へ変更します。
- ストレージプール（ユーザー領域）の作成
以下のストレージプール（ユーザー領域）を作成します。
 - ミラー（RAID1）構成：mirpool
 - RAID-Z構成：rzpool
 - RAID-Z2構成：rz2pool
- ホットスペアディスクの登録
「mirpool」と「rzpool」に、ホットスペアディスクを登録します。
- ストレージプールの確認
「rpool」を例にして、ストレージプールの状態を確認します。

ルートプール（システム領域）の構成変更

■ シングル構成にディスクを追加し、多面ミラー構成に変更します。

- デフォルト（OSインストール時）のルートプールの構成は、単一のディスクです。
OSのインストール後に、シングル構成からミラー構成に変更できます。

✓ ミラーディスクの追加（zpool attachコマンド）

【書式】 zpool attach <プール名> <ミラー元ディスク> <ミラーディスク>

- 単一ディスク構成から2面ミラー構成へ

```
# zpool attach rpool c2d0s0 c2d1s0
```



- 2面ミラー構成から3面ミラー構成へ

```
# zpool attach rpool c2d0s0 c2d2s0
```

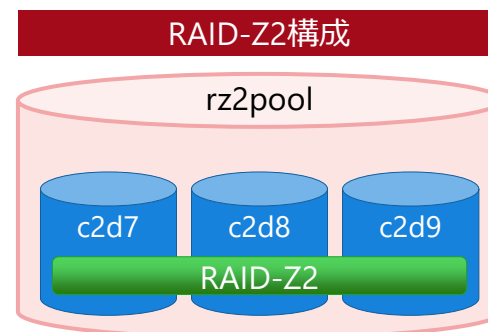
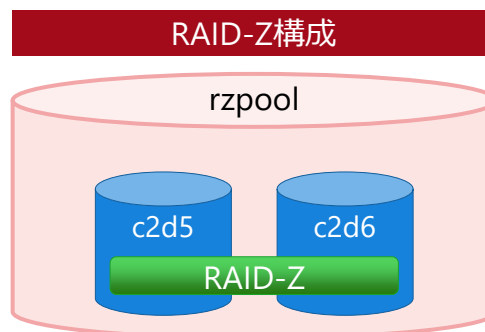
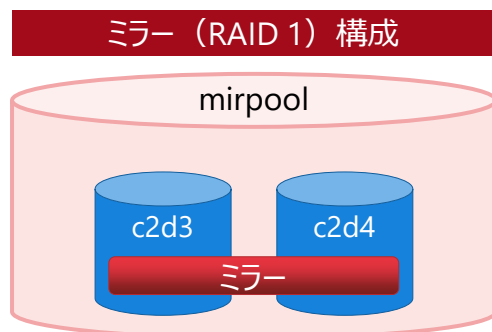


- SPARC M12/M10は、OSインストール時のルートプールのディスクラベルがデフォルトでEFI（GPT）ラベルになります。
※SPARC M10の場合は、XCP2230以降です。XCP2230以前の場合は、ルートプールにSMI（VTOC）ラベルを使用する必要があります。

ストレージプール（ユーザー領域）の作成

■ ストレージプール（ユーザー領域）を作成します。

- ストレージプールのRAIDレベル（RAID 1, RAID-Z, RAID-Z2など）は、ストレージプール（ユーザー領域）の作成時に指定します。



✓ ストレージプールの作成（zpool createコマンド）

【書式】zpool create <プール名> [RAID] <デバイス名> ...

※ [RAID]を指定しない場合は、ストライプ構成となります。

● ミラー（RAID 1）構成

```
# zpool create mirpool mirror c2d3 c2d4
```

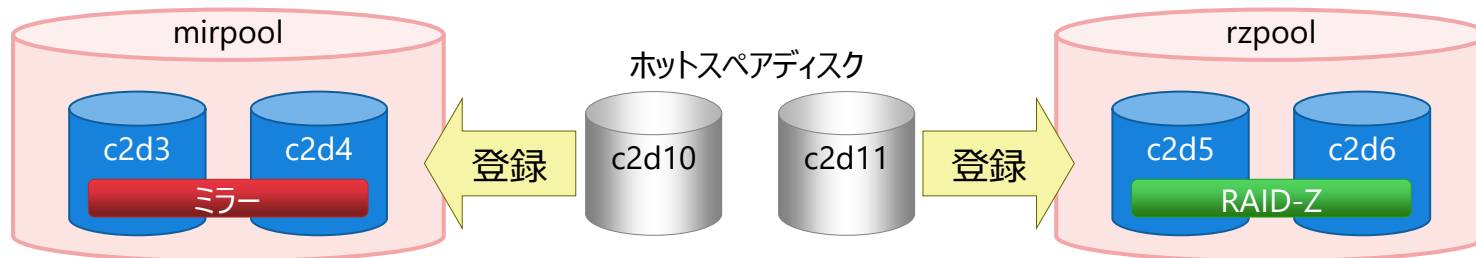
● RAID-Z構成

```
# zpool create rzpool raidz c2d5 c2d6
```

● RAID-Z2構成

```
# zpool create rz2pool raidz2 c2d7 c2d8 c2d9
```


- ストレージプールのディスク故障に備えて、ホットスペアディスクを設定できます。



✓ ホットスペアディスクの登録 (zpool addコマンド)

【書式】 zpool add <プール名> spare <デバイス名> ...

```
# zpool add mirpool spare c2d10  
# zpool add rzpool spare c2d11
```

✓ ホットスペアディスクの解除 (zpool removeコマンド)

【書式】 zpool remove <プール名> <ホットスペアデバイス名> ...

```
# zpool remove mirpool c2d10  
# zpool remove rzpool c2d11
```



• 複数のストレージプールでホットスペアディスクを共有しないでください。データが破損する可能性があります。

■ ストレージプールの状態確認には、主に以下のコマンドを使用します。

- zpool listコマンド
ストレージプールの名称や使用容量など、基本的な情報を確認します。
- zpool statusコマンド
上記に加え、エラーデータや障害の有無などの詳細情報を確認します。

✓ ストレージプールの基本情報の確認（zpool listコマンド）

```
# zpool list
```

NAME	SIZE	ALLOC	FREE	CAP	DEDUP	HEALTH	ALTROOT
mirpool	29.8G	104K	29.7G	0%	1.00x	ONLINE	-
rpool	49.8G	14.6G	35.2G	29%	1.00x	ONLINE	/mnt
rz2pool	89.5G	225K	89.5G	0%	1.00x	ONLINE	-
rzpool	59.5G	216K	59.5G	0%	1.00x	ONLINE	-

① ② ③ ④ ⑤ ⑥ ⑦ ⑧

出力形式

- ① : ストレージプール名
- ② : ストレージプールの合計サイズ
- ③ : 割り当てられている物理容量
- ④ : 割り当てられていない容量
- ⑤ : 使用されているディスク容量の割合 (%)
- ⑥ : 重複除外された量

⑦ : 状態

- ONLINE : 正常な状態
- OFFLINE : 管理者が手動でオフラインにした状態
- DEGRADED : 仮想デバイスに障害が発生しているが、使用可能な状態
- UNAVAIL : デバイスや仮想デバイスにアクセスできない状態
- REMOVED : システム稼動中にデバイスが物理的に取り出された状態

⑧ : ZFS代替ルートプール、代替ルートプールのマウントポイント

※ 代替ルートプールは、ルートプールから起動できなくなった場合に、ルートプールの代替として起動するためのブートイメージです。

✓ ストレージプールの詳細情報の確認 (zpool statusコマンド)

```
# zpool status
pool: rpool ①
state: ONLINE ②
scan: resilvered 12.6G in 2m49s with 0 errors on Fri Mar 1 15:37:54 2019 ③

config:
  ④      ⑤      ⑥      ⑦      ⑧
  NAME    STATE    READ  WRITE  CKSUM
  rpool
    mirror-0
      c2d0s0  ONLINE    0      0      0
      c2d1s0  ONLINE    0      0      0
      c2d2s0  ONLINE    0      0      0

errors: No known data errors ⑨
```

出力形式

- ① : ストレージプール名
- ② : ストレージプールの状態
- ③ : スクラブおよび再同期化の状態
- ④ : ストレージプール名、RAID、ディスク名
- ⑤ : 状態
- ⑥ : 読み込みエラー数
- ⑦ : 書き込みエラー数
- ⑧ : チェックサムエラー数
- ⑨ : エラー情報

正常の場合は、「No known data errors」（既知のデータエラーはありません）が出力されます。

■ ストレージプールのプロパティ情報を確認できます。

✓ ストレージプールのプロパティ情報の確認 (zpool getコマンド)

```
# zpool get all rpool
NAME      PROPERTY          VALUE          SOURCE
rpool     allocated         14.6G          -
rpool     altroot           -              default
~ (省略) ~
rpool     size              49.8G          -
rpool     version           44             default
```

✓ 指定したプロパティ情報を個別に確認

プロパティを「,」（カンマ）で区切りながら指定します。

```
# zpool get bootfs,listsnapshots rpool
NAME      PROPERTY          VALUE          SOURCE
rpool     bootfs            rpool/ROOT/be01 local
rpool     listsnapshots     off            default
```

指定したプロパティのみ
表示されます。

《参考》ストレージプールのプロパティ

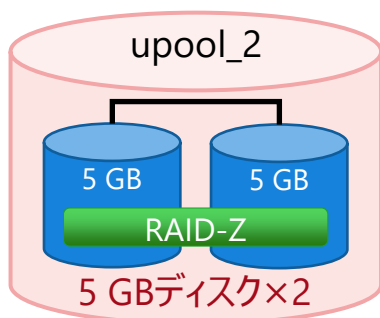
プロパティ名	意味
allocated	物理的に割り当てられているプール内のストレージ領域の量を識別する、読み取り専用の値です。
altroot	代替ルートディレクトリを識別します。
autoexpand	自動プール拡張を制御します。
autoreplace	自動デバイス交換を制御します。
bootfs	ルートプールのデフォルトのブート可能ファイルシステムを識別します。
cachefile	プール構成情報がキャッシュされる場所を制御します。
capacity	使用されるプール領域の割合（％）を識別する、読み取り専用の値です。
dedupditto	複製解除されたブロックの、参照カウントのしきい値を設定します。
dedupratio	プールに対して達成された、読み取り専用の複製解除率です。
delegation	特権のないユーザーに、ファイルシステムに対して定義されているアクセス権を許可できるかどうかを制御します。
failmode	壊滅的なプール障害が発生した場合の、システムの動作を制御します。
free	割り当てられていないプール内のブロック数を識別する、読み取り専用の値です。
guid	一意の識別子を識別する、読み取り専用プロパティです。
health	現在の健全性を識別する、読み取り専用プロパティです。
listshares	zfs listコマンドで、プール内の共有情報を表示するかどうかを制御します。
listsnapshots	zfs listコマンドで、プールに関連付けられているスナップショット情報を表示するかどうかを制御します。
readonly	プールを変更できるかどうかを指定します。
size	ストレージプールの合計サイズを識別する、読み取り専用プロパティです。
version	プールの現在のディスク上バージョンを識別します。

■ ストレージプールの実効容量を確認するには、zfs listコマンドを使用します。

✓ ストレージプールの実効容量の確認（zfs listコマンド）

【書式】 zfs list <ストレージプール名>

実行例：RAID-Z構成のストレージプール



片方のディスクが
パリティのため、
実効容量は約5 GB

```
# zfs list upool_2
NAME      USED  AVAIL  REFER  MOUNTPOINT
upool_2   75K   4.89G   31K    /upool_2
```

- 現在使用中の容量が「USED」に表示されます。
また、現在の空き容量が「AVAIL」に表示されます。
- 実効容量は、「USED」と「AVAIL」の合計値となります。

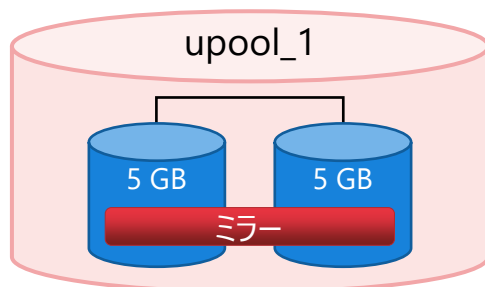


- RAID構成によって、zpool list コマンドでは実効容量を確認できないことがあります。
詳しくは、「《参考》ストレージプールの実効容量の確認方法 2/2」を参照してください。
- 実行容量の詳細については、Oracle社のマニュアルを参照してください。
『Oracle Solaris 11.4 でのZFS ファイルシステムの管理』（Oracle社）
https://docs.oracle.com/cd/E75431_01/html/E75197/index.html
「第11章 Oracle Solaris ZFS のトラブルシューティングとプールの回復」-「ZFS ストレージプール内のデータの問題を解決する」-「ZFS の領域の問題を解決する」

《参考》ストレージプールの実効容量の確認方法 2/2

zpool listコマンドで表示される容量は、RAID構成によって意味合いが異なります。ご注意ください。

ミラー構成の場合

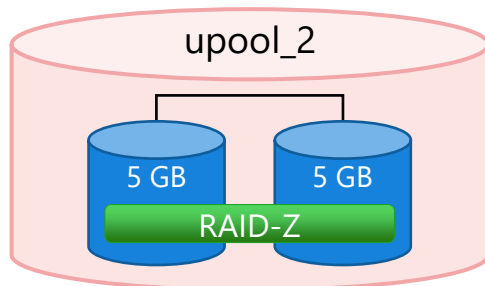


実効容量は5 GB

```
# zpool list upool_1
NAME      SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
upool_1   4.97G  98.5K   4.97G   0%   1.00x  ONLINE  -
```

実効容量を表示

RAID-Z構成の場合

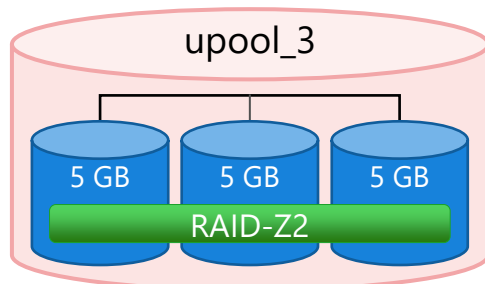


一方のディスクが
パリティのため、
実効容量は5 GB

```
# zpool list upool_2
NAME      SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
upool_2   9.94G  150K   9.94G   0%   1.00x  ONLINE  -
```

ディスクの合計容量を表示

RAID-Z2構成の場合



2つのディスクが
パリティのため、
実効容量は5 GB

```
# zpool list upool_3
NAME      SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
upool_3   14.9G  225K   14.9G   0%   1.00x  ONLINE  -
```

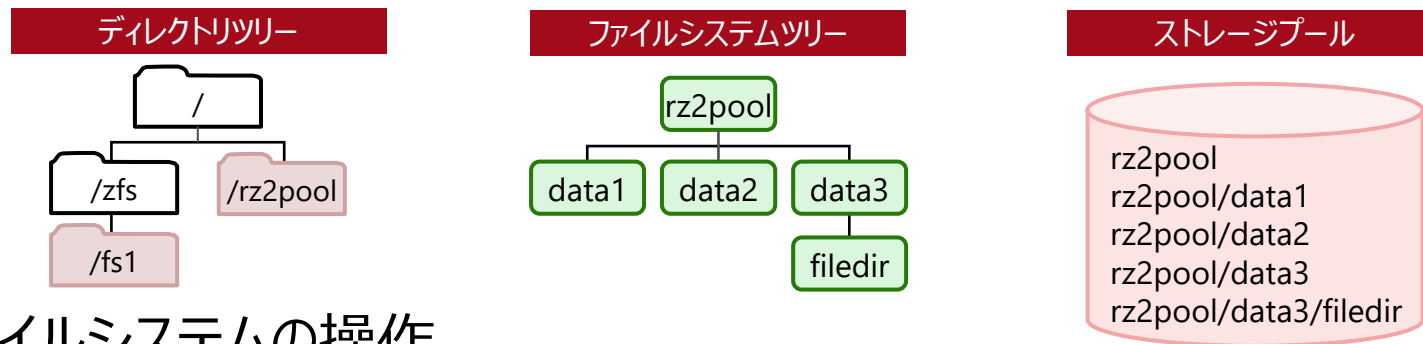
ディスクの合計容量を表示

2.ファイルシステムの構築

ファイルシステムの作成方法と各種設定方法を説明します。

■ 作成するファイルシステムの環境

本章では、ファイルシステムの作成方法やファイルシステムのプロパティの設定方法を説明します。
構築するファイルシステムの構成は、以下のとおりです。



■ ファイルシステムの操作

• [ファイルシステムの作成](#)

