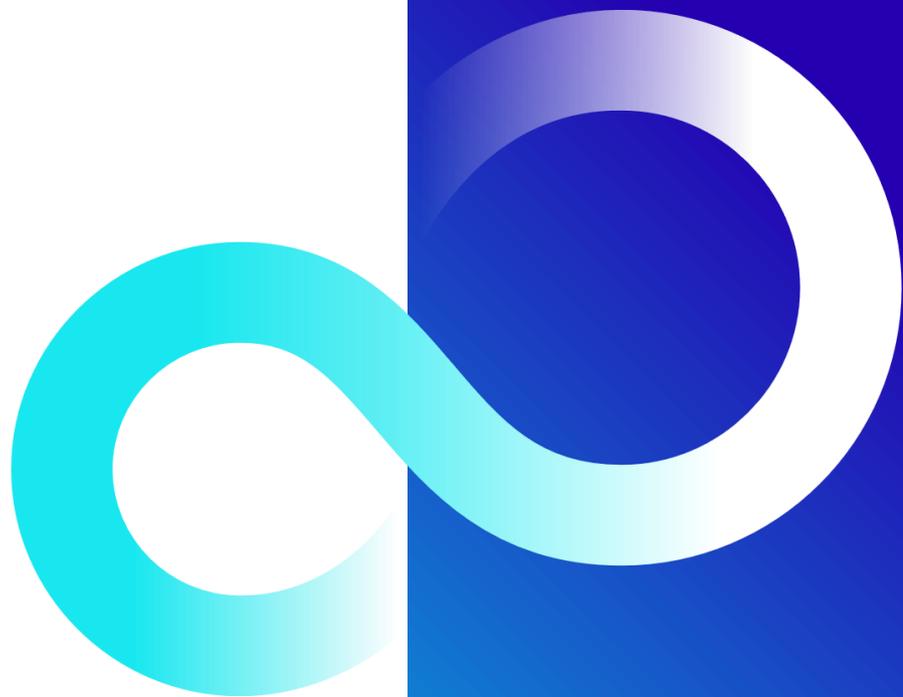


Oracle Solaris 11  
ZFSを使ってみよう  
(構築・運用手順書)



2019年6月

第2.0版

富士通株式会社

## ■ 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。

## ■ 商標について

- 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 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。

## はじめに

### 本書の内容

- Oracle Solaris 11 ZFS の構築・運用手順を紹介しています。  
別冊の『Oracle Solaris 11 ZFS を使ってみよう(構築・運用ガイド)』を参照しながら、本手順書を活用してください。
- Oracle Solaris 11 ZFS を使ってみよう(構築・運用ガイド)  
<https://www.fujitsu.com/jp/sparc-technical/document/solaris/#solaris11zfs>

### 留意事項

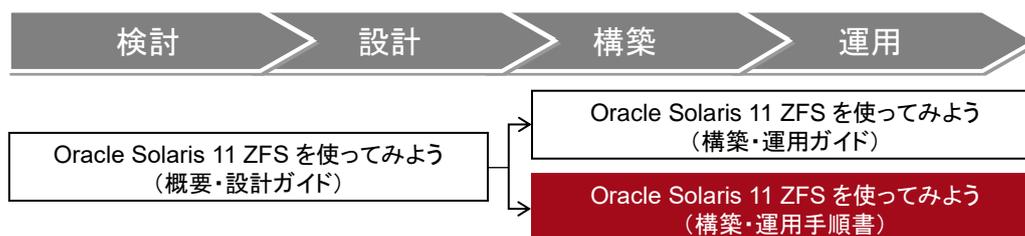
- 本資料は、Oracle Solaris 11.4 を使用した手順を紹介しています。
- 本資料では、Solaris 環境を Oracle VM Server for SPARC 上に構築した場合の手順を記載しています。一部 Oracle VM Server for SPARC に依存した内容がありますのでご了承ください。  
例：ディスクやネットワーク・インターフェースなどのデバイス名が Oracle VM Server for SPARC 固有の値となっており、物理サーバ環境とは異なります。
- 本手順書に記載しているコマンドの実行結果は、環境によって異なることがあります。

### 本書での表記

- 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

### ドキュメントの位置づけ



## 目次

1. ストレージプールの構築 .....	1
1.1. ストレージプール作成のポイント .....	1
1.2. ルートプール(システム領域)の構成変更 .....	2
1.2.1. シングル構成から2面ミラー構成への変更 .....	2
1.2.2. 2面ミラー構成から3面ミラー構成への変更 .....	4
1.3. ストレージプール(ユーザー領域)の作成 .....	6
1.4. デバイスの追加(add) .....	12
1.5. デバイスの削除(remove) .....	14
1.6. デバイスの接続／切り離し(attach/detach) .....	18
1.6.1. デバイスの接続(attach) .....	18
1.6.2. デバイスの切り離し(detach) .....	22
1.7. デバイスのオンライン／オフライン(online/offline) .....	23
1.7.1. デバイスのオフライン(offline) .....	24
1.7.2. デバイスのオンライン(online) .....	26
1.8. エラーのクリア(clear) .....	27
1.9. デバイスの置き換え(replace) .....	28
1.10. ホットスペア(spare) .....	31
1.10.1. ホットスペアディスクの登録 .....	32
1.10.2. ホットスペアディスクのアクティブ化 .....	33
1.10.3. ホットスペアディスクの非アクティブ化 .....	34
2. ファイルシステムの構築 .....	36
2.1. ファイルシステムの作成 .....	36
2.2. ファイルシステム名の変更 .....	37
2.3. プロパティの設定 .....	38
2.3.1. マウントポイントの変更 .....	38
2.3.2. ファイルシステムの共有と共有解除 .....	39
2.3.3. ファイルシステムの使用可能領域の上限設定(quota プロパティ) .....	42
2.3.4. ユーザー／グループ割り当て制限 .....	43
2.3.5. データ重複の検知・排除 .....	44
2.3.6. データの暗号化 .....	45
2.3.7. プロパティの継承 .....	47