既存のストレージプール「rz2pool」にファイルシステムを作成します。

• [ファイルシステム名の変更](#)

• [マウント／アンマウント](#)

• [プロパティの設定](#)

ファイルシステムプロパティを設定し、以下の各種機能を実現します。

- マウントポイントの変更
- NFS共有設定
- ファイルシステムの割り当て制限と予約
- ユーザー／グループ割り当て制限
- データ重複の検知・排除、データ暗号化

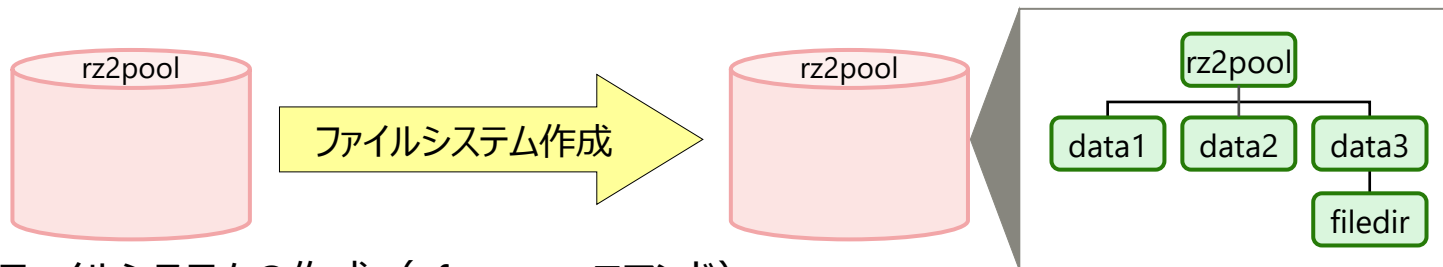
• [ファイルシステムの確認](#)

ファイルシステムの使用状況や、プロパティ情報を確認します。

• [ファイルシステムの削除](#)

■ ファイルシステム名を指定してファイルシステムを作成します。

- ファイルシステムを作成すると、「マウントポイントの作成」と「マウント」が自動的に実行されます。



✓ ファイルシステムの作成 (zfs createコマンド)

```
# zfs create rz2pool/data1
# zfs create rz2pool/data2
# zfs list -r rz2pool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
rz2pool	112K	29.3G	31.9K	/rz2pool
rz2pool/data1	30.9K	29.3G	30.9K	/rz2pool/data1
rz2pool/data2	30.9K	29.3G	30.9K	/rz2pool/data2

※ ファイルシステム名と同じマウントポイントが / (ルート) 直下に作成され、マウントされます。

✓ ファイルシステムの作成 (途中階層を同時に作成する場合)

```
# zfs create -p rz2pool/data3/filedir
# zfs list -r rz2pool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
rz2pool	227K	29.3G	32.9K	/rz2pool
rz2pool/data1	30.9K	29.3G	30.9K	/rz2pool/data1
rz2pool/data2	30.9K	29.3G	30.9K	/rz2pool/data2
rz2pool/data3	62.8K	29.3G	31.9K	/rz2pool/data3
rz2pool/data3/filedir	30.9K	29.3G	30.9K	/rz2pool/data3/filedir

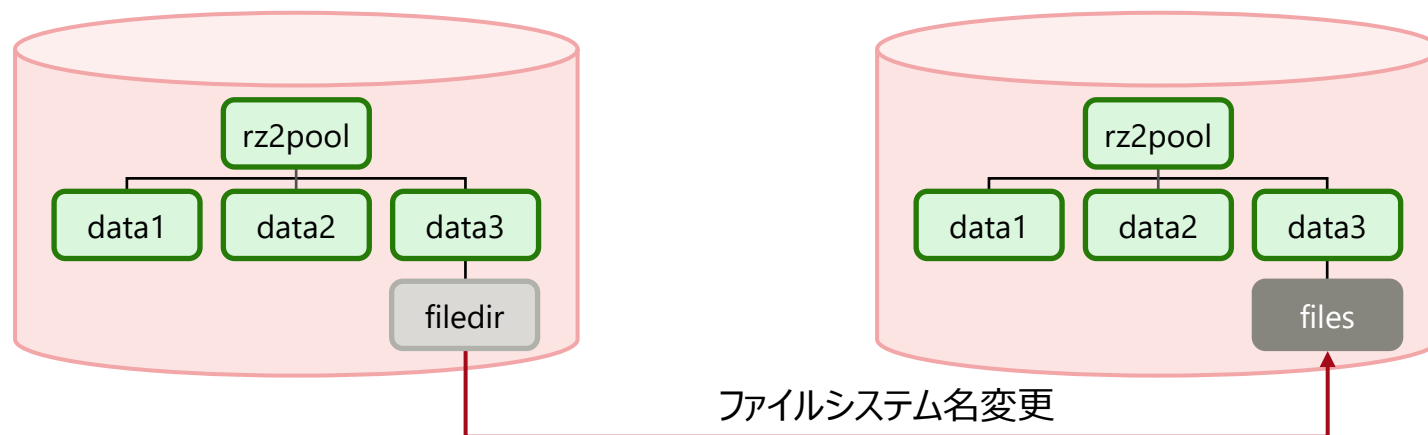


- ファイルシステムの作成と同時に、マウントポイントを指定することも可能です (mountpointプロパティ)。

```
# zfs create -o mountpoint=/zfs/fs4 rz2pool/data4
# zfs list -r rz2pool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
~ (省略) ~				
rz2pool/data4	30.9K	29.3G	30.9K	/zfs/fs4

- 作成したファイルシステムの名称を変更できます。



- ✓ ファイルシステム名の変更 (zfs renameコマンド)

```
# zfs rename rz2pool/data3/filedir rz2pool/data3/files
# zfs list -r rz2pool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
rz2pool	229K	29.3G	32.9K	/rz2pool
rz2pool/data1	30.9K	29.3G	30.9K	/rz2pool/data1
rz2pool/data2	30.9K	29.3G	30.9K	/rz2pool/data2
rz2pool/data3	62.8K	29.3G	31.9K	/rz2pool/data3
rz2pool/data3/files	30.9K	29.3G	30.9K	/rz2pool/data3/files



• ファイルシステム名を変更すると、同時にマウントポイントも変更されます。

- ZFSは、OS起動時、またはファイルシステム作成時に、自動マウントされます。
- 手動でマウント／アンマウントすることもできます。

- ✓ ファイルシステムのマウント（zfs mountコマンド）

```
# zfs mount rz2pool/data1
```

- ✓ ファイルシステムのアンマウント（zfs unmountコマンド）

```
# zfs unmount rz2pool/data1
```

-  • アンマウントは、ファイルシステムまたはマウントポイントを指定して行いますが、マウントはファイルシステムの指定のみで行います。
- UFSと同じ方法でマウントすることもできます。
詳しくは、「[《参考》レガシーマウントの設定](#)」を参照してください。

代表的なファイルシステムプロパティの設定方法を説明します。

✓ プロパティの設定（zfs setコマンドなど）

```
# zfs set <プロパティ>=value . . .
```

※ プロパティ名と値を指定します。値の指定方法は、プロパティによって異なります。

※ zfs createコマンド（ファイルシステム作成）など、zfs setコマンド以外で指定できるプロパティもあります。

■ 代表的なプロパティ

- マウントポイントの変更

mountpointプロパティ

- NFS共有設定

share.nfsプロパティ

- ファイルシステムの割り当て制限と予約

quotaプロパティ、reservationプロパティ

- ユーザー／グループ割り当て制限

defaultuserquotaプロパティ、defaultgroupquotaプロパティ、userquota@userプロパティ、groupquota@groupプロパティ

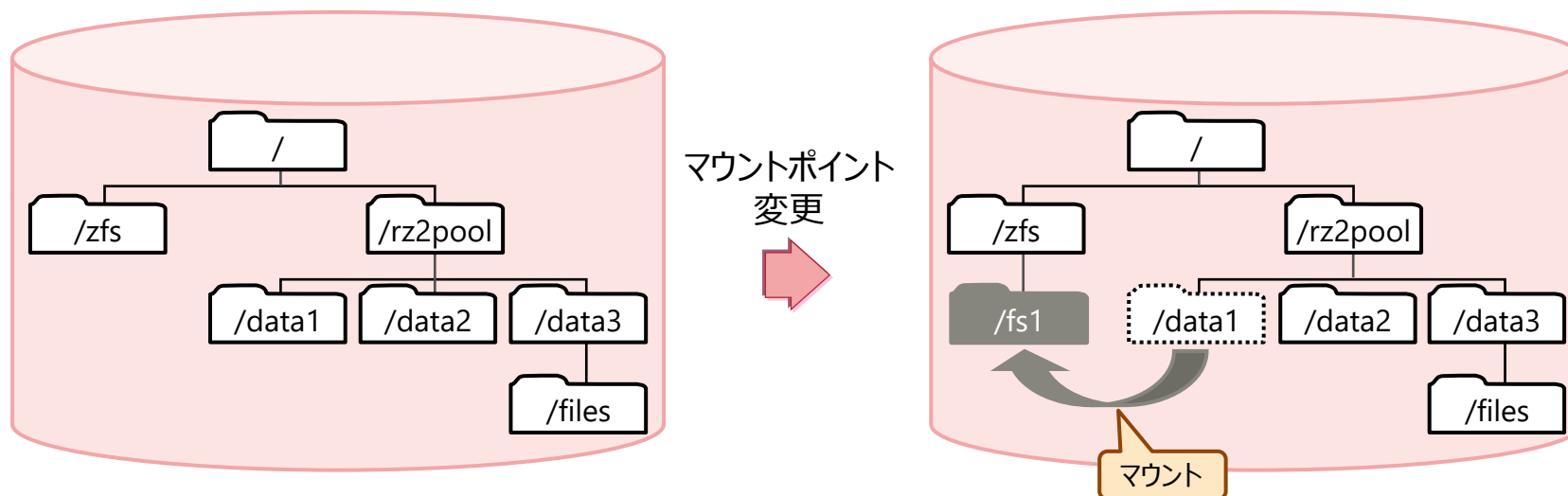
- データ重複の検知・排除

dedupプロパティ

- データ暗号化

encryptionプロパティ

■ マウントポイントの変更



✓ マウントポイントの変更 (mountpointプロパティ)

```
# zfs set mountpoint=/zfs/fs1 rz2pool/data1
```



- デフォルトでは、ファイルシステム名と同じマウントポイントが/（ルート）直下に作成されます。
- マウントポイントを変更しても、ファイルシステム名は変更されません。

■ NFS共有設定

Solaris 11.1以降の場合

- sharenfsプロパティは、share.nfsプロパティに変更されます。

✓ 共有の設定（share.nfsプロパティ）

【書式】 zfs set share.nfs=[on | off] <データセット>

```
# zfs set share.nfs=on rz2pool/data2
```

Solaris 11 11/11の場合

- 2つの手順で共有設定を行います。1つ目の手順で共有を作成（zfs set share）し、2つ目の手順でNFS共有を公開（sharenfs=on）します。

※ Solaris 11.1以降でも、この設定方法を使用できます。

① 共有の作成（zfs set shareコマンド）

【書式】 zfs set share=name=<共有名>,path=<共有パス>,prot=<ファイル共有プロトコル[smb | nfs]> <データセット>

```
# zfs set share=name=d2,path=/rz2pool/data2,prot=nfs rz2pool/data2
name=d2,path=/rz2pool/data2,prot=nfs
```

② 共有の公開（sharenfsプロパティ）

【書式】 zfs set sharenfs=[on | off] <データセット>

```
# zfs set sharenfs=on rz2pool/data2
```



• データセットは、ストレージプール内に作成されるファイルシステム、ボリューム、スナップショット、クローンの総称です。

■ ファイルシステムの割り当て制限と予約

- ✓ ファイルシステムの使用可能領域の確保（reservationプロパティ）

事前に領域を確保できます。

```
# zfs set reservation=500M rz2pool/data3
# zfs get reservation rz2pool/data3
NAME                PROPERTY    VALUE    SOURCE
rz2pool/data3      reservation 500M     local
```

- ✓ 使用可能領域の上限設定（quotaプロパティ）

quota（上限設定）を設定できます。

```
# zfs set quota=600M rz2pool/data3
# zfs get quota rz2pool/data3
NAME                PROPERTY    VALUE    SOURCE
rz2pool/data3      quota       600M     local
```


■ ユーザー／グループ割り当て制限

- ✓ ユーザー割り当て制限のデフォルト値設定（defaultuserquotaプロパティ）

1ユーザーごとの使用容量の上限を設定できます。

```
# zfs set defaultuserquota=25G rz2pool/data3
```

- ✓ グループ割り当て制限のデフォルト値設定（defaultgroupquotaプロパティ）

1グループごとの使用容量の上限を設定できます。

```
# zfs set defaultgroupquota=50G rz2pool/data3
```

- ✓ 特定ユーザーへの割り当て制限（userquota@userプロパティ）

特定のユーザーに対する使用容量の上限を設定できます。

```
# zfs set userquota@user01=300M rz2pool/data3
```

ユーザーが複数のグループに所属しており、ユーザーとそのプライマリグループ、セカンダリグループにそれぞれ割り当て制限がされている場合、各プロパティ中の最低値が優先されます。

例：以下のような値で割り当て制限がされている場合は、ユーザーへの割り当て制限が最小値のため、「300M」が優先されます。

- ・ ユーザー : 300M ←最小値の300Mが優先されます。
- ・ プライマリグループ : 500M
- ・ セカンダリグループ : 700M

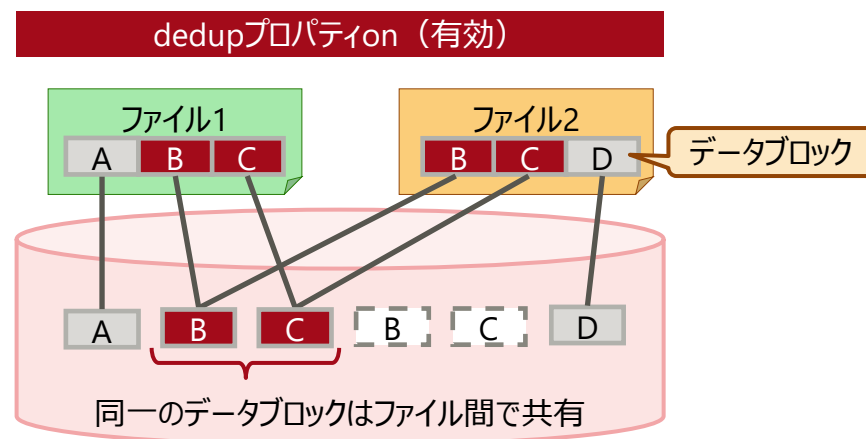
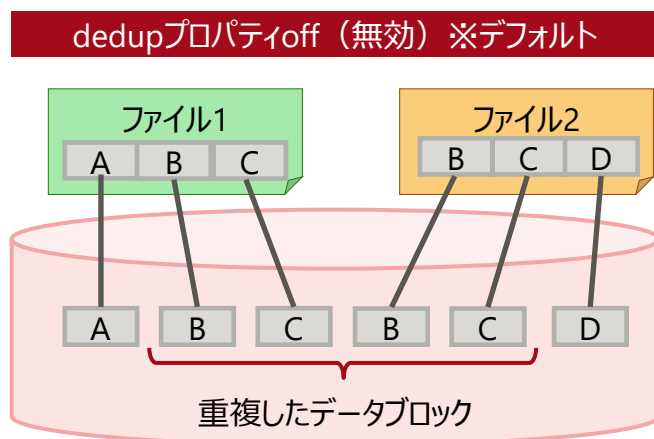
- ✓ 特定グループへの割り当て制限（groupquota@groupプロパティ）