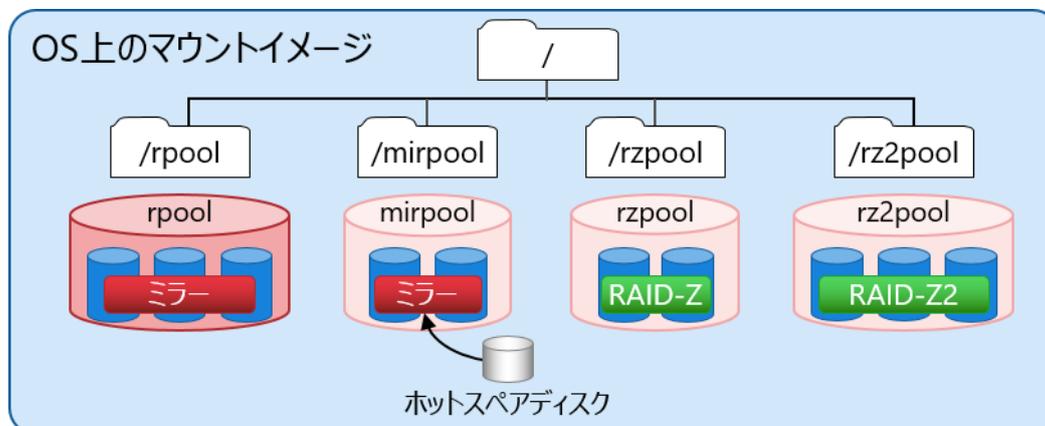
2.4. ファイルシステムの削除 .....	48
2.5. ファイルシステムのマウント／アンマウント.....	49
2.6. ファイルシステムミラーマウント .....	52
3. スナップショットとクローンの作成.....	54
3.1. スナップショットの作成.....	54
3.2. スナップショットの差分表示.....	57
3.3. クローンの作成 .....	59
3.4. ファイルシステムの置き換え .....	61
3.5. スナップショットからのロールバック .....	63
3.6. スナップショットの保護.....	65
3.7. ZFS 自動スナップショット.....	68
4. バックアップ／リストア .....	72
4.1. フルバックアップ／リストア.....	72
4.1.1. バックアップ .....	73
4.1.2. リストア.....	75
4.2. 特定のファイルシステムのバックアップ／リストア .....	77
4.2.1. バックアップ .....	77
4.2.2. リストア.....	78
4.3. 増分バックアップ／リストア .....	79
4.3.1. 増分バックアップ .....	79
4.3.2. リストア.....	82
5. ミラーディスクの分割.....	86
6. ストレージプールの移行 .....	89
6.1. エクスポート(export) .....	89
6.2. インポート(import) .....	90
7. ZFS データの完全性のチェック.....	93
8. トラブルシューティング .....	94

---

8.1. ストレージプールの入出力統計情報 .....	94
8.2. ZFS 操作の監視.....	95
8.3. ZFS コマンドの履歴表示 .....	96
8.4. その他の問題や障害の識別・回復・解消 .....	96
改版履歴 .....	97

## 1. ストレージプールの構築

本書で構築するストレージプールの構成を、以下に示します。



本書は、以下の前提で解説しています。

- Solaris 11 がインストール済み
- SRU (Support Repository Update) が適用済み
- ESF (Enhanced Support Facility) などの必要パッケージがインストール済み

Oracle Solaris 11 のインストール、SRU の適用、および ESF のインストールなどの詳細については、以下を参照してください。

- 『Oracle Solaris 11 を使ってみよう(構築・運用手順書)』  
<https://www.fujitsu.com/jp/sparc/technical/document/solaris/#os>

### 1.1. ストレージプール作成のポイント

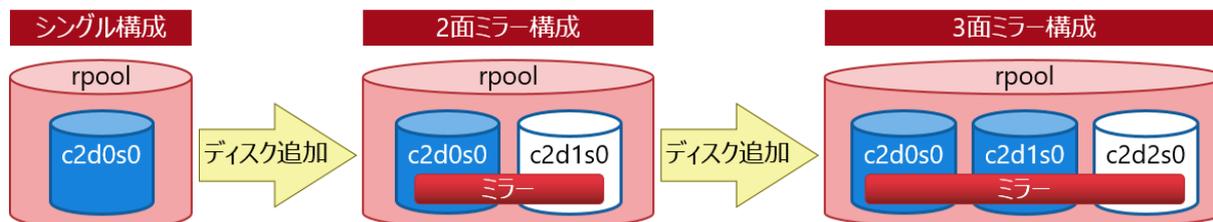
- Oracle Solaris 11 で構成可能なルートファイルシステム(/)は、ZFS のみとなります。UFS のルートファイルシステムは構成できません。
- ストレージプールの推奨サイズは 8 GB です。最小サイズは 64 MB です。推奨値の 8 GB 未満では空きプール領域の割り当てや再利用が難しくなるため、推奨値以上のサイズにしてください。
- ストレージプールに使用するデバイスのサイズは、128 MB 以上にしてください。
- ミラー構成を作成する場合は、複数のコントローラ配下のディスクで構成することを推奨します。
- ZFS のディスクラベルは、ルートファイルシステムも含め、EFI ラベルまたは SMI(VTOC)ラベルで作成可能です。

※ ルートファイルシステムの EFI ラベルについては、SPARC M10 のファームウェア XCP2230 以降の環境でサポートされます。ディスクラベルの説明を以下に示します。

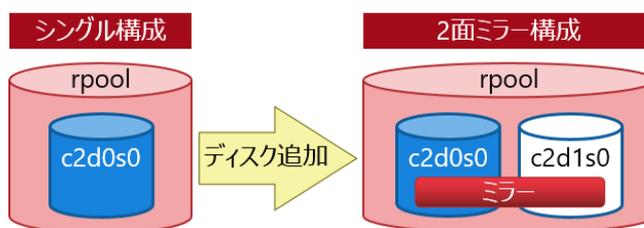
ディスクラベル	説明
SMI(VTOC)	2 TB に満たないディスク用 従来(UFS)のラベル

ディスクラベル	説明
EFI (Extensible Firmware Interface)	2 TB を超えるディスク用 Solaris 11 のルートファイルシステム含むストレージプール