特定のグループに対する使用容量の上限を設定できます。

```
# zfs set groupquota@group01=500M rz2pool/data3
```

■ データ重複の検知・排除

- データセットごとに重複除外機能を設定できます。
- 複数のファイルにおける重複したデータ領域（データブロック）を検知・排除することで、ストレージプールの容量を節約します。



✓ 重複除外機能の設定（dedupプロパティ）

【書式】 zfs set dedup=[on | off] <対象とするファイルシステム名>

```
# zfs set dedup=on rz2pool/data1
```



・仮想化やバックアップ時に、ディスク容量の有効活用が可能です。

■ dedupプロパティを有効 (on) にする場合の留意事項

- zdbコマンドを使用し、重複除外率 (dedup) と必要メモリを計算します。
重複除外率が「2」以上かつ、必要メモリに対して搭載メモリに余裕があれば容量節約効果を発揮でき、性能への影響も最小に抑えることができます。

✓ 必要メモリの計算方法

必要メモリ = 割り当てられているブロック数 × 320 (※1)

※1: 「320」は係数です。dedupテーブルのサイズ (バイト) です。

zdb -S rpool ~ 重複除外対象のストレージプール

Simulated DDT histogram:

bucket					allocated				referenced			
refcnt	blocks	LSIZE	PSIZE	DSIZE	blocks	LSIZE	PSIZE	DSIZE	blocks	LSIZE	PSIZE	DSIZE
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
1	733K	58.7G	57.1G	57.1G	733K	58.7G	57.1G	57.1G	733K	58.7G	57.1G	57.1G
2	34.8K	865M	832M	832M	74.3K	1.82G	1.74G	1.74G	74.3K	1.82G	1.74G	1.74G
~ (省略) ~												
Total	772K	59.6G	58.0G	58.0G	3.28M	376G	375G	375G	3.28M	376G	375G	375G

dedup = 6.46, compress = 1.00, copies = 1.00, dedup * compress / copies = 6.48

左記の例の場合

- 重複除外率 (dedup) ... 6.46
- 必要メモリ... 247.04 M
(772 K × 320 = 247.04 M)



重複除外率が2以上で、搭載メモリに余裕がある。

搭載メモリ - 必要メモリ = 16G - 247.04M
= 15.75G

※小数点第二位まで四捨五入

※本書の搭載メモリは16G



dedupプロパティを有効にすると効果がある。

※ 重複除外は、同一ストレージプール内のdedup=onに設定したデータセット内、またはデータセット間で行われます。ファイルシステムが異なってもストレージプールが同じであれば、重複除外されます。

※ dedup=onを設定したあとに書き込まれたデータに対して有効になります。設定前に書き込まれたデータに関しては、重複除外されません。

※ zfs listコマンドで表示される容量は、重複除外分は考慮されません。参照容量が表示されるため、実際の使用量よりも多く表示されます。



- dedupプロパティは、Oracle社のマニュアルを参照し、事前に考慮事項を確認してから有効にしてください。

『Oracle Solaris 11.4 での ZFS ファイルシステムの管理』(Oracle社)

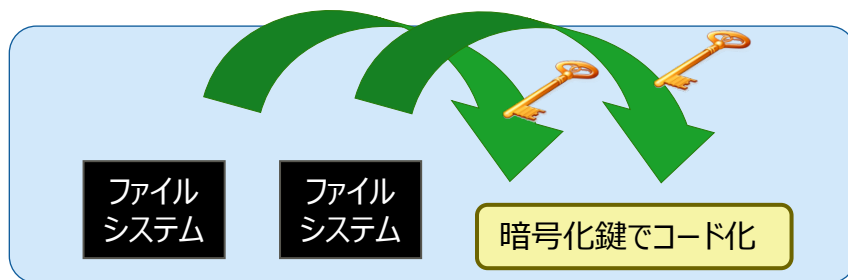
https://docs.oracle.com/cd/E75431_01/html/E75197/index.html

「第 7 章 Oracle Solaris ZFS ファイルシステムの管理」-「ZFS のプロパティの概要」-「設定可能な ZFS ネイティブプロパティ」-「dedup プロパティ」

■ データ暗号化

- 暗号化ポリシー（※1）は、zfs createコマンドによるファイルシステムの作成時に、データセット単位で設定できます。
- 暗号化ポリシー（※1）は変更できません。

※1：暗号化ポリシーは、暗号化する（on）／しない（off）の設定のことです。



✓ ファイルシステムの暗号化（encryptionプロパティ）

```
# zfs create -o encryption=on rz2pool/data5
Enter passphrase for 'rz2pool/data5':*****
Enter again:*****
```

最低 8 文字の長さが必要なパスフレーズの入力が必要です（デフォルトの暗号化ポリシー）。

✓ ファイルシステムの暗号化が有効になっていることを確認

```
# zfs get encryption rz2pool/data5
NAME          PROPERTY  VALUE  SOURCE
rz2pool/data5 encryption on      local
```

■ 読み取り専用設定

- ✓ ファイルシステムの読み取り専用設定（readonlyプロパティ）

ファイルシステムを読み取り専用にすることができます。

```
# zfs set readonly=on rz2pool/data1
```

■ データ圧縮

- ✓ ファイルシステムのデータ圧縮（compressionプロパティの設定）

ファイルシステムのデータを自動で圧縮します。

次の圧縮形式を指定できます。

[on | lzjb | gzip | gzip-[0-9]]

```
# zfs set compression=on rz2pool/data1
```

※ onとlzjbは、どちらを指定しても同じです。

※ gzipとgzip-6は、どちらを指定しても同じです。

■ ファイルシステムの使用状況とマウントポイントを確認できます。

- ✓ すべてのファイルシステム情報を確認（zfs listコマンド）

```
# zfs list
NAME                                USED    AVAIL    REFER    MOUNTPOINT
rpool                               14.6G   34.4G    73.5K    /rpool
~ (省略) ~
rz2pool                             500M    28.8G    34.9K    /rz2pool
rz2pool/data1                      30.9K    28.8G    30.9K    /zfs/fs1
rz2pool/data2                      30.9K    29.3G    30.9K    /rz2pool/data2
rz2pool/data3                      65.7K    600M    34.9K    /rz2pool/data3
~ (省略) ~
```

①	②	③	④	⑤
NAME	USED	AVAIL	REFER	MOUNTPOINT
rpool	14.6G	34.4G	73.5K	/rpool
~ (省略) ~				
rz2pool	500M	28.8G	34.9K	/rz2pool
rz2pool/data1	30.9K	28.8G	30.9K	/zfs/fs1
rz2pool/data2	30.9K	29.3G	30.9K	/rz2pool/data2
rz2pool/data3	65.7K	600M	34.9K	/rz2pool/data3
~ (省略) ~				

出力形式

- ① : データセット名
- ② : 使用済み領域サイズ
- ③ : 使用可能領域サイズ
- ④ : データセットからアクセス可能なデータ量
- ⑤ : マウントポイント

- ✓ 指定したファイルシステムの情報を確認

```
# zfs list rz2pool
NAME                                USED    AVAIL    REFER    MOUNTPOINT
rz2pool                             500M    28.8G    34.9K    /rz2pool
```

- ✓ 指定したファイルシステムと、配下の情報を確認

```
# zfs list -r rz2pool
NAME                                USED    AVAIL    REFER    MOUNTPOINT
rz2pool                             500M    28.8G    34.9K    /rz2pool
rz2pool/data1                      30.9K    28.8G    30.9K    /zfs/fs1
rz2pool/data2                      30.9K    29.3G    30.9K    /rz2pool/data2
rz2pool/data3                      65.7K    600M    34.9K    /rz2pool/data3
rz2pool/data3/files                30.9K    600M    30.9K    /rz2pool/data3/files
~ (省略) ~
```

■ データセット単位で、ファイルシステムのプロパティ情報を確認できます。

✓ ファイルシステムのプロパティ情報の確認 (zfs getコマンド)

```
# zfs get all rz2pool
```

NAME	PROPERTY	VALUE	SOURCE
rz2pool	aclinherit	restricted	default
rz2pool	aclmode	discard	default
rz2pool	atime	on	default
～ (省略)	～		
①	②	③	④

✓ 指定したプロパティ情報を個別に確認

```
# zfs get type rz2pool
```

NAME	PROPERTY	VALUE	SOURCE
rz2pool	type	filesystem	-
①	②	③	④

指定したプロパティのみ
表示されます。

出力形式

① : データセット名

② : プロパティ名

③ : プロパティの値

④ : プロパティの状態

default	: 明示的に設定されていない (デフォルト値)
local	: 明示的に設定した値
inherited from データセット名	: 表示されているデータセット名からの継承
temporary	: マウントの有効期間のみ有効
- (なし)	: 読み取り専用

■ zfs destroyコマンドでファイルシステムを削除します。

✓ ファイルシステムの削除（zfs destroyコマンド）

```
# zfs destroy rz2pool/data1
# zfs list -r rz2pool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
rz2pool	500M	28.8G	34.9K	/rz2pool
rz2pool/data2	30.9K	29.3G	30.9K	/rz2pool/data2
rz2pool/data3	65.7K	600M	34.9K	/rz2pool/data3
rz2pool/data3/files	30.9K	600M	30.9K	/rz2pool/data3/files

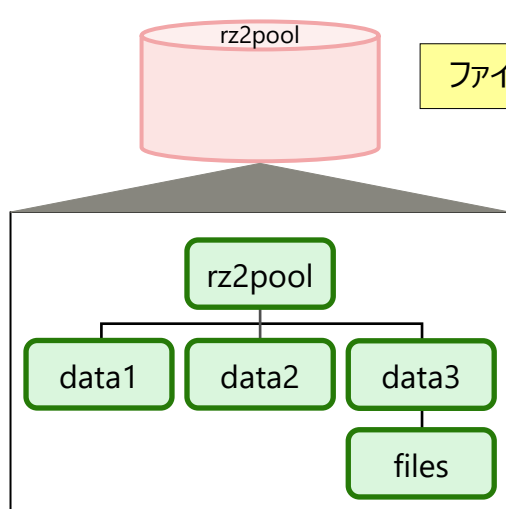
～（省略）～

✓ 配下のファイルシステムも再帰的に削除

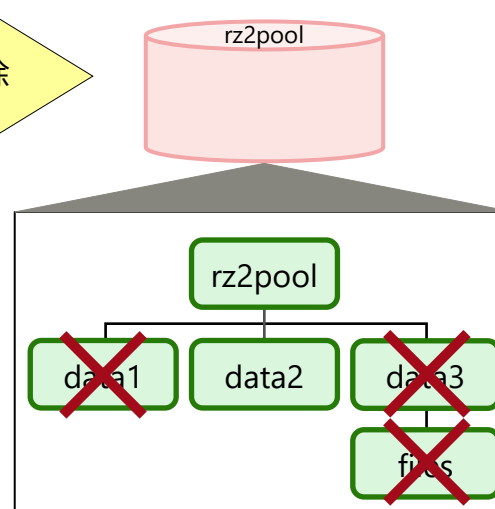
```
# zfs destroy -r rz2pool/data3
# zfs list -r rz2pool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
rz2pool	220K	29.3G	32.9K	/rz2pool
rz2pool/data2	30.9K	29.3G	30.9K	/rz2pool/data2

data3を指定して削除した場合、
配下のfilesも削除されます。



ファイルシステム削除



-rオプションにより、
配下のファイルシステムも
同時に削除できます。

※ 指定したファイルシステムがアンマウントされ、削除されます。

《参考》ファイルシステムのプロパティ 1/4

プロパティ名	意味	備考
available	データセット、およびそのデータセットのすべての子孫が使用可能な容量。	
compressratio	データセットに適用される圧縮率。乗数で表記される。	
creation	データセットが作成された日付と時間。	
defer_destroy	スナップショットの保持情報。zfs destroy -dコマンドを使用し、スナップショットが遅延破棄してマークされている場合はオンに設定される。それ以外の場合は、オフに設定される。	
keychangedate	zfs key -c操作により、ラッピング鍵を最後に変更した日付。鍵の変更操作が実行されていない場合は、値なし。	
keystatus	データセットの暗号化鍵のステータス。	Solaris 11 11/11以降
mounted	ファイルシステムが現在マウントされているかどうかを示す。	
origin	複製されたファイルシステム、またはボリュームの作成元のスナップショット。	
referenced	データセットからアクセス可能なデータ量。	
rekeydate	zfs key -Kまたはzfs clone -K操作により、データ暗号化鍵を最後に変更した日付。	Solaris 11 11/11以降
type	データセットのタイプ（filesystem、volume、snapshot）。	
used	データセット、およびそのデータセットのすべての子孫が消費する容量。	
usedbychildren	データセットの子によって使用される容量。	Solaris 10 10/09以降
usedbydataset	データセット自体によって使用される容量。	Solaris 10 10/09以降
usedbyreservation	データセットに設定されている reservation（データセットに保証される最小容量）によって使用される容量。	Solaris 10 10/09以降
usedbysnapshots	データセットのスナップショットによって消費される容量。	Solaris 10 10/09以降
userused@user	指定されたユーザーによって消費されるデータセットの容量。	Solaris 10 10/09以降
userrefs	スナップショットでユーザーを保持する数。ユーザーの保持は、zfs holdコマンドを使用して設定する。	
groupused@group	指定されたグループによって消費されるデータセットの容量。	Solaris 10 10/09以降
volblocksize	ボリュームに対してボリュームのブロックサイズを指定する。	

《参考》ファイルシステムのプロパティ 2/4

プロパティ名	意味	備考
effectivereadlimit	データセットがディスクから読み取り可能な、1 秒間の最大バイト数。	Solaris 11.4以降
effectivewritelimit	データセットがディスクに書き込み可能な、1 秒間の最大バイト数。	Solaris 11.4以降
aclmode	Chmod操作中にACLエントリがどのように変更されるかを制御する。	
aclinherit	ファイルとディレクトリが作成されるときに、ACLエントリをどのように継承するかを制御する。	
atime	ファイルを読み取るときに、ファイルのアクセス時間を更新するかどうかを制御する。	
canmount	ファイルシステムがzfs mount コマンドを使ってマウントできるかどうかを制御する。	
checksum	データの完全性を検証するために使用するチェックサムを制御する。	
compression	データセットで 사용되는圧縮アルゴリズムを制御する。	
copies	データセットに関して、保存されるデータのコピー数を制御する。	
dedup	データセットに対して、複製解除を適用するかどうかを制御する。	Solaris 11 11/11以降
devices	ファイルシステムのデバイスファイルを開くことができるかどうかを制御する。	
exec	ファイルシステム内のプログラムの実行を許可するかどうかを制御する。	
logbias	ZFSがデータセットに対する同期要求を最適化する方法を制御する。	
mlslabel	データセットをゾーンにマウントできるかどうかを指定する機密ラベル。	Trusted Extensions有効時
mountpoint	ファイルシステムで使用されるマウントポイントを制御する。	
nbmand	ファイルシステムをnbmand（非ブロッキング強制ロック）でマウントするべきかどうかを制御する。	
primarycache	プライマリキャッシュ（ARC）にキャッシュされる内容を制御する。	Solaris 10 10/09以降
quota	データセット、およびその子孫が消費できる容量を制限する。	
readlimit	データセットの読み取り速度（バイト/秒）を制限する。	Solaris 11.4以降
writelimit	データセットの書き込み速度（バイト/秒）を制限する。	Solaris 11.4以降
defaultreadlimit	データセットに対する読み取りのデフォルトの制限（バイト/秒）を設定する。	Solaris 11.4以降
defaultwritelimit	データセットに対する書き込みのデフォルトの制限（バイト/秒）を設定する。	Solaris 11.4以降

《参考》ファイルシステムのプロパティ 3/4

プロパティ名	意味	備考
sync	ファイルシステムトランザクションを同期する程度を指定する。	
defaultuserquota	デフォルトのユーザー割り当て制限を設定する。	Solaris 11.3以降
defaultgroupquota	デフォルトのグループ割り当て制限を設定する。	Solaris 11.3以降
userquota@user	指定されたユーザーによって消費される容量を制限する。	Solaris 10 10/09以降
groupquota@group	指定されたグループによって消費される容量を制限する。	Solaris 10 10/09以降
readonly	データセットを変更できるかどうかを制御する。	Solaris 11 11/11以降
recordsize	ファイルシステムに格納するファイルの推奨ブロックサイズを指定する。	
refquota	1つのデータセットが消費できる容量を制限する。	
refreservation	データセットに保証される最小容量。	
reservation	データセット、およびその子孫に保証される最小容量。	
rstchown	ファイルシステムの所有者が、ファイルの所有権の変更を許可できるかどうかを示す。	Solaris 11 11/11以降
secondarycache	セカンダリキャッシュ（L2ARC）にキャッシュされる内容を制御する。	Solaris 10 10/09以降
setuid	setuidビットをファイルシステムで考慮するかどうかを制御する。	
shadow	ZFSファイルシステムを、URIで記述されたファイルシステムのshadowとして識別する。	
share.nfs (別名sharenfs)	ZFSデータセットをNFS共有として作成、および公開するかどうかを制御する。	Solaris 11.1以降
share.smb (別名sharesmb)	ZFSデータセットをSMB共有として作成、および公開するかどうかを制御する。	Solaris 11.1以降
snapdir	ファイルシステムのルートで、.zfsディレクトリを非表示にするか、表示するかを制御する。	
version	ファイルシステムのディスク上バージョン。	
volsize	ボリュームの論理サイズを指定する。	
vscan	ファイルを開くときや閉じるときに、ファイルのウイルススキャンを実行するかどうかを制御する。	

《参考》ファイルシステムのプロパティ 4/4

プロパティ名	意味	備考
xattr	ファイルシステムで拡張属性が有効かどうかを制御する。	
zoned	データセットを非大域ゾーンから管理するかどうかを制御する。	
casesensitivity	ファイルシステムで使用するファイル名照合アルゴリズムで、大文字と小文字を区別するか、区別しないか、または両方の照合方式の組み合わせを許可するかを指定する。	
normalization	ファイルシステムで2つのファイル名を比較するとき、常にファイル名のunicode正規化を実行するかどうか、およびどの正規化アルゴリズムを使用するかを指定する。	
utf8only	UTF-8文字コードセットに存在しない文字が含まれているファイル名を、ファイルシステムで拒否するかどうかを設定する。	
encryption	暗号化されたデータセットに使用する暗号化アルゴリズムと鍵長を定義する。	Solaris 11 11/11以降
multilevel	ZFS ファイルシステム内のオブジェクトで、ラベルの使用を有効化するために使用する。	Trusted Extensions有効時
keysource	データセットの鍵をラップする鍵の形式と場所を定義する。	Solaris 11 11/11以降

■ UFSと同じ方法でマウントすることもできます。

- レガシーマウントとは、従来のファイルシステム（UFS）と同様に、/etc/vfstabやmountコマンドで管理することです。
- mountpointプロパティに「legacy」を指定することで、設定します。
- レガシーマウントは、Solaris ゾーンで共有ファイルシステムを指定する際になどに設定します。

※ レガシーマウントを設定した場合、手動マウント／手動アンマウントはzfs mount／zfs unmountコマンドではなく、mount／umountコマンドを使用します。

✓ レガシーマウントの設定（mountpointプロパティ）

```
# zfs set mountpoint=legacy rz2pool/data2
```

✓ マウント（mountコマンド）

```
# mount -F zfs rz2pool/data2 /mnt
```

ファイルシステムタイプには「zfs」を指定します。

✓ アンマウント（umountコマンド）

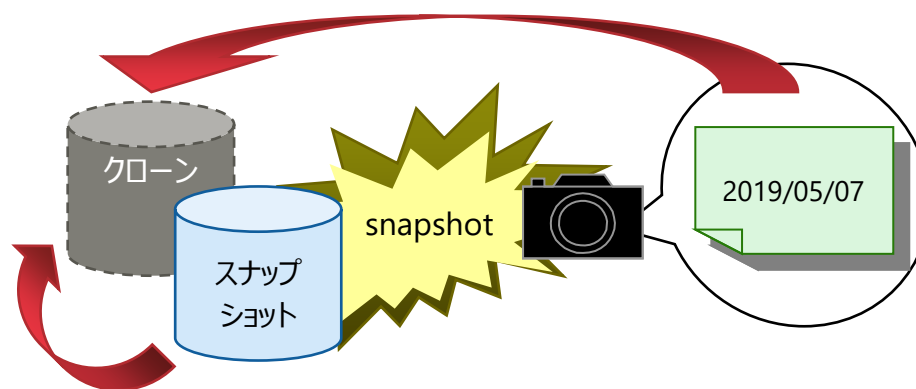
```
# umount /mnt
```

3.スナップショットとクローンの作成

スナップショットとクローンの作成・操作方法を説明します。

■ スナップショットとクローン作成

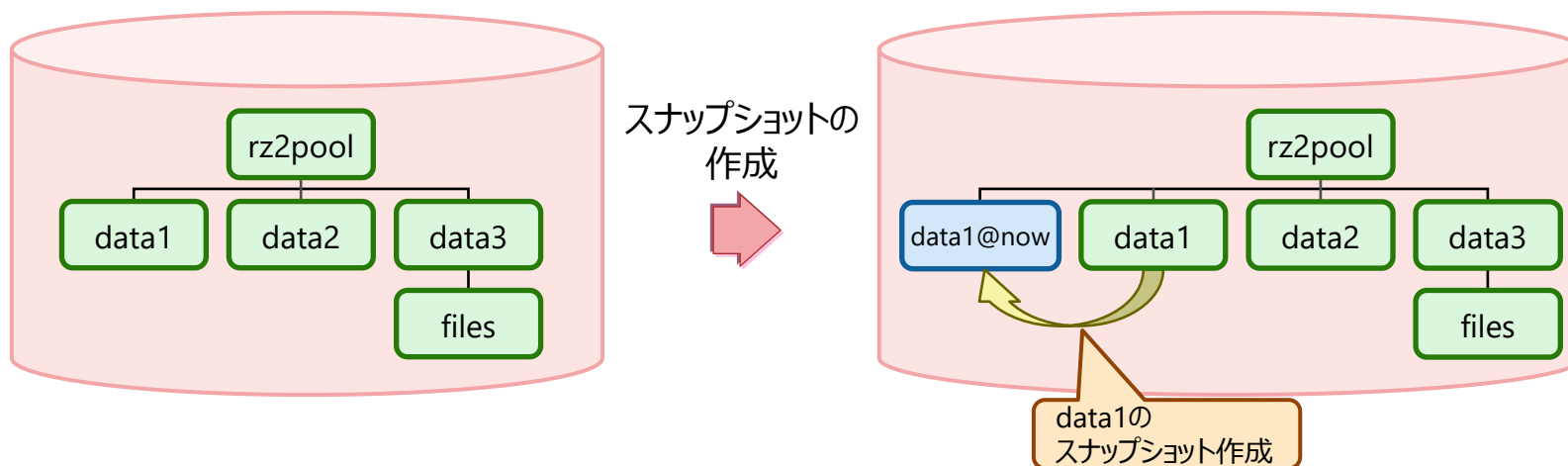
本章では、スナップショットとクローンの作成・操作方法を説明します。



■ スナップショットの操作、クローンの操作

- [スナップショットの作成](#)
- [スナップショット名の変更](#)
- [スナップショットの差分表示](#)
- [クローンの作成](#)
- [ファイルシステムの置き換え](#)
- [スナップショットからのロールバック](#)
- [自動スナップショットの設定](#)

- データセット（主にファイルシステム）を指定し、スナップショットを作成します。



✓ スナップショットの作成（zfs snapshotコマンド）

【書式】 zfs snapshot <データセット名@スナップショット名>

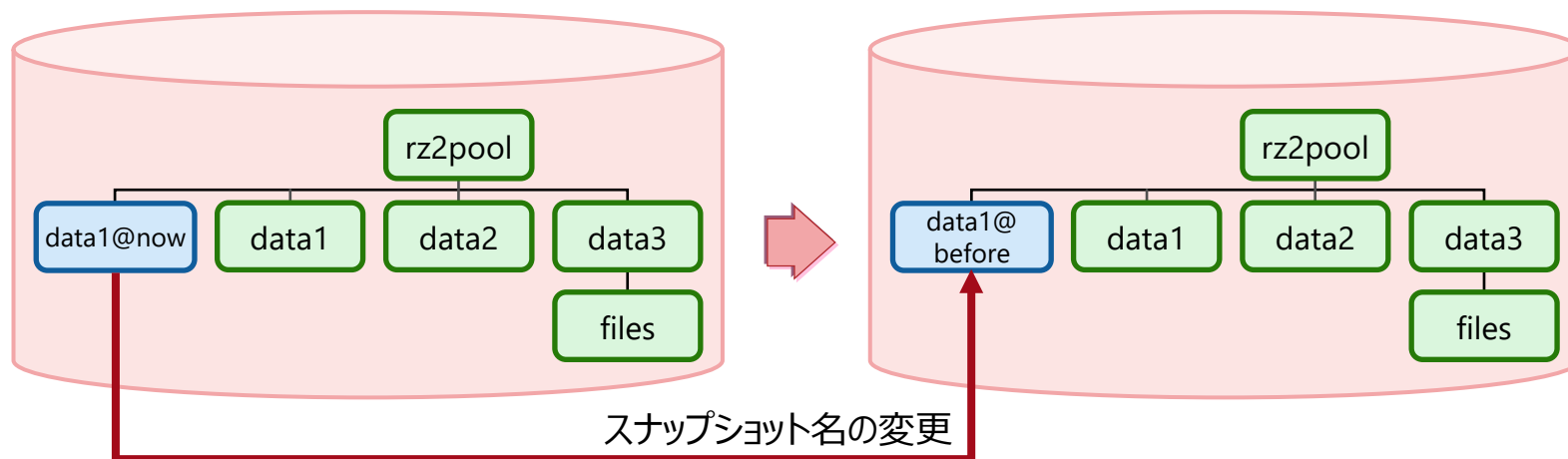
```
# zfs snapshot rz2pool/data1@now
```



- スナップショットは、指定したデータセットと同じストレージプール内に作成されます。
- -r オプションを指定することで、指定したデータセット配下の、すべてのデータセットのスナップショットを同時に作成できます。

```
# zfs snapshot -r rz2pool/data3@now
```


- スナップショット作成後にスナップショット名を変更できます。



✓ スナップショット名の変更 (zfs renameコマンド)

【書式】 zfs rename <データセット名@変更前のスナップショット名> <データセット名@変更後のスナップショット名>

```
# zfs rename rz2pool/data1@now rz2pool/data1@before
```

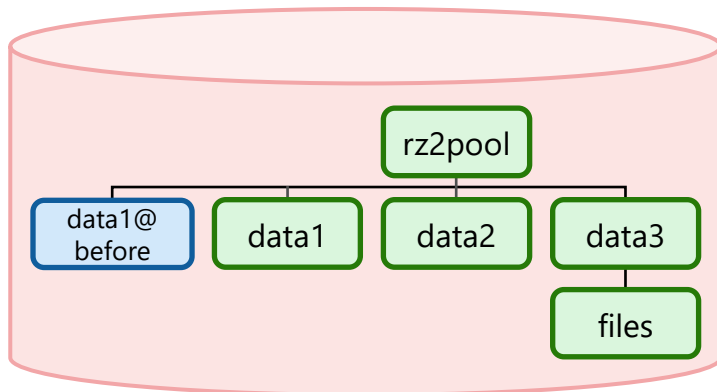
- 💡 スナップショットは、「データセット名@スナップショット名」の形で作成されます。そのため、スナップショット名を変更すると、@より後ろの「スナップショット名」の部分が変更されます。

スナップショットの差分表示

■ 2つのスナップショットの差分を確認できます。

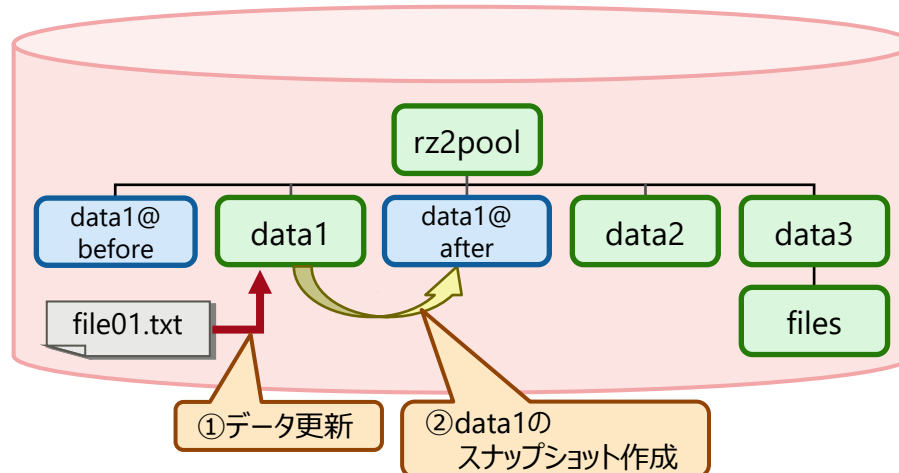
data1を更新する前のスナップショット

rz2pool/data1@before



data1を更新したあとのスナップショット

rz2pool/data1@after



✓ データ更新前とデータ更新後のスナップショットの差分確認 (zfs diffコマンド)

【書式】 zfs diff <スナップショット名1> <スナップショット名2>

```
# zfs diff rz2pool/data1@before rz2pool/data1@after
M      /rz2pool/data1/
+      /rz2pool/data1/file01.txt
```