## 1.2. ルートプール(システム領域)の構成変更



### 1.2.1. シングル構成から 2 面ミラー構成への変更



#### 1) ルートプールの状態を確認します。

エラーが発生していないことを確認してください。

```
# zpool status
pool: rpool
state: ONLINE
scan: none requested
config:

    NAME      STATE    READ WRITE CKSUM
    rpool     ONLINE   0     0     0
    c2d0s0    ONLINE   0     0     0

errors: No known data errors
```

#### 2) ルートプールのディスクのラベル情報を、追加するディスクにコピーします。

c2d0s0 のラベル情報を c2d1s0 へコピーします。

```
# prtvtoc /dev/rdisk/c2d0s0 | fmthard -s - /dev/rdisk/c2d1s0
fmthard: New volume table of contents now in place.
```

#### 《参考》EFI ラベル時のラベル情報のコピー

```
# prtvtoc /dev/rdisk/c2d0 | fmthard -s - /dev/rdisk/c2d0
```

▶ EFI ラベルの場合は、スライス番号が不要です。

## 3) 既存のディスク c2d0s0 に、新しいディスク c2d1s0 を接続します。

```
# zpool attach rpool c2d0s0 c2d1s0
Make sure to wait until resilver is done before rebooting.
```

- ▶ ディスクを追加すると、自動的に同期が開始され、ミラー構成になります。
- ▶ ルートプールの冗長構成は、ミラー構成のみ構成可能です。
- ▶ Oracle Solaris 11 では、適切なラベル付けとブートブロックが自動的に適用されます。
- ▶ メッセージ「SUNW-MSG-ID: ZFS-8000-QJ」が出力されることがありますが、問題はありません。

## 《参考》EFI ラベル時のミラー用ディスクの追加

```
# zpool attach rpool c2d0 c2d1
Make sure to wait until resilver is done before rebooting.
```

- ▶ EFI ラベルの場合は、スライス番号が付きません。

## 4) ミラー構成の同期の進行状況を確認します。

同期中は進行状況が%で表示され、STATE に「DEGRADED」と表示されます。

```
# zpool status
pool: rpool
state: DEGRADED
status: One or more devices are currently being resilvered.  The pool will
continue to function in a degraded state.
action: Wait for the resilver to complete.
Run 'zpool status -v' to see device specific details.
scan: resilver in progress since Fri Mar 1 15:35:05 2019
14.6G scanned
7.76G resilvered at 84.1M/s, 54.61% done, 1m20s to go
config:
NAME          STATE          READ WRITE CKSUM
rpool        DEGRADED          0    0    0
  mirror-0    DEGRADED          0    0    0
    c2d0s0    ONLINE           0    0    0
    c2d1s0    DEGRADED          0    0    0 (resilvering)
errors: No known data errors
```

- ▶ 同期中でも、ルートプールへの読み込み／書き込みのアクセスは可能です。

## 5) 同期が完了するまで定期的に zpool status コマンドを実行し、同期の進行状況を確認します。

同期が正常に完了すると、state および STATE に「ONLINE」が表示されます。

- 6) **state** および **STATE** に「**ONLINE**」が表示されていること、および **errors** にエラーが表示されていないことを確認します。また、**rpool** にディスク **c2d1s0** が追加され、2 面ミラー構成であることを確認します。

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

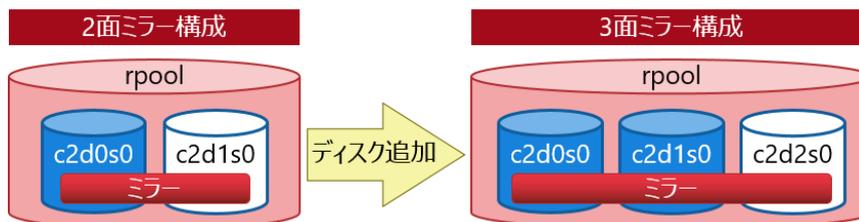
config:

NAME          STATE      READ WRITE CKSUM
rpool         ONLINE    0     0     0
  mirror-0    ONLINE    0     0     0
    c2d0s0    ONLINE    0     0     0
    c2d1s0    ONLINE    0     0     0

errors: No known data errors
```

- ▶ ZFS 上でのミラー構成は、「mirror-0」で表示されます。「mirror-0」の下に表示されているデバイスが、ミラー状態のディスクです。

### 1.2.2. 2 面ミラー構成から 3 面ミラー構成への変更



- 1) ルートプールの状態を確認します。

エラーが発生していないことを確認してください。

```
# zpool status
pool: rpool
state: ONLINE
scan: resilvered 12.6G in 2m39s with 0 errors on Fri Mar 1 15:33:09 2019

config:

NAME          STATE      READ WRITE CKSUM
rpool         ONLINE    0     0     0
  mirror-0    ONLINE    0     0     0
    c2d0s0    ONLINE    0     0     0
    c2d1s0    ONLINE    0     0     0

errors: No known data errors
```

## 2) ルートプールのディスクのラベル情報を、追加するディスクにコピーします。

c2d0s0 のラベル情報を c2d2s0 へコピーします。

```
# prtvtoc /dev/rdisk/c2d0s0 | fmthard -s - /dev/rdisk/c2d2s0
fmthard: New volume table of contents now in place.
```

## 3) 既存のディスク c2d0s0 に、新しいディスク c2d2s0 を接続します。

```
# zpool attach rpool c2d0s0 c2d2s0
Make sure to wait until resilver is done before rebooting.
```

## 4) 同期が完了するまで定期的に zpool status コマンドを実行し、同期の進行状況を確認します。

同期が正常に完了すると、state および STATE に「ONLINE」が表示されます。

## 5) state および STATE に「ONLINE」が表示されていること、および errors にエラーが表示されていないことを確認します。また、rpool にディスク c2d2s0 が追加され、3 面ミラー構成であることを確認します。

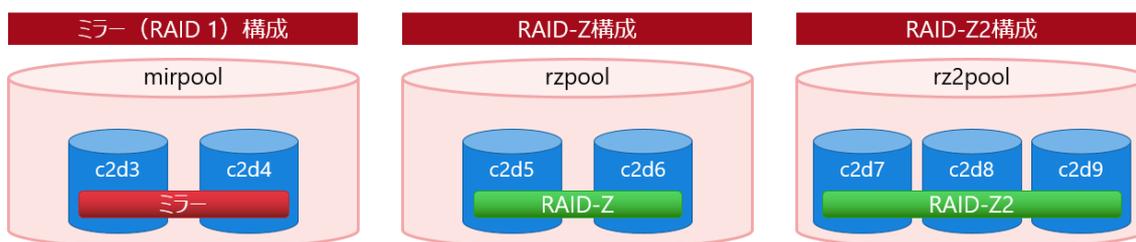
```
# zpool status
pool: rpool
state: ONLINE
scan: resilvered 12.6G in 2m49s with 0 errors on Fri Mar 1 15:37:54 2019

config:

    NAME      STATE    READ WRITE CKSUM
    rpool     ONLINE   0     0     0
      mirror-0 ONLINE   0     0     0
        c2d0s0 ONLINE   0     0     0
        c2d1s0 ONLINE   0     0     0
        c2d2s0 ONLINE   0     0     0

errors: No known data errors
```

### 1.3. ストレージプール(ユーザー領域)の作成



ストレージプールは、ディスク全体(例: c2t0d1)、または個別のスライス(例: c2t0d1s0)をストレージデバイスとして利用可能です。性能や管理の面から、ディスク全体を使用する方法を推奨します。

ディスク全体を使用する場合、ディスクのスライス設定は不要であり、ZFS によって EFI ラベルのディスクとしてフォーマットされます。

format コマンドで表示されるパーティションテーブルは次のようになります。例として、ストレージプールを構成するディスクのパーティションテーブル(EFI ラベル)を、以下に示します。

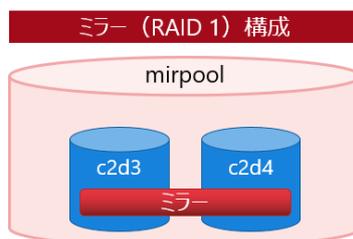
```
Current partition table (original):
Total disk sectors available: 104841149 + 16384 (reserved sectors)
```

Part	Tag	Flag	First Sector	Size	Last Sector
0	usr	wm	256	49.99GB	104841182
1	unassigned	wm	0	0	0
2	unassigned	wm	0	0	0
3	unassigned	wm	0	0	0
4	unassigned	wm	0	0	0
5	unassigned	wm	0	0	0
6	unassigned	wm	0	0	0
8	reserved	wm	104841183	8.00MB	104857566

ZFS データセット(ファイルシステム、ボリューム、スナップショット、クローンなどのストレージプール上での総称)に名前を付けるときの規則は、以下のとおりです。

- 空のデータセット名は許可されません。
- 各データセットに使用できる文字は、英数字および次の 5 つの特殊文字だけです。  
下線(\_)、ハイフン(-)、コロン(:)、ピリオド(.)、空白(" ")
- ストレージプール名の先頭は文字にする必要があります。ただし、次の制限事項があります。
  - C0~C9 で始まる名前は許可されません。例えば、「C3testpool」という名前は許可されません。
  - log, mirror, raidz, または spare で始まる名前は予約されているため、使用できません。
  - ストレージプール名には、パーセント記号(%)を含めないでください。
- データセット名の先頭は、英数字にする必要があります。
- データセット名には、パーセント記号(%)を含めないでください。

## 1) ミラー(RAID1)構成ストレージプールを作成します。



- i) ディスク c2d3 と c2d4 でミラーを構成するストレージプール「mirpool」を作成します。

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

- ▶ ストレージプールの作成時にマウントポイントを指定しない場合、ストレージプール名がデフォルトマウントポイントとなります。上記の場合、/mirpoolとしてマウントされます。

- ii) 作成したストレージプールを確認します。

```
# zpool list
NAME      SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
mirpool   29.8G  98.5K  29.7G   0%  1.00x  ONLINE  -
rpool     49.8G  14.6G  35.2G  29%  1.00x  ONLINE  -
```

zpool list コマンドで表示されるプロパティを以下に示します。

プロパティ	説明
NAME	ストレージプールの名前
SIZE	ストレージプールの合計サイズ 最上位レベルにあるすべての仮想デバイスの合計サイズ
ALLOC	すべてのデータセットおよび内部メタデータから割り当てられた容量 ※ この容量は、df コマンドなどのファイルシステムレベルで報告される容量とは一致しない
FREE	ストレージプール内で割り当てられていない容量
CAP	使用されている容量の割合 ※ 総容量に対する割合(パーセント)で表現される
DEDUP	重複除外された割合
HEALTH	ストレージプールの現在の状態
ALTROOT	ストレージプールの代替ルート

iii) 作成したストレージプールの状態を確認します。

```
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: none requested
config:

    NAME      STATE      READ WRITE CKSUM
    mirpool   ONLINE    0     0     0
    mirror-0  ONLINE    0     0     0
    c2d3      ONLINE    0     0     0
    c2d4      ONLINE    0     0     0

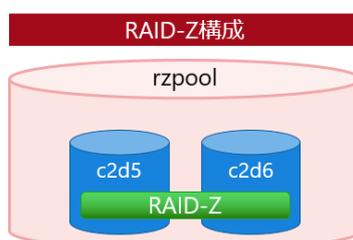
errors: No known data errors
```

- ▶ `-x` オプションを使用すると、エラーが発生しているか、または使用不可能なストレージプールの状態だけを表示します。
- ▶ `-v` オプションを使用すると詳細なデータエラー情報が表示され、前回のスクラブ完了後の、ストレージプールのすべてのデータエラーが出力されます。

`zpool status` コマンドで表示される状態を、以下に示します。

状態	説明
DEGRADED	仮想デバイスに障害が発生しているが、引き続き利用できる状態 例: ミラーデバイスまたは RAID-Z デバイスを構成するデバイスのうち、1 つ以上のデバイスが失われた
OFFLINE	管理者が仮想デバイスを明示的にオフラインにしている状態
ONLINE	デバイスが正常に機能している状態
REMOVED	システム稼働中にデバイスが物理的に取り外されている状態
UNAVAIL	デバイスまたは仮想デバイスを開くことができない状態 最上位レベルの仮想デバイスが使用できない場合は、最上位レベルの仮想デバイスの、ストレージプールのデバイスへのアクセスは不可

2) RAID-Z 構成のストレージプールを作成します。



i) ディスク `c2d5` と `c2d6` で構成される RAID-Z 構成のストレージプールを、「`rzpool`」という名前で作成します。

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

- ii) 作成したストレージプールを確認します。