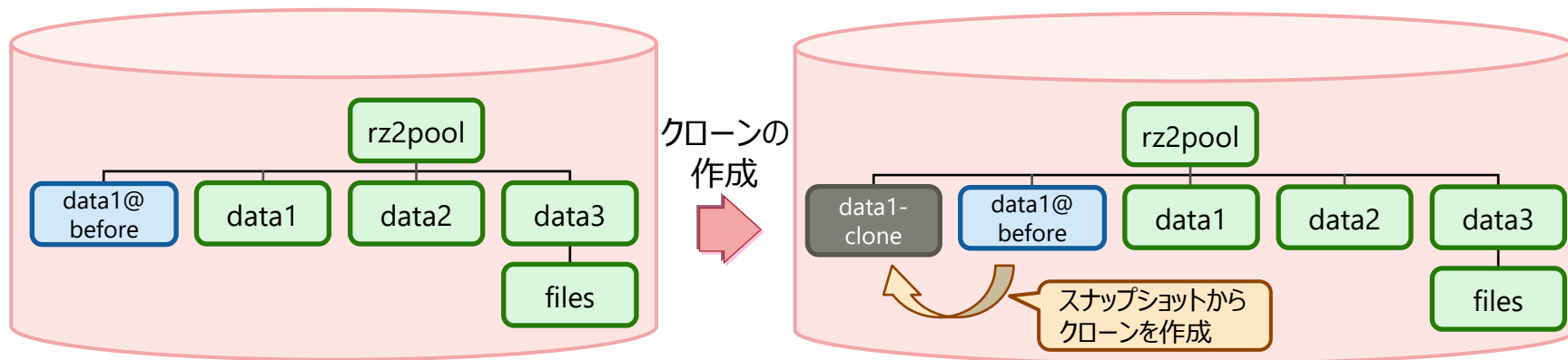
スナップショットの差分表記

- M : ファイルまたはディレクトリの変更を示します。
- R : ファイルまたはディレクトリの名前が変更されたことを示します。
- : 古いスナップショットには存在し、新しいスナップショットには存在しないことを示します。
- + : 新しいスナップショットには存在し、古いスナップショットには存在しないことを示します。



- zfs diffコマンド実行時にスナップショットを1つだけ指定した場合、ファイルシステムの現在の状態との差分が表示されます。
- -rオプションで、指定したスナップショットとすべての子スナップショットの差分が表示されます。

- スナップショットからデータセットのクローンを作成します。



✓ クローンの作成 (zfs cloneコマンド)

【書式】 zfs clone <データセット名@スナップショット名> <クローンのファイルシステム名>

```
# zfs clone rz2pool/data1@before rz2pool/data1-clone
```

✓ 暗号化されたファイルシステムのクローンの作成

```
# zfs snapshot rz2pool/data5
@encryption
# zfs clone rz2pool/data5@encryption rz2pool/data5-clone
Enter passphrase for 'rz2pool/data5-clone':*****
Enter again:*****
```

※ 暗号化されたファイルシステムのクローンを作成する際に、ポリシーが適用され、パスワードが求められます。
ファイルシステムの暗号化については、「[プロパティの設定 8/9 -データ暗号化-](#)」を参照してください。



• クローンは、指定した同一ストレージプール内の別のファイルシステム配下にも作成できます。例を以下に示します。

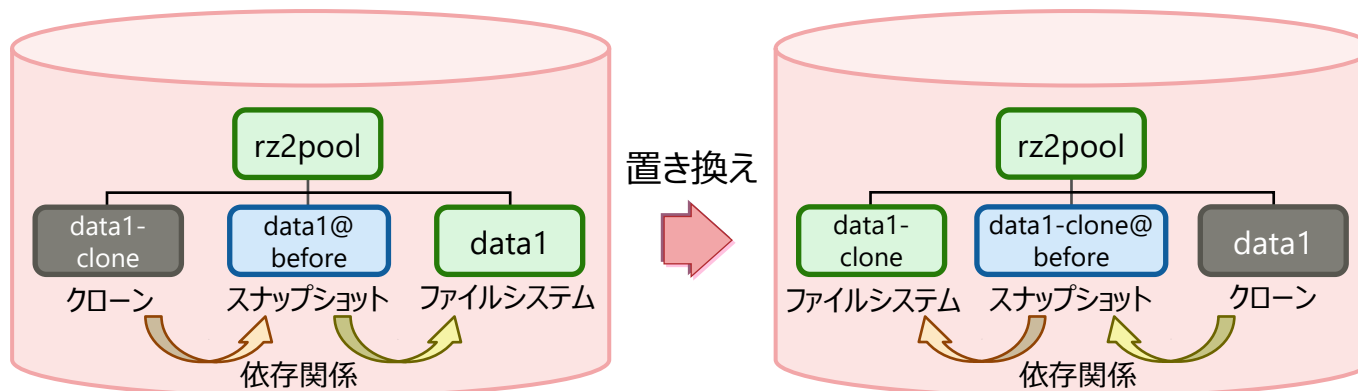
```
# zfs clone rz2pool/data1@before rz2pool/data3/data1-clone
```

- 元ファイルシステム（マスター）とクローンの依存関係を入れ替え、クローンをマスターファイルシステムにします。

- ✓ 元ファイルシステムとクローンの置き換え（zfs promoteコマンド）

【書式】 zfs promote <クローンのファイルシステム名>

```
# zfs promote rz2pool/data1-clone
```



```
# zfs get -r origin rz2pool
```

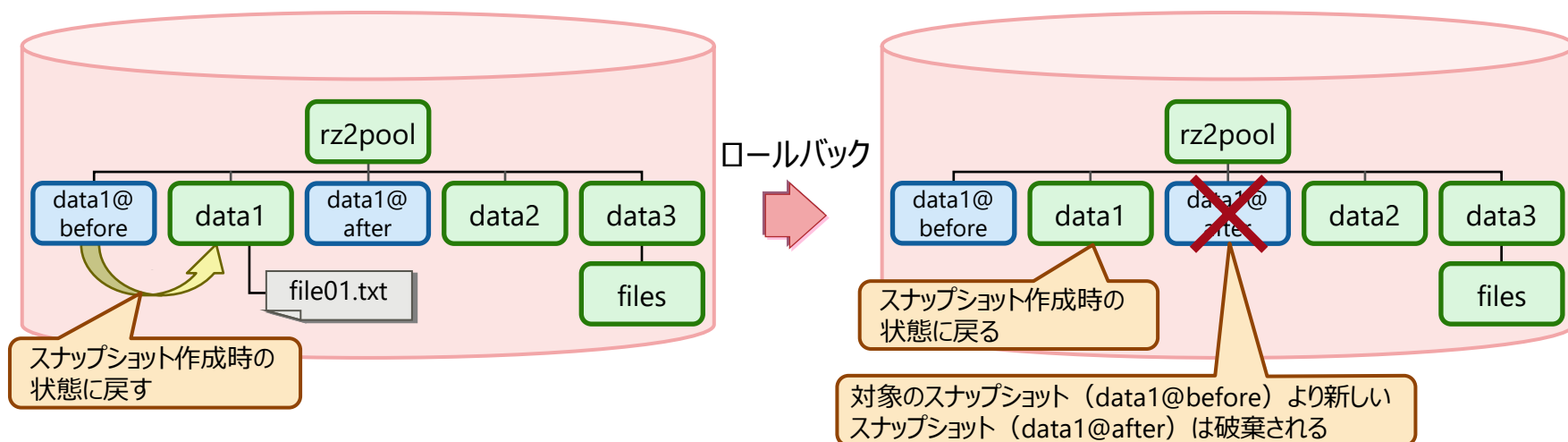
NAME	PROPERTY	VALUE	SOURCE
rz2pool	origin	-	-
rz2pool/data1	origin	-	-
rz2pool/data1@before	origin	-	-
rz2pool/data1-clone	origin	rz2pool/data1@before	-

```
# zfs get -r origin rz2pool
```

NAME	PROPERTY	VALUE	SOURCE
rz2pool	origin	-	-
rz2pool/data1	origin	rz2pool/data1-clone@before	-
rz2pool/data1-clone	origin	-	-
rz2pool/data1-clone@before	origin	-	-

- ZFSでは、クローンとファイルシステムの依存関係を置き換えることができます。置き換え後には、スナップショットのデータセット名も変更されます。
- クローンとファイルシステムでは異なるプロパティが設定されているため、置き換え後にファイルシステムのプロパティを環境に合わせて設定する必要があります。

- スナップショットからロールバックすることで、ファイルシステムをスナップショット作成時の状態に戻します。



✓ ロールバック (zfs rollbackコマンド)

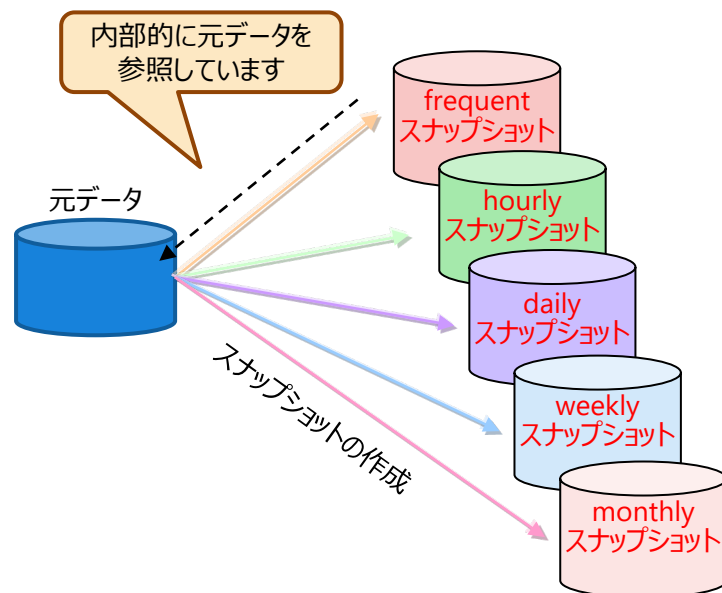
【書式】 zfs rollback <スナップショット名>

```
# zfs rollback -r rz2pool/data1@before
```



- 最新ではないスナップショットには、ロールバックできません。
- 最新のスナップショット以外にロールバックするには、対象のスナップショットが最新の状態になるように、対象のスナップショットより新しいスナップショットをすべて破棄する必要があります。
- ロールバック時に-rオプションを指定することで、指定したスナップショットを最新としてロールバックが可能です。

- 一定の間隔でスナップショットを自動で作成します。



	作成タイミング	保持可能な数
frequentスナップショット	15分ごと	4
hourlyスナップショット	毎時	24
dailyスナップショット	毎日	31
weeklyスナップショット	毎週	7
monthlyスナップショット	毎月	12

- ✓ 自動スナップショットプロパティの設定変更 (com.sun:auto-snapshotプロパティ)

【書式】 zfs set com.sun:auto-snapshot=[true | false] <対象とするファイルシステム名>

```
# zfs set com.sun:auto-snapshot=true rz2pool/data1
```

- ✓ 自動スナップショットサービスの有効化

【書式】 svcadm enable svc:/system/filesystem/zfs/auto-snapshot:[frequent | hourly | daily | weekly | monthly]

```
# svcadm enable svc:/system/filesystem/zfs/auto-snapshot:frequent
# svcadm enable svc:/system/filesystem/zfs/auto-snapshot:hourly
```



• 自動スナップショット機能を有効にするデータセット、および作成タイミングを指定することで、ディスクを有効活用できます。

4.バックアップ／リストア

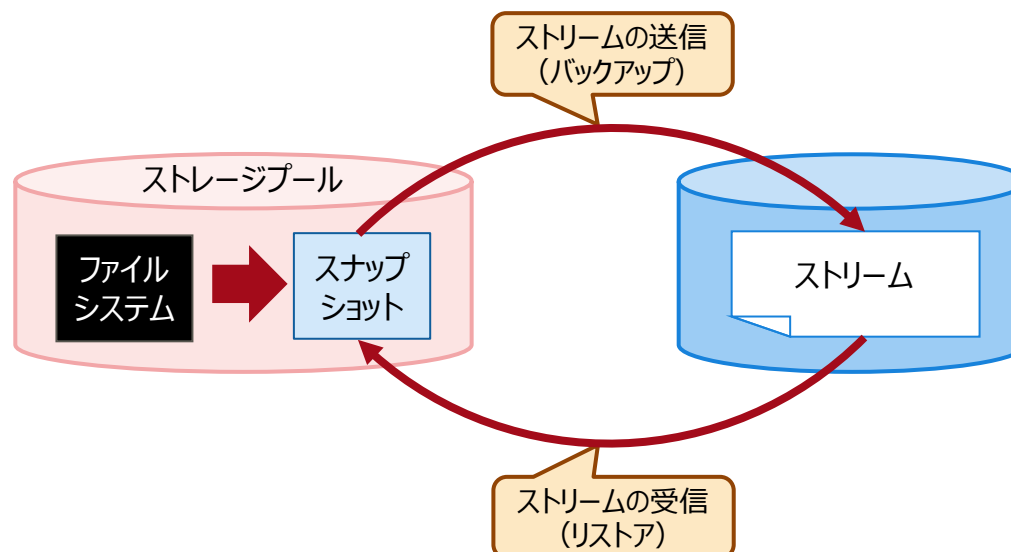
ZFSでは、ストリームの送信／受信によって、ファイルシステムのバックアップ／リストアを実現します。

ここでは、ストリームの送信／受信の方法を説明します。

※ システムボリュームのバックアップ／リストアについては、『Oracle Solaris 11を使ってみよう（構築・運用ガイド）』を参照してください。

■ バックアップ／リストアのながれ

ZFSでは、ストリームの送信／受信によって、ファイルシステムのバックアップ／リストアを実現します。
本章では、ストリームの送信／受信の方法を説明します。

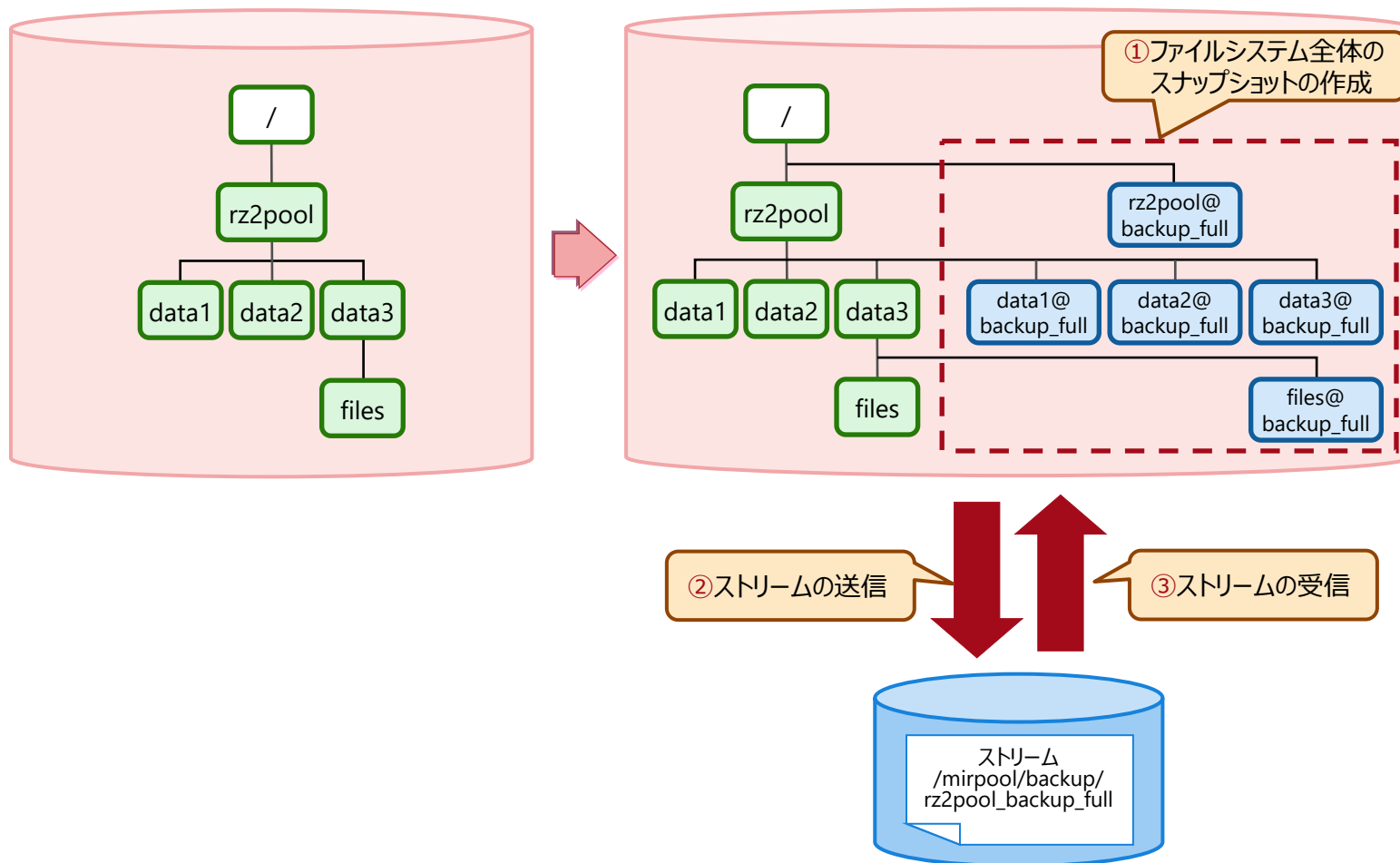


■ バックアップ／リストアの方法

- [方法1：フルバックアップ／リストア](#)
ストレージプール内のすべてのファイルシステムをバックアップ／リストアします。
- [方法2：特定のファイルシステムのバックアップ／リストア](#)
- [方法3：増分バックアップ／リストア](#)
ストレージプール内のすべてのファイルシステムを増分バックアップ／増分リストアします。

方法1：フルバックアップ／リストア 1/2

- ストレージプール内のすべてのファイルシステムをバックアップ／リストアします。



■ バックアップ

① ファイルシステム全体のスナップショットの作成

-r オプションを指定すると、スナップショット作成時に指定したファイルシステムと、すべての下位ファイルシステムのスナップショットを同時に作成します。

```
# zfs snapshot -r rz2pool@backup_full
```

② ストリームの送信

-R オプションで指定されたスナップショットの、すべての下位ファイルシステムのストリームを送信します。

※ ストリームの送信先は任意です。本書では送信先に /mirpool/backup 配下を指定しています。

```
# zfs send -Rv rz2pool@backup_full > /mirpool/backup/rz2pool_backup_full
```

1つのファイルに保存する場合は、gzipコマンドを使用して圧縮します。

```
# zfs send -Rv rz2pool@backup_full | gzip > /mirpool/backup/rz2pool_backup_full.gz
```

■ リストア

③ ストリームの受信

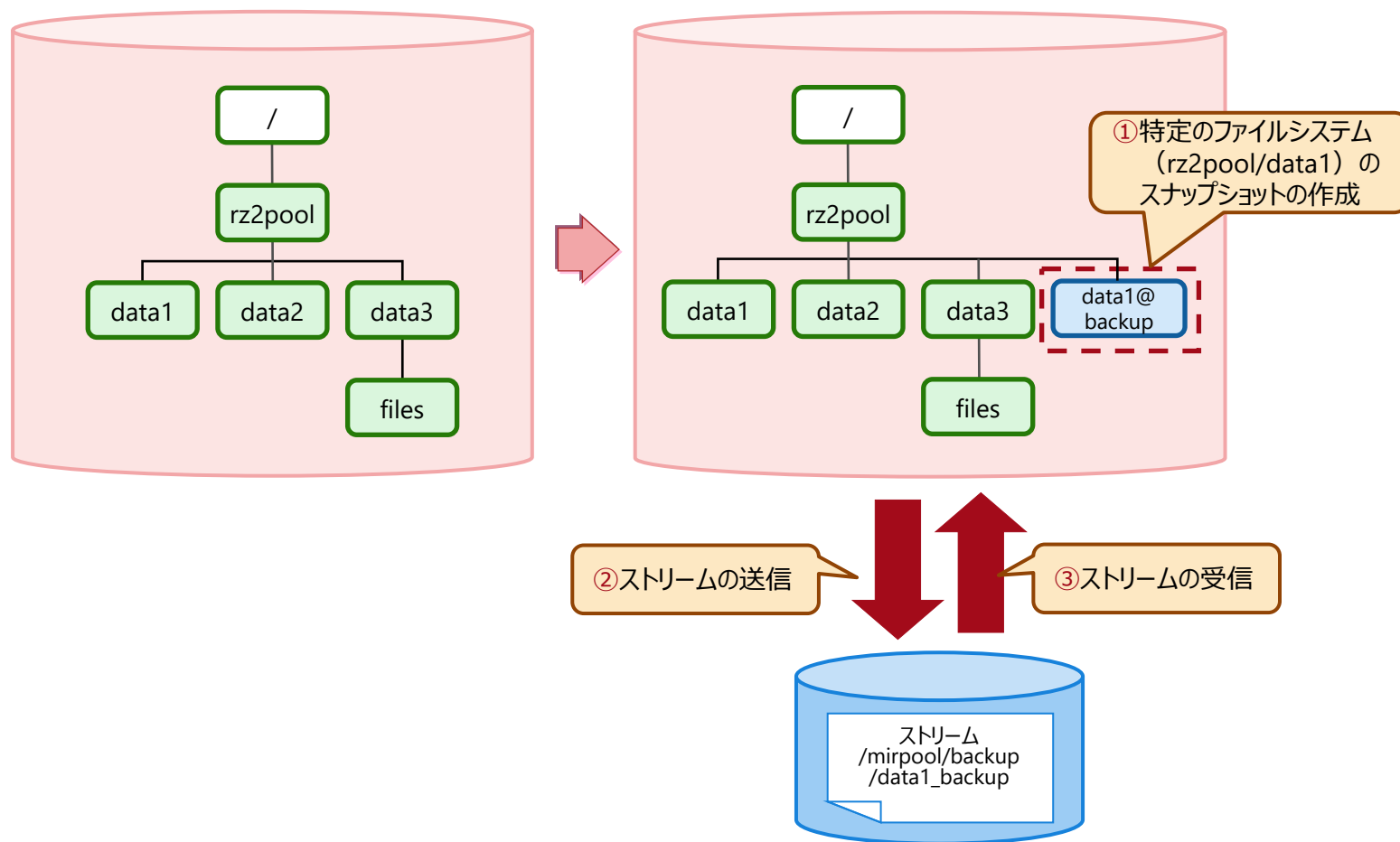
```
# zfs receive -F -d -v rz2pool < /mirpool/backup/rz2pool_backup_full
```

gzipコマンドで圧縮されたストリームを受信する場合

```
# gzcatt /mirpool/backup/rz2pool_backup_full.gz | zfs receive -F -d -v rz2pool
```

※ -R オプションを指定して作成したストリームを-d オプションを指定せずに受信すると、すべての下位ファイルシステムが正しい名称で復元されません。

■ 特定のファイルシステムをバックアップ／リストアします



①特定のファイルシステムのスナップショットの作成

```
# zfs snapshot rz2pool/data1@backup
```

②ストリームの送信

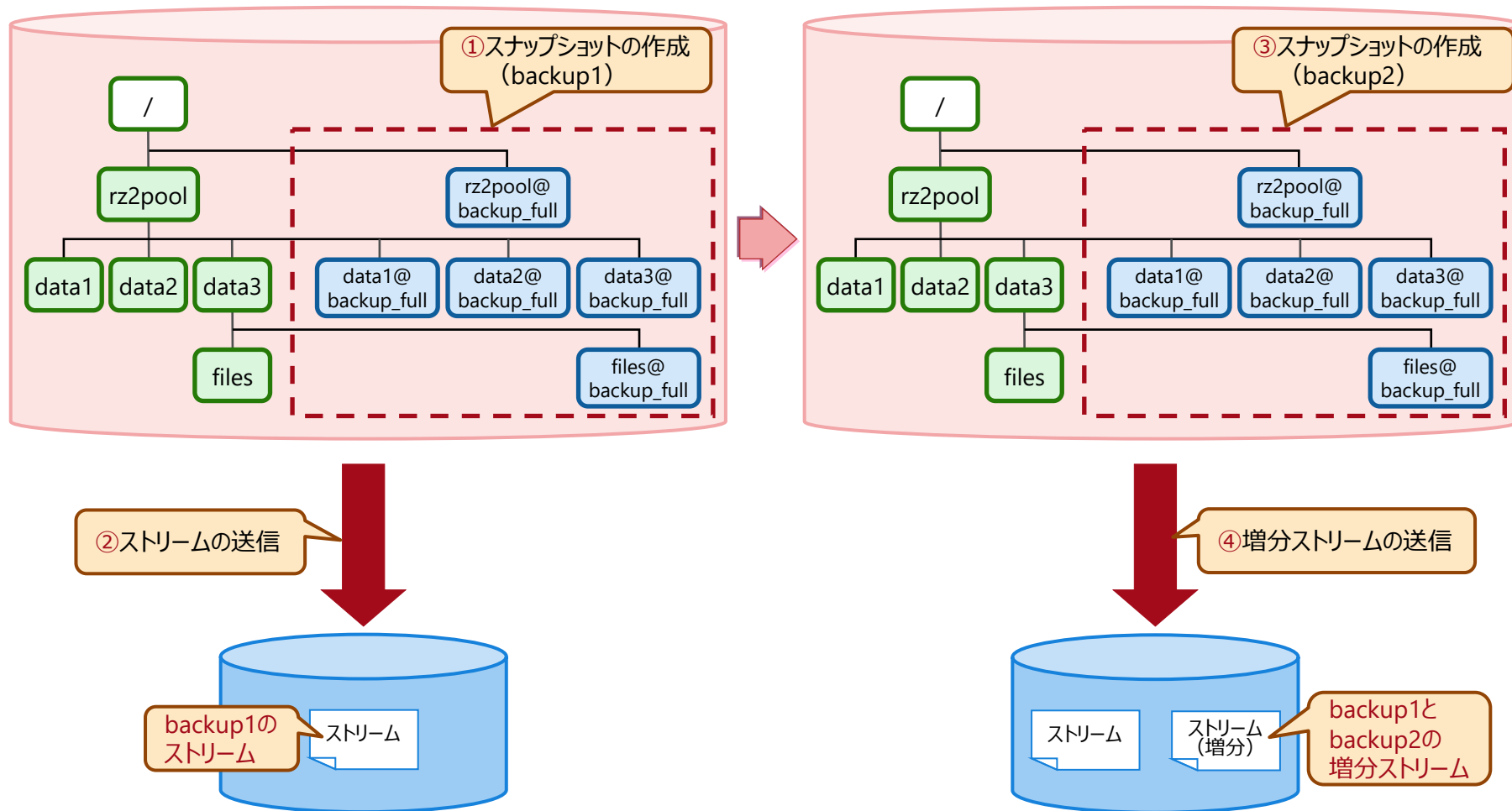
```
# zfs send -v rz2pool/data1@backup > /mirpool/backup/data1_backup
```

③ストリームの受信

```
# zfs receive -v rz2pool/data1 < /mirpool/backup/data1_backup
```

方法3：増分バックアップ／リストア 1/4

- ストレージプール内のすべてのファイルシステムを増分バックアップします。



①スナップショットの作成 (backup1)

```
# zfs snapshot -r rz2pool@backup1
```

②ストリームの送信

```
# zfs send -Rv rz2pool@backup1 > /mirpool/backup/rz2pool_backup1
```

③スナップショットの作成 (backup2)

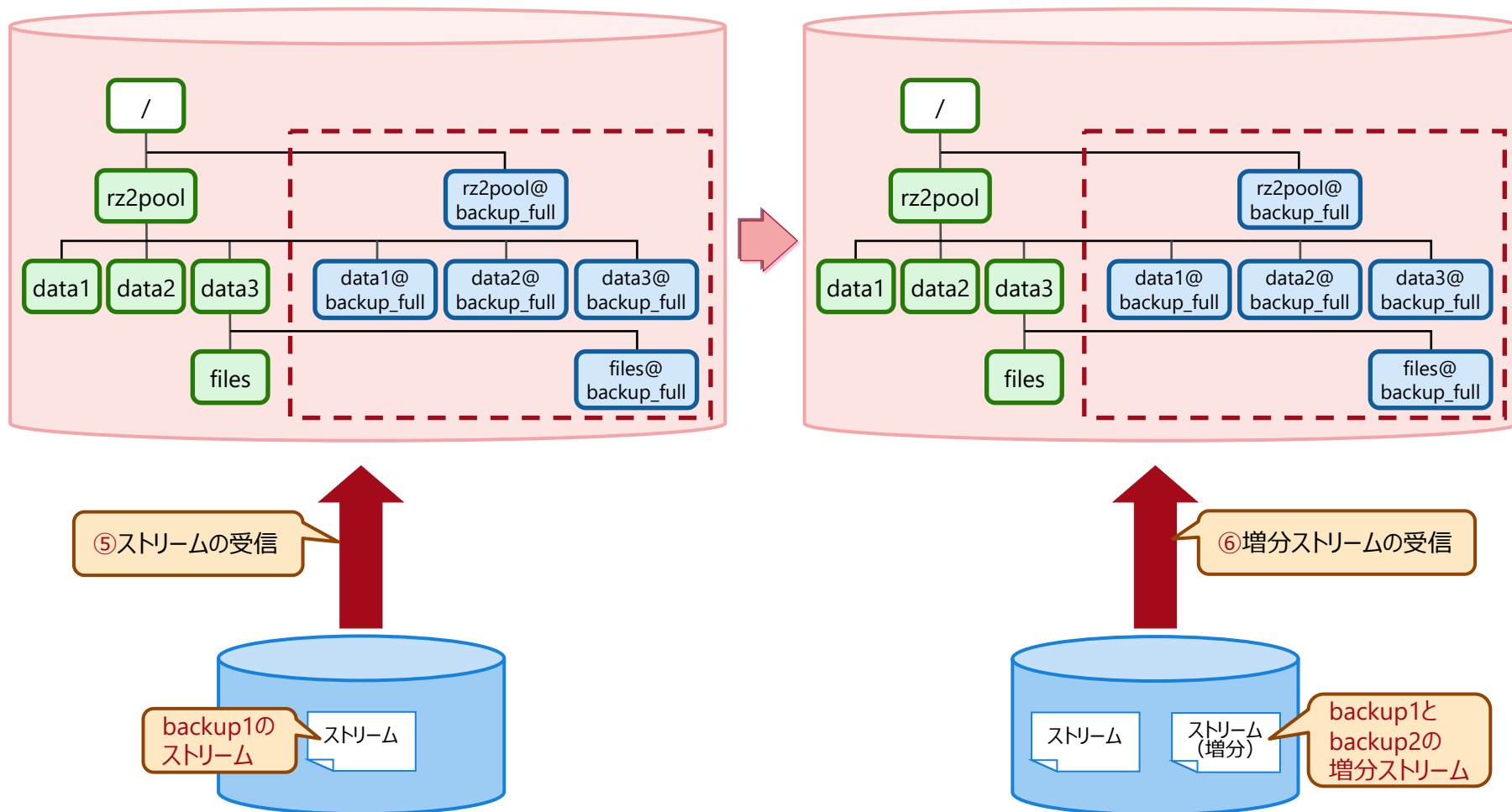
```
# zfs snapshot -r rz2pool@backup2
```

④増分ストリームの送信

```
# zfs send -Rv -i rz2pool@backup1 rz2pool@backup2 > /mirpool/backup/rz2pool_backup2
```

方法3：増分バックアップ／リストア 3/4

- backup1のストリーム、およびbackup1とbackup2の増分ストリームを受信し、リストアします。



■ リストア

⑤ ストリームの受信

```
# zfs receive -F -d -v rz2pool < /mirpool/backup/rz2pool_backup1
```

⑥ 増分ストリームの受信

```
# zfs receive -F -d -v rz2pool < /mirpool/backup/rz2pool_backup2
```


《参考》ストリーム送信時の進捗表示

Solaris 11.2からzfs sendコマンドの進捗状況を確認する機能が追加されました。

- pvコマンドを間に挿入することで、zfs sendコマンドの進捗状況を確認できます。ZFSの退避に関するスケジュールや、計画の精度改善に役立ちます。

```
# zfs send -Rv rpool@backup | pv | gzip > /mirpool/backup/rpool.zfs.gz
estimating full stream rpool@backup (size = 82.1K)
estimating full stream rpool/VARSHARE@backup (size = 2.75M)
estimating full stream rpool/VARSHARE/sstore@backup (size = 91.1M)
estimating full stream rpool/VARSHARE/zones@backup (size = 39.6K)
estimating full stream rpool/VARSHARE/pkg@backup (size = 40.5K)
estimating full stream rpool/VARSHARE/pkg/repositories@backup (size = 39.6K)
estimating full stream rpool/VARSHARE/kvol@backup (size = 39.6K)
estimating full stream rpool/VARSHARE/tmp@backup (size = 39.6K)
estimating full stream rpool/export@backup (size = 40.5K)
estimating full stream rpool/export/home@backup (size = 40.5K)
estimating full stream rpool/export/home/user01@backup (size = 42.4K)
estimating full stream rpool/ROOT@backup (size = 39.6K)
~ (省略) ~
estimating clone stream rpool/ROOT/solaris@backup (size = 87.0M)
estimating clone stream rpool/ROOT/solaris/var@backup (size = 284K)
estimated total stream size: 4.76G
1.56GiB 0:07:04 [3.77MiB/s] [ <=> ]
```

コマンド実行直後に
バックアップ対象の
すべてのスナップ
ショットを表示

実行後に作成されるzfsストリーム
ファイルのサイズを表示
(圧縮前のサイズ)

作成途中のサイズ、
経過時間、速度を表示

5.ミラーディスクの分割

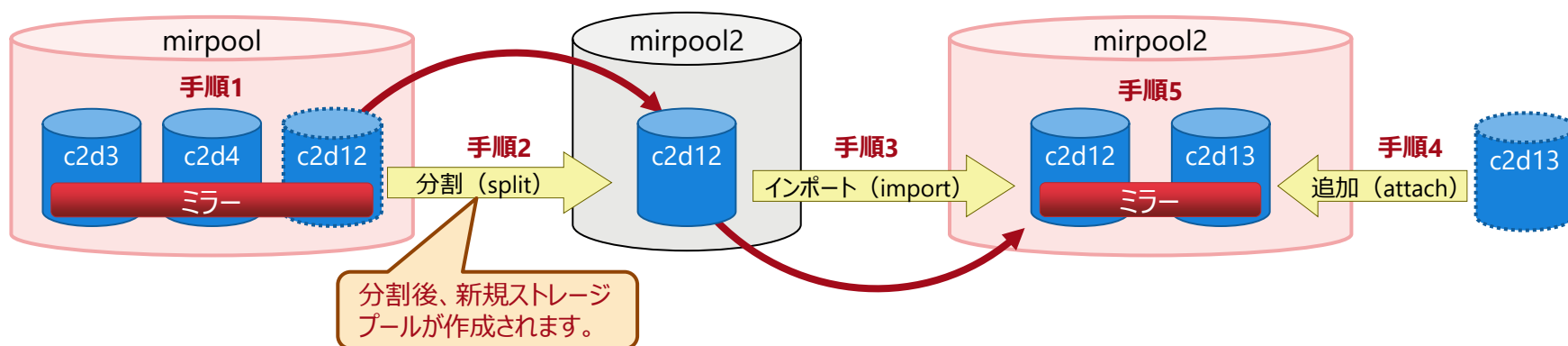
ミラーディスクを分割し、分割したディスクから新しいストレージプールを作成する方法を説明します。

■ ミラーディスクの分割の環境

本章では、ミラーディスクを分割し、分割したディスクから新たにストレージプールを作成する方法を説明します。

ミラー構成のストレージプールからミラーディスクを分割し、分割したミラーディスクをインポートすることで、分割元と同一のファイルシステムとデータを保持した新規ストレージプールを作成します。インポート後は、ディスクを追加して2面ミラー構成とします。

構築する環境は以下のとおりです。



■ ミラーディスクの分割のながれ

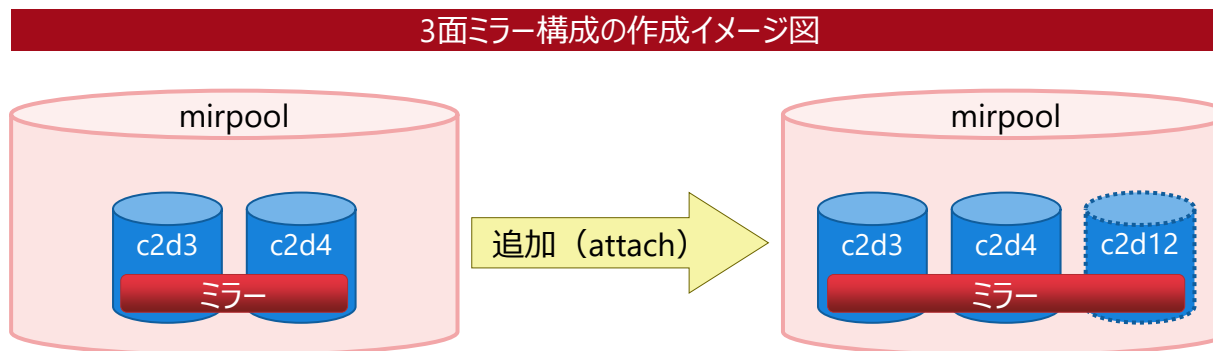
- [手順1：ミラー構成の変更](#)
- [手順2：ディスクの分割と新規ストレージプールの作成](#)
- [手順3：新規ストレージプールのインポート](#)
- [手順4：新規ストレージプールでのミラー構成の作成](#)
- [手順5：新規ストレージプールの構成確認](#)



• 新規ストレージプールは、容易に別サーバへ移動できます。

手順1：ミラー構成の変更

- 2面ミラー構成から3面ミラー構成を作成し、ミラーディスクの分割の準備をします。



✓ ディスクの追加（zpool attachコマンド）

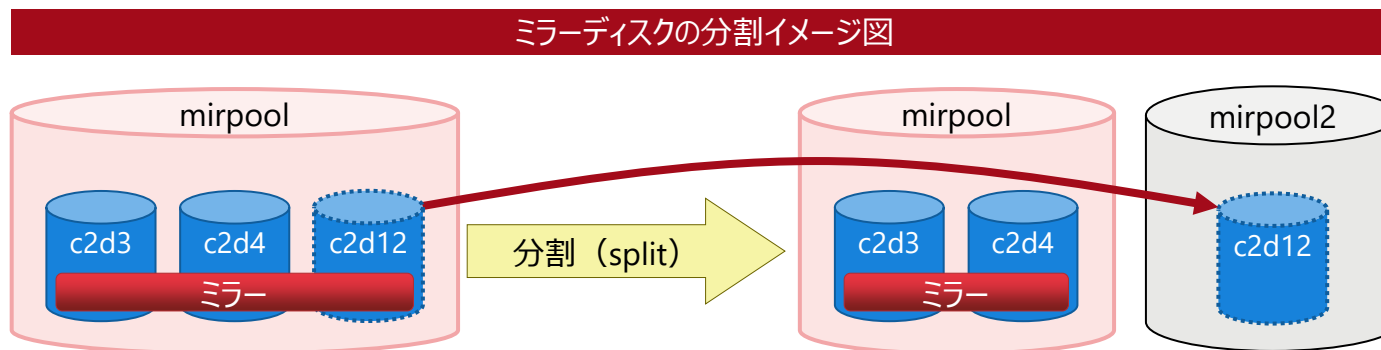
【書式】 zpool attach <プール名> <ミラー元ディスク> <ミラーディスク>

```
# zpool attach mirpool c2d3 c2d12
```



• 2面ミラー構成のミラーディスクからディスクを分割することもできますが、分割後はディスクの冗長性が失われます。ディスクの冗長性を維持するため、3面ミラー構成を推奨します。

- 3面ミラー構成のディスクを分割し、新規ストレージプールを作成します。



✓ ストレージプールからのディスクの分割（zpool splitコマンド）

【書式】 zpool split <ストレージプール名> <新規ストレージプール名> [分割するディスク名]

```
# zpool split mirpool mirpool2
```

※ 分割するディスク名を指定しない場合、デフォルト設定では最後にストレージプールへ接続されたディスク（ここではc2d12）が分割されます。
zpool statusコマンドの表示順が、ディスクの接続順となります。

※ 作成されたストレージプール（mirpool2）には、作成元のストレージプール（mirpool）と同一のデータが存在します。

✓ 《参考》指定したディスクをストレージプールから分割する

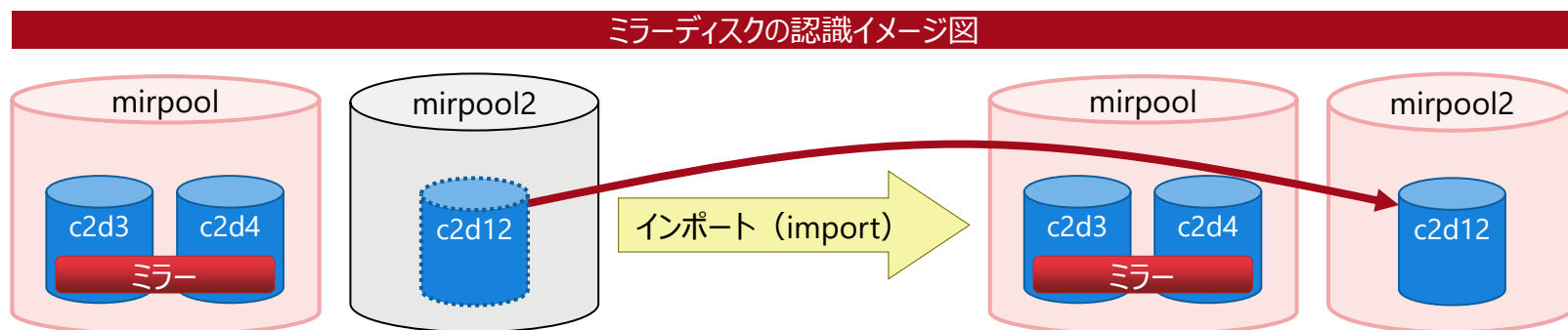
```
# zpool split mirpool mirpool2 c2d12
```



- 少ない負荷で、ストレージプールからディスクを分割することができます。
ディスク分割中のディスク負荷については、「[《参考》zpool split実行時のディスク負荷率](#)」を参照してください。
- ディスクの分割と同時に新規ストレージプールが作成され、エクスポート（export）されます。
※再同期化が進行中の場合は、ディスクを分割できません。

手順3：新規ストレージプールのインポート

- ディスクの分割で作成された新規ストレージプールをインポートします。
 - ・ ミラーディスクの分割によって作成される新規ストレージプールは、接続が解除（エクスポート）された状態で作成されます。
 - ・ 新規ストレージプールを使用する場合は、手動でインポートを実行します。



- ✓ ディスクの分割で作成された新規ストレージプールのインポート（zpool importコマンド）
【書式】zpool import <ストレージプール名>

```
# zpool import mirpool2
```

※ ストレージプール名を指定しない場合、インポート可能なストレージプールを確認できます。

```
# zpool import
pool: mirpool2
id: 2478347627476883301
state: ONLINE
action: The pool can be imported using its name or numeric identifier.
config:

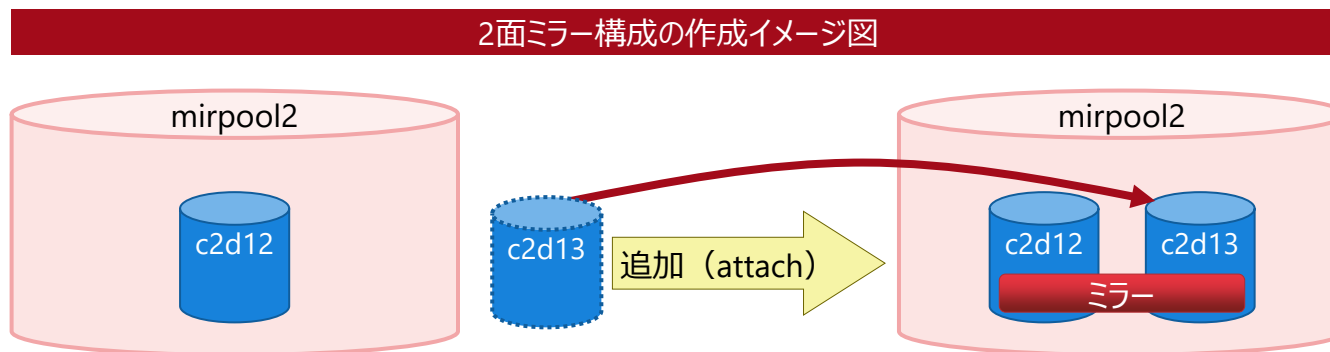
    mirpool2  ONLINE
    c2d12     ONLINE
```



- ・ 少ない負荷で、ディスクをインポートすることができます。
分割したディスクのインポート中のディスク負荷については、「[《参考》zpool import実行時のディスク負荷率](#)」を参照してください。

手順4：新規ストレージプールでのミラー構成の作成

- 新規ストレージプールにディスクを追加し、2面ミラー構成を作成します。



- ✓ ディスクの追加（zpool attachコマンド）

【書式】zpool attach <プール名> <ミラー元ディスク> <ミラーディスク>

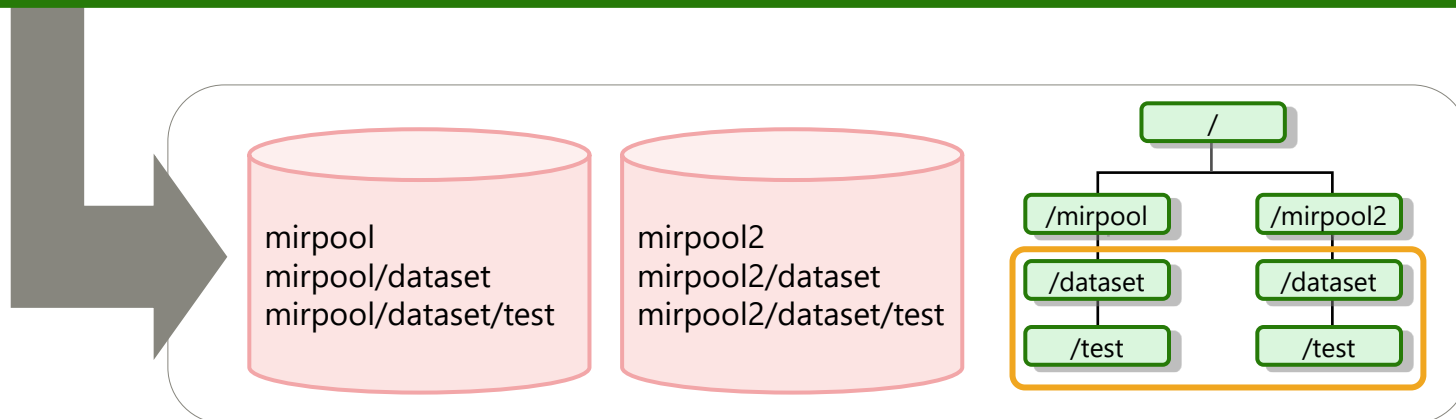
```
# zpool attach mirpool2 c2d12 c2d13
```

手順5：新規ストレージプールの構成確認

■ 新規ストレージプールの構成を確認します。

✓ ファイルシステムの確認（zfs listコマンド）

```
# zfs list
NAME                                USED    AVAIL    REFER  MOUNTPOINT
mirpool                            1.57G   27.7G   1.57G   /mirpool
mirpool/dataset                    63.5K   27.7G    32K    /mirpool/dataset
mirpool/dataset/test               31.5K   27.7G   31.5K   /mirpool/dataset/test
mirpool2                            1.57G   27.7G   1.57G   /mirpool2
mirpool2/dataset                   63.5K   27.7G    32K    /mirpool2/dataset
mirpool2/dataset/test              31.5K   27.7G   31.5K   /mirpool2/dataset/test
~ (省略) ~
```



• mirpoolとmirpool2のファイルシステム構成が、同じであることを確認します。

《参考》zpool split実行時のディスク負荷率

- 少ない負荷でストレージプールからディスクを分割することができます。

✓ zpool split 実行5秒間のディスク負荷率

```
# iostat -sxnp 1

              extended device statistics
    r/s    w/s   kr/s   kw/s wait actv wsvc_t asvc_t  %w  %b device
    0.0    0.0    0.0    0.0  0.0  0.0   0.0   0.0   0   0  c2d3
    0.0    0.0    0.0    0.0  0.0  0.0   0.0   0.0   0   0  c2d4
    0.0    0.0    0.0    0.0  0.0  0.0   0.0   0.0   0   0  c2d12

              extended device statistics
    r/s    w/s   kr/s   kw/s wait actv wsvc_t asvc_t  %w  %b device
    4.0  121.8  135.8  652.8  0.0  2.8   0.0  22.6   0  69  c2d3
    4.0  122.8  135.8  652.8  0.0  2.8   0.0  22.3   0  70  c2d4
    3.0   88.8   24.0 1120.0  0.0  2.7   0.0  29.2   0  62  c2d12

              extended device statistics
    r/s    w/s   kr/s   kw/s wait actv wsvc_t asvc_t  %w  %b device
    0.0   48.0    0.0  540.9  0.0  0.5   0.0  10.7   0  10  c2d3
    0.0   49.0    0.0  540.9  0.0  0.7   0.0  14.0   0  13  c2d4
   51.0 142.1  585.0 1172.4  0.0  1.1   0.0   5.5   0  26  c2d12

              extended device statistics
    r/s    w/s   kr/s   kw/s wait actv wsvc_t asvc_t  %w  %b device
    0.0    0.0    0.0    0.0  0.0  0.0   0.0   0.0   0   0  c2d3
    0.0    0.0    0.0    0.0  0.0  0.0   0.0   0.0   0   0  c2d4
    0.0    0.0    0.0    0.0  0.0  0.0   0.0   0.0   0   0  c2d12

    extended device statistics
    r/s    w/s   kr/s   kw/s wait actv wsvc_t asvc_t  %w  %b device
    0.0    0.0    0.0    0.0  0.0  0.0   0.0   0.0   0   0  c2d3
    0.0    0.0    0.0    0.0  0.0  0.0   0.0   0.0   0   0  c2d4
    0.0    0.0    0.0    0.0  0.0  0.0   0.0   0.0   0   0  c2d12
```

各項目の内容

項目	説明
r/s	1秒あたりの読み込み数
w/s	1秒あたりの書き込み数
kr/s	1秒あたりの読み込みキロバイト数
kw/s	1秒あたりの書き込みキロバイト数
wait	実行待ちのI/O要求数の平均値
actv	処理中のI/O要求数の平均値
wsvc_t	I/O要求の待機時間の平均値 (ミリ秒単位)
asvc_t	I/O要求の処理時間の平均値 (ミリ秒単位)
%w	待機中のI/O要求が存在する時間の割合
%b	ディスクがビジー状態になっている時間の割合
device	デバイス名



- 上記は、iostat -sxnp 1コマンドで1秒間ごとの入出力統計情報を出力した結果のうち、5秒間を抜粋したものとなります。
 - ディスク負荷率は、%b列を確認します。数値が0以外のときにディスク負荷が発生しており、数値が高いほどディスク負荷率が高いと判断できます。
 - 上記の例では、zpool split（分割）実行時に、約2秒間だけディスク負荷が発生していることが確認できます。
- ※ 測定値は、検証環境での参考値です。

《参考》zpool import実行時のディスク負荷率

■ 少ない負荷でディスクをインポートすることができます。

✓ zpool import 実行5秒間のディスク負荷率

```
# iostat -sxnp 1

              extended device statistics
    r/s    w/s    kr/s    kw/s wait actv wsvc_t asvc_t  %w  %b device
    0.0    0.0    0.0    0.0  0.0  0.0  0.0    0.0  0  0  c2d3
    0.0    0.0    0.0    0.0  0.0  0.0  0.0    0.0  0  0  c2d4
    0.0    0.0    0.0    0.0  0.0  0.0  0.0    0.0  0  0  c2d12

              extended device statistics
    r/s    w/s    kr/s    kw/s wait actv wsvc_t asvc_t  %w  %b device
    10.0    0.0 1277.7    0.0  0.0  0.0  0.0    0.1  0  0  c2d3
    10.0    0.0 1277.7    0.0  0.0  0.0  0.0    0.1  0  0  c2d4
    184.7  308.4 5437.1 1937.5  0.0  2.2  0.0    4.6  0 47  c2d12

              extended device statistics
    r/s    w/s    kr/s    kw/s wait actv wsvc_t asvc_t  %w  %b device
    0.0    0.0    0.0    0.0  0.0  0.0  0.0    0.0  0  0  c2d3
    0.0    0.0    0.0    0.0  0.0  0.0  0.0    0.0  0  0  c2d4
    6.0    26.0    5.5   376.3  0.0  0.3  0.0    9.0  0  5  c2d12

              extended device statistics
    r/s    w/s    kr/s    kw/s wait actv wsvc_t asvc_t  %w  %b device
    0.0    0.0    0.0    0.0  0.0  0.0  0.0    0.0  0  0  c2d3
    0.0    0.0    0.0    0.0  0.0  0.0  0.0    0.0  0  0  c2d4
    0.0    0.0    0.0    0.0  0.0  0.0  0.0    0.0  0  0  c2d12

    extended device statistics
    r/s    w/s    kr/s    kw/s wait actv wsvc_t asvc_t  %w  %b device
    0.0    0.0    0.0    0.0  0.0  0.0  0.0    0.0  0  0  c2d3
    0.0    0.0    0.0    0.0  0.0  0.0  0.0    0.0  0  0  c2d4
    0.0    0.0    0.0    0.0  0.0  0.0  0.0    0.0  0  0  c2d12
```

各項目の内容

項目	説明
r/s	1秒あたりの読み込み数
w/s	1秒あたりの書き込み数
kr/s	1秒あたりの読み込みキロバイト数
kw/s	1秒あたりの書き込みキロバイト数
wait	実行待ちのI/O要求数の平均値
actv	処理中のI/O要求数の平均値
wsvc_t	I/O要求の待機時間の平均値 (ミリ秒単位)
asvc_t	I/O要求の処理時間の平均値 (ミリ秒単位)
%w	待機中のI/O要求が存在する時間の割合
%b	ディスクがビジー状態になっている時間の割合
device	デバイス名



- 上記は、iostat -sxnp 1コマンドで1秒間ごとの入出力統計情報を出力した結果のうち、5秒間を抜粋したものとなります。
 - ディスク負荷率は、%b列を確認します。数値が0以外のときにディスク負荷が発生しており、数値が高いほどディスク負荷率が高いと判断できます。
 - 上記の例では、zpool import（インポート）実行時に、約2秒間だけディスク負荷が発生していることが確認できます。
- ※ 測定値は、検証環境での参考値です。

参考情報

- ストレージプール、またはストレージプールを構成するデバイス（ディスク） ごとに入出力統計情報を確認できます。

- Solaris標準のiostatコマンドと同様に、ZFSにもデータの入出力の統計情報を表示するコマンドが実装されています。

✓ ストレージプールの統計情報（zpool iostatコマンド）

【書式】 zpool iostat [-v] <プール名> <interval（秒）> [回数]>

```
# zpool iostat rz2pool 5 5
          capacity      operations      bandwidth
pool    alloc  free    read  write    read  write
-----
rz2pool 10.1G  79.4G      0     0        1  14.8K
rz2pool 10.3G  79.2G      0    77        0  7.95M
rz2pool 10.3G  79.2G      0    54        0  6.75M
rz2pool 10.3G  79.2G      0    64        0  7.39M
rz2pool 10.5G  79.0G      0    50        0  5.14M
```

表示間隔を指定します。
この例では、5秒間隔で表示します。

✓ デバイスの統計情報

-vオプションを指定すると、仮想デバイス全体のレイアウトと入出力統計を取得できます。

```
# zpool iostat -v rz2pool 5 5
          capacity      operations      bandwidth
pool    alloc  free    read  write    read  write
-----
rz2pool 49.9M  89.5G      0     60     102  4.73M
raidz2-0 49.9M  89.5G      0     60     102  4.73M
c2d7      -      -        0     56     102  4.73M
c2d8      -      -        0     56        0  4.68M
c2d9      -      -        0     57        0  4.73M
-----
```

～（省略）～

項目	説明
pool	ストレージプール名、デバイス名
capacity	
alloc	使用済みデータサイズ
free	使用可能データサイズ
operations	
read	入出力読み取り操作数
write	入出力書き込み操作数
bandwidth	
read	読み取り操作の帯域幅（1秒あたりのデータ量）
write	読み取り操作の帯域幅（1秒あたりのデータ量）

ストレージプールの移行 (export/import)

■ ストレージプールをエクスポートし、別システムにインポートできます。

✓ ストレージプールのエクスポート (zpool exportコマンド)

エクスポートしたシステムからストレージプールは認識されなくなります。

```
# zpool export mirpool
```

✓ ストレージプールのインポート (zpool importコマンド)

インポート可能なストレージプールを確認します。

```
# zpool import
pool: mirpool
id: 3833992751035069071
state: ONLINE
action: The pool can be imported using its name or numeric identifier.
config:

    mirpool      ONLINE
        mirror-0  ONLINE
            c2d3   ONLINE
            c2d4   ONLINE
    spares
        c2d10
```

ストレージプールをインポートします。

```
# zpool import mirpool
```

■ ストレージプール、およびファイルシステムの操作を監視します。

✓ ZFS操作の監視（zpool monitorコマンド）

【書式】 zpool monitor -t <監視するタスク> [-T d|u] [[-p] -o <表示フィールド>[...]] <pool> ... <interval（秒）> [回数]>

- -t <監視するタスク> : 監視するタスク操作を指定します。
- -o <表示フィールド> : 表示するフィールドを指定します。カンマ区切りで複数指定可能です。
指定しない場合は、デフォルトのフィールドが表示されます。

監視するタスク		
タスク	説明	備考
send	ZFSデータの送信	zfsコマンド
receive(recv)	ZFSデータの受信	zfsコマンド
destroy	スナップショット破棄	zfsコマンド
scrub	ファイルシステム検証	zpoolコマンド
resilver	プール再同期	zpoolコマンド

表示フィールド		
表示フィールド	説明	デフォルト表示
DONE	処理されたデータ量	
OTHER	処理中の項目やタスク状態など	
PCTDONE	処理されたデータの割合	○
POOL	監視するプール名	○
PROVIDER	監視するタスク	○
SPEED	1 秒あたりの単位	○
STRTIME	タスクが起動された時間	
TAG	操作全体を区別する一意の値	
TIMELEFT	タスクが完了する相対時間	○
TIMESTAMP	スナップショットが取得される時間	
TOTAL	処理されるデータの合計量の見積もり	○
all	すべてのフィールドを表示	

```
# zpool monitor -t scrub 5
POOL      PROVIDER  PCTDONE  TOTAL  SPEED  TIMELEFT
rz2pool   scrub      3.3      11.5G  130M   1m27s
rz2pool   scrub      9.7      11.5G  143M   1m14s
rz2pool   scrub     17.5     11.5G  158M   1m01s
rz2pool   scrub     24.8     11.5G  162M   54s
~（省略）~
```

※ 進捗状況は、[Ctrl]+[C]キーを押すか、該当の処理が完了するまで出力され続けます。
intervalを指定しなかった場合は、コマンド実行時の状況が表示されます。

- ZFS関連の操作は記録されており、zpool historyコマンドによってZFS関連コマンドの履歴を表示できます。

- ✓ ZFS関連コマンド履歴の表示 (zpool historyコマンド)

```
# zpool history rz2pool
History for 'rz2pool':
2019-05-07.16:41:27 zpool create rz2pool raidz c2d7 c2d8 c2d9
2019-05-08.15:51:30 zfs create rz2pool/data1
2019-05-08.15:51:35 zfs create rz2pool/data2
2019-05-08.15:51:47 zfs create -p rz2pool/data3/filedir
2019-05-08.15:52:18 zfs create -o mountpoint=/zfs/fs4 rz2pool/data4
2019-05-08.15:52:55 zfs create -p -V 2G rz2pool/data4/user1
2019-05-08.15:54:34 zfs rename rz2pool/data3/filedir rz2pool/data3/files
~ (省略) ~
```

付録

『Oracle Solaris 11.4 での ZFS ファイルシステムの管理』（Oracle社）

https://docs.oracle.com/cd/E75431_01/pdf/E75197.pdf

SPARCサーバ／Oracle Solarisの技術情報を掲載



今すぐアクセス！！！！

<https://www.fujitsu.com/jp/sparc-technical/>

版数	更新日時	更新内容
初版	2012年8月	新規作成
第2.0版	2013年4月	SPARC M10 および、Oracle Solaris 11.1 に対応
第3.0版	2015年6月	Oracle Solaris 11.2 に対応
第4.0版	2016年5月	Oracle Solaris 11.3 に対応
第5.0版	2019年6月	SPARC M12 および、Oracle Solaris 11.4 に対応

■ 使用条件

■ 著作権・商標権・その他の知的財産権について

- コンテンツ（文書・画像・音声等）は、著作権・商標権・その他の知的財産権で保護されています。本コンテンツは、個人的に使用する範囲でプリントアウトまたはダウンロードできます。ただし、これ以外の利用（ご自分のページへの再利用や他のサーバへのアップロード等）については、当社または権利者の許諾が必要となります。

■ 保証の制限

- 本コンテンツについて、当社は、その正確性、商品性、ご利用目的への適合性等に関して保証するものではなく、そのご利用により生じた損害について、当社は法律上のいかなる責任も負いかねます。本コンテンツは、予告なく変更・廃止されることがあります。

■ 輸出または提供

- 本製品を輸出又は提供する場合は、外国為替及び外国貿易法及び米国輸出管理関連法規等の規制をご確認の上、必要な手続きをおとり下さい。

■ 商標

- UNIXは、米国およびその他の国におけるオープン・グループの登録商標です。
- SPARC Enterprise、SPARC64、SPARC64 ロゴおよびすべてのSPARC商標は、米国SPARC International, Inc.のライセンスを受けて使用している、同社の米国およびその他の国における商標または登録商標です。
- OracleとJavaは、Oracle Corporation およびその子会社、関連会社の米国およびその他の国における登録商標です。
- その他各種製品名は、各社の製品名称、商標または登録商標です。