```
# zpool list
NAME      SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
mirpool   29.8G  98.5K  29.7G   0%  1.00x  ONLINE  -
rpool     49.8G  14.6G  35.2G  29%  1.00x  ONLINE  -
rzpool   59.5G  204K   59.5G   0%  1.00x  ONLINE  -
```

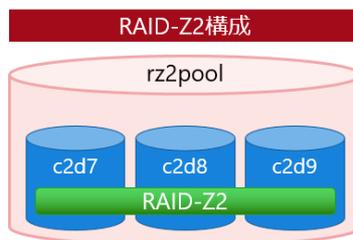
- iii) 作成したストレージプールの状態を確認します。

```
# zpool status rzpool
pool: rzpool
state: ONLINE
scan: none requested
config:

   NAME      STATE      READ WRITE CKSUM
   rzpool    ONLINE      0     0     0
   raidz1-0  ONLINE      0     0     0
   c2d5     ONLINE      0     0     0
   c2d6     ONLINE      0     0     0

errors: No known data errors
```

- 3) RAID-Z2 構成のストレージプールを作成します。



- i) ディスク c2d7、c2d8、および c2d9 で構成される RAID-Z2 構成のストレージプールを、「rz2pool」という名前で作成します。

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

- ii) 作成したストレージプールを確認します。

```
# zpool list
NAME      SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
mirpool   29.8G  98.5K  29.7G   0%  1.00x  ONLINE  -
rpool     49.8G  14.6G  35.2G  29%  1.00x  ONLINE  -
rz2pool  89.5G  306K   89.5G   0%  1.00x  ONLINE  -
rzpool    59.5G  150K   59.5G   0%  1.00x  ONLINE  -
```

iii) 作成したストレージプールの状態を確認します。

```
# zpool status rz2pool
pool: rz2pool
state: ONLINE
scan: none requested
config:

    NAME            STATE             READ WRITE CKSUM
    rz2pool          ONLINE            0     0     0
    raidz2-0         ONLINE            0     0     0
    c2d7              ONLINE            0     0     0
    c2d8              ONLINE            0     0     0
    c2d9              ONLINE            0     0     0
errors: No known data errors
```

#### 《参考》 マウントポイントを指定したストレージプールの作成

ここでは例として、マウントポイントに「/export/test」を指定し、ミラー構成の「upool\_1」というストレージプールをディスク c2d13 と c2d14 で作成します。

```
# zpool create -m /export/test upool_1 mirror c2d13 c2d14
```

## 《参考》EFI ラベルを SMI ラベルに戻す方法(例: デバイス名 c2d3)

-e オプションを使用します。

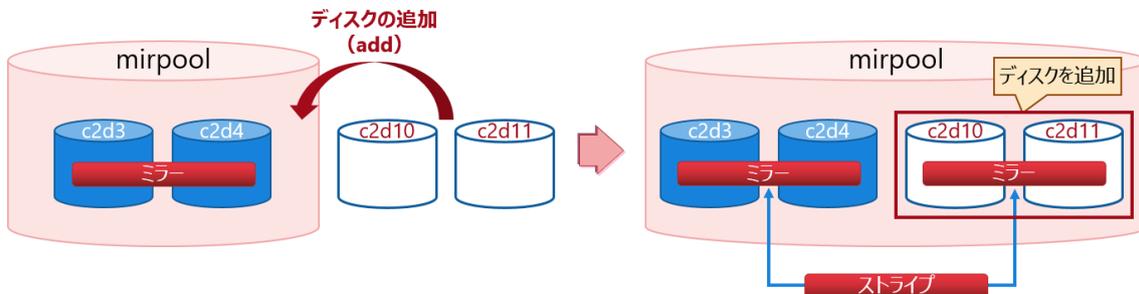
```
# format -e c2d3
selecting c2d3

FORMAT MENU:
  disk          - select a disk
  type          - select (define) a disk type
  partition     - select (define) a partition table
  current       - describe the current disk
  format        - format and analyze the disk
  repair        - repair a defective sector
  show          - translate a disk address
  label         - write label to the disk
  analyze       - surface analysis
  defect        - defect list management
  backup        - search for backup labels
  verify        - read and display labels
  inquiry       - show disk ID
  volname       - set 8-character volume name
  !<cmd>       - execute <cmd>, then return
  quit

format> label ←「label」を入力します。
[0] SMI Label
[1] EFI Label
Specify Label type[1]: 0 ←「0」を入力します。
format> quit ←「quit」または「q」を入力します。
```

## 1.4. デバイスの追加(add)

ストレージプールにデバイスを追加することで、ストレージプールの領域を動的に拡張できます。ストレージプール内のすべてのデータセットは、拡張した領域をすぐに利用できます。



### 1) ディスクを追加する前に、ストレージプール「mirpool」の構成を確認します。

```
# zpool list mirpool
NAME      SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
mirpool   29.8G  1.57G  28.2G  5%   1.00x  ONLINE  -
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: scrub repaired 0 in 19s with 0 errors on Wed Mar 13 16:56:36 2019

config:

    NAME      STATE      READ WRITE CKSUM
    mirpool   ONLINE    0     0     0
    mirror-0  ONLINE    0     0     0
    c2d3      ONLINE    0     0     0
    c2d4      ONLINE    0     0     0

errors: No known data errors
```

▶ 「mirpool」は 30 GB のディスク 2 本からなるミラー構成です。

### 2) ディスク c2d10 と c2d11 をストレージプール「mirpool」に追加し、ミラー構成にします。

ストレージプール「mirpool」に 30 GB のディスク 2 本をミラー構成で追加します。

```
# zpool add mirpool mirror c2d10 c2d11
```

## 3) ストレージプール「mirpool」の構成を確認します。

```
# zpool list mirpool
NAME      SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
mirpool   59.5G 1.57G 57.9G  2%  1.00x  ONLINE  -
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: scrub repaired 0 in 19s with 0 errors on Wed Mar 13 16:56:36 2019

config:

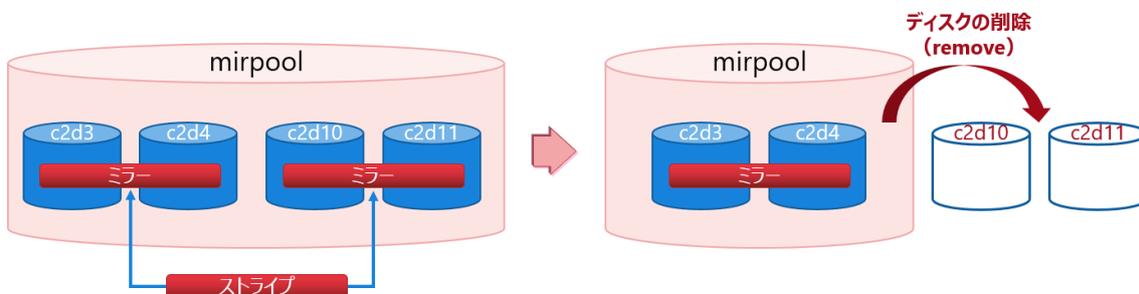
      NAME      STATE      READ WRITE CKSUM
mirpool  ONLINE      0     0     0
  mirror-0  ONLINE      0     0     0
    c2d3     ONLINE      0     0     0
    c2d4     ONLINE      0     0     0
  mirror-1  ONLINE      0     0     0
    c2d10    ONLINE      0     0     0
    c2d11    ONLINE      0     0     0

errors: No known data errors
```

- ▶ 30 GB のディスク 2 本をミラー構成で追加したため、SIZE の値は約 60 GB です。
- ▶ 追加したディスクは、ストライピングになります。RAID-Z などの拡張はできません。  
例えば、2+1 構成の RAID-Z にディスクを 1 本追加した場合、3+1 構成になるのではなく、2+1 構成の RAID-Z とディスク 1 本のストライピングになります。
- ▶ 追加するディスクは、元のストレージプールと同一容量のディスク、および同一の RAID 構成を推奨します。

## 1.5. デバイスの削除(remove)

ストレージプールからデバイスを削除するには、`zpool remove` コマンドを使用します。「[1.4. デバイスの追加\(add\)](#)」で追加した `mirror-1` などの ID を参照し、デバイスを削除できます。



### 1) デバイスを削除する前のストレージプール「mirpool」の構成を確認します。

ミラー構成「`mirror-0`」と「`mirror-1`」は、2つのミラー化されたデバイスで構成されています。

```
# zpool list mirpool
NAME      SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
mirpool   59.5G  1.57G  57.9G   2%  1.00x  ONLINE  -
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: scrub repaired 0 in 19s with 0 errors on Wed Mar 13 16:56:36 2019

config:

      NAME      STATE      READ WRITE CKSUM
mirpool  ONLINE      0     0     0
  mirror-0  ONLINE      0     0     0
    c2d3    ONLINE      0     0     0
    c2d4    ONLINE      0     0     0
  mirror-1  ONLINE      0     0     0
    c2d10   ONLINE      0     0     0
    c2d11   ONLINE      0     0     0

errors: No known data errors
```

### 2) ストレージプール「mirpool」から、ミラー構成「mirror-1」を削除します。

```
# zpool remove mirpool mirror-1
```

- ▶ 「cannot remove device(s): operation not supported on this type of pool (操作はこのタイプのプールでサポートされていません)」というメッセージが表示され、`zpool remove` コマンドが失敗する場合は、「[《参考》 zpool remove コマンドが失敗する場合](#)」を実施してください。

### 3) ストレージプール「mirpool」の構成を確認します。

30 GB のディスク 2 本のミラー構成を削除したため、SIZE の値が約 30 GB であることを確認します。

```
# zpool list mirpool
NAME      SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
mirpool   29.8G  1.57G  28.2G  5%   1.00x  ONLINE  -
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: resilvered 1K in 1s with 0 errors on Mon Mar 18 14:37:33 2019

config:

      NAME                                STATE      READ WRITE CKSUM
      mirpool                              ONLINE         0   0   0
        mirror-0                            ONLINE         0   0   0
          c2d3                              ONLINE         0   0   0
          c2d4                              ONLINE         0   0   0

errors: No known data errors
```

**《参考》 zpool remove コマンドが失敗する場合**

zpool detach コマンドでミラー構成「mirror-1」のディスク 1 本を解除後に、zpool remove コマンドを実施します。

(1) デバイスを削除する前のストレージプール「mirpool」の構成を確認します。

ミラー構成「mirpool-0」と「mirpool-1」は、2 つのミラー化されたデバイスで構成されています。

```
# zpool list mirpool
NAME      SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
mirpool   59.5G  1.57G  57.9G  2%   1.00x  ONLINE  -
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: scrub repaired 0 in 1s with 0 errors on Fri May 17 21:57:17 2019

config:

      NAME      STATE      READ WRITE CKSUM
mirpool  ONLINE      0     0     0
  mirror-0  ONLINE      0     0     0
    c2d3     ONLINE      0     0     0
    c2d4     ONLINE      0     0     0
  mirror-1  ONLINE      0     0     0
    c2d10    ONLINE      0     0     0
    c2d11    ONLINE      0     0     0

errors: No known data errors
```

(2) ストレージプール「mirpool」のミラー構成「mirror-1」から、ディスク c2d10 を切り離します。

```
# zpool detach mirpool c2d10
```

(3) 切り離し後のストレージプール「mirpool」の状態を確認します。

```
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: scrub repaired 0 in 1s with 0 errors on Fri May 17 21:57:17 2019

config:

    NAME          STATE          READ WRITE CKSUM
    mirpool       ONLINE         0     0     0
      mirror-0    ONLINE         0     0     0
        c2d3      ONLINE         0     0     0
        c2d4      ONLINE         0     0     0
        c2d11     ONLINE         0     0     0

errors: No known data errors
```

(4) ストレージプール「mirpool」から、ディスク c2d11 を削除します。

```
# zpool remove mirpool c2d11
```

(5) ストレージプール「mirpool」の構成を確認します。

30 GB のディスク 2 本のミラー構成を削除したため、SIZE の値が約 30 GB であることを確認します。

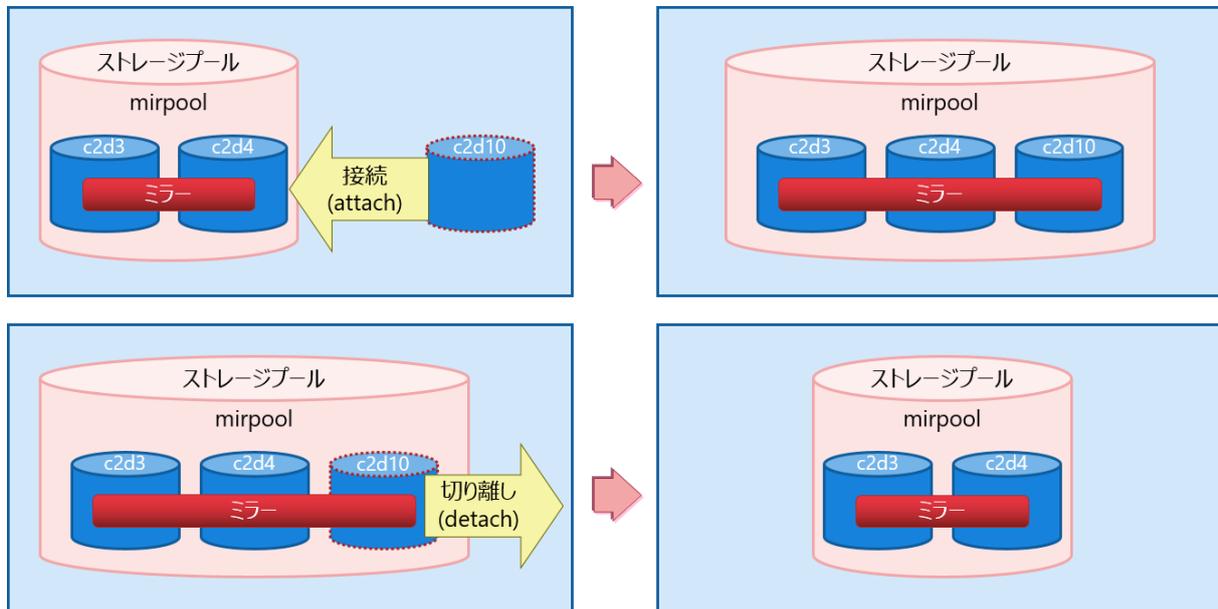
```
# zpool list mirpool
NAME          SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
mirpool       29.8G  1.57G  28.2G  5%   1.00x  ONLINE  -
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: resilvered 512 in 2s with 0 errors on Thu May 23 15:53:49 2019

config:

    NAME          STATE          READ WRITE CKSUM
    mirpool       ONLINE         0     0     0
      mirror-0    ONLINE         0     0     0
        c2d3      ONLINE         0     0     0
        c2d4      ONLINE         0     0     0

errors: No known data errors
```

## 1.6. デバイスの接続／切り離し(attach/detach)



### 1.6.1. デバイスの接続(attach)

- 1) ストレージプール「mirpool」の状態を確認します。

```
# zpool list mirpool
NAME      SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
mirpool  29.8G  1.57G  28.2G   5%  1.00x  ONLINE  -
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: resilvered 1K in 1s with 0 errors on Mon Mar 18 14:37:33 2019

config:

    NAME                STATE      READ  WRITE CKSUM
    mirpool              ONLINE    0    0    0
    mirror-0            ONLINE    0    0    0
    c2d3                 ONLINE    0    0    0
    c2d4                 ONLINE    0    0    0

errors: No known data errors
```

- 2) ストレージプール「mirpool」のディスク c2d3 に、ディスク c2d10 を接続します。

```
# zpool attach mirpool c2d3 c2d10
```

### 3) ミラー構成の同期の進行状況を確認します。

同期中は進行状況が%で表示され、STATE に「DEGRADED」と表示されます。

```
# zpool status mirpool
pool: mirpool
state: DEGRADED
status: One or more devices are currently being resilvered.  The pool will
       continue to function in a degraded state.
action: Wait for the resilver to complete.
       Run 'zpool status -v' to see device specific details.
scan: resilver in progress since Mon Mar 18 15:13:28 2019
      1.57G scanned
      410M resilvered at 19.0M/s, 25.13% done, 1m03s to go
config:

      NAME                                STATE      READ WRITE CKSUM
mirpool
  mirror-0
    c2d3                                ONLINE    0     0     0
    c2d4                                ONLINE    0     0     0
    c2d10                               DEGRADED  0     0     0 (resilvering)

errors: No known data errors
```

### 4) 同期が完了するまで定期的に zpool status コマンドを実行し、同期の進行状況を確認します。

同期が正常に完了すると、state および STATE に「ONLINE」が表示されます。

- 5) **state** および **STATE** に「**ONLINE**」が表示されていること、および **errors** にエラーが表示されていないことを確認します。また、「**mirpool**」のディスク **c2d3** にディスク **c2d10** が接続され、3 面ミラー構成であることを確認します。

```
# zpool list mirpool
NAME      SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
mirpool   29.8G  1.57G  28.2G  5%   1.00x  ONLINE  -
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: resilvered 1.57G in 1m31s with 0 errors on Mon Mar 18 15:14:59 2019

config:

    NAME                STATE          READ  WRITE CKSUM
    mirpool              ONLINE         0     0     0
      mirror-0           ONLINE         0     0     0
        c2d3             ONLINE         0     0     0
        c2d4             ONLINE         0     0     0
        c2d10            ONLINE         0     0     0

errors: No known data errors
```

- ▶ ミラー構成のため、ディスクを追加しても **SIZE** の値は変わりません。

#### 《参考》 デバイスを追加してストレージプールをミラー構成にする場合

ここでは例として、単一のディスク **c2d3** で構成されているストレージプールにディスク **c2d4** を **attach** し、ミラー構成に変更する手順を示します。

- (1) ストレージプール「**mirpool**」の状態を確認します。

```
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: none requested

config:

    NAME                STATE          READ  WRITE CKSUM
    mirpool              ONLINE         0     0     0
      c2d3               ONLINE         0     0     0

errors: No known data errors
```

- (2) ストレージプール「**mirpool**」のディスク **c2d3** に、ディスク **c2d4** を接続します。

```
# zpool attach mirpool c2d3 c2d4
```

- ▶ ディスクを追加すると自動的に同期が開始され、ミラー構成になります。

- (3) ミラー構成の同期の進行状況を確認します。同期が完了するまで定期的に `zpool status` コマンドを実行し、同期の進行状況を確認します。

```
# zpool status mirpool
pool: mirpool
state: DEGRADED
status: One or more devices are currently being resilvered. The pool will
continue to function in a degraded state.
action: Wait for the resilver to complete.
Run 'zpool status -v' to see device specific details.
scan: resilver in progress since Mon Mar 18 15:32:17 2019
1.57G scanned
1.06G resilvered at 27.4M/s, 59.82% done, 23s to go
config:

NAME                STATE      READ WRITE CKSUM
mirpool              DEGRADED    0    0    0
  mirror-0           DEGRADED    0    0    0
    c2d3              ONLINE     0    0    0
    c2d4              DEGRADED    0    0    0 (resilvering)

errors: No known data errors
```

同期が正常に完了すると、state および STATE に「ONLINE」が表示されます。

- (4) state および STATE に「ONLINE」が表示されていること、および errors にエラーが表示されていないことを確認します。また、「mirpool」のディスク c2d3 にディスク c2d4 が接続され、2 面ミラー構成であることを確認します。

```
# zpool list mirpool
NAME      SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
mirpool  29.8G  1.57G  28.2G  5%  1.00x  ONLINE  -
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: resilvered 1.57G in 1m05s with 0 errors on Mon Mar 18 15:33:22 2019
config:

NAME                STATE      READ WRITE CKSUM
mirpool              ONLINE    0    0    0
  mirror-0           ONLINE    0    0    0
    c2d3              ONLINE     0    0    0
    c2d4              ONLINE    0    0    0

errors: No known data errors
```

## 1.6.2. デバイスの切り離し(detach)

### 1) ストレージプール「mirpool」の状態を確認します。

3面ミラー構成であることを確認します。

```
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: resilvered 1.57G in 1m08s with 0 errors on Mon Mar 18 15:45:27 2019

config:

    NAME                STATE          READ WRITE CKSUM
    mirpool              ONLINE         0     0     0
      mirror-0
        c2d3             ONLINE         0     0     0
        c2d4             ONLINE         0     0     0
        c2d10            ONLINE         0     0     0

errors: No known data errors
```

### 2) ストレージプール「mirpool」のディスク c2d10 を切り離します。

```
# zpool detach mirpool c2d10
```

### 3) detach 後のストレージプール「mirpool」の状態を確認します。

切り離れたディスク c2d10 が表示されないことを確認します。

```
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: resilvered 1.57G in 1m08s with 0 errors on Mon Mar 18 15:45:27 2019

config:

    NAME                STATE          READ WRITE CKSUM
    mirpool              ONLINE         0     0     0
      mirror-0
        c2d3             ONLINE         0     0     0
        c2d4             ONLINE         0     0     0

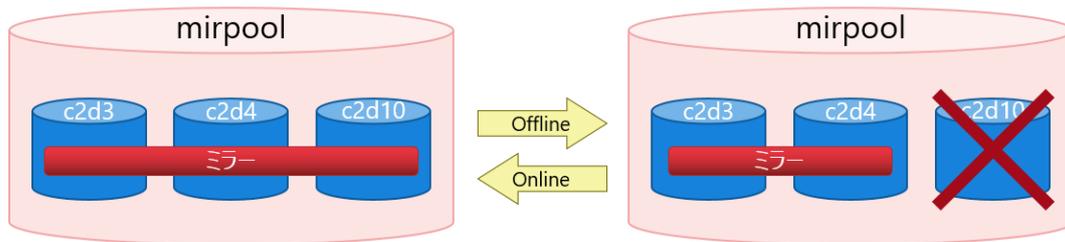
errors: No known data errors
```

## 1.7. デバイスのオンライン/オフライン (online/offline)

ZFS では、デバイスを個別にオフラインまたはオンラインにできます。オフラインにしたデバイスには、I/O 要求は送信されません。

ハードウェアに障害が発生している場合、ZFS ではその状態を一時的な状態と見なして、デバイスへのデータの読み取り／書き込みを続行します。

継続的に障害が発生している場合は、デバイスをオフラインにして無視するように設定できます。また、オンラインに戻したあとは自動で再同期化されます。



デバイスをオフラインにするときは、以下の点に注意してください。

- ストレージプールを構成するすべてのデバイスをオフラインにすることはできません。
- デフォルト設定では、デバイスをオフラインにすると、システムを再起動してもオフラインのままです。一時的にオフラインにするには、`-t` オプションを使用します。一時的にオフラインにした場合、システムを再起動すると自動的にオンライン状態に戻ります。
- デバイスは、オフラインになってもストレージプールから切り離されません。元のストレージプールが破棄されても、オフラインのデバイスを別のストレージプールで使用しようとすると、エラーが表示されて使用できません。元のストレージプールを破棄したあとで、オフラインのデバイスを別のストレージプールで使用する場合は、まずデバイスをオンラインに戻してから、元のストレージプールを破棄します。

### 1.7.1. デバイスのオフライン(offline)

#### Point

「[1.6.2. デバイスの切り離し\(detach\)](#)」でディスク c2d10 を切り離している場合は、以下のコマンドを実行し、ディスク c2d10 を接続してください。

```
# zpool attach mirpool c2d3 c2d10
```

#### 1) ストレージプール「mirpool」の状態を確認します。

```
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: resilvered 1.57G in 1m07s with 0 errors on Mon Mar 18 16:02:30 2019

config:

    NAME                STATE          READ WRITE CKSUM
    mirpool              ONLINE        0     0     0
      mirror-0          ONLINE        0     0     0
        c2d3             ONLINE        0     0     0
        c2d4             ONLINE        0     0     0
        c2d10            ONLINE        0     0     0

errors: No known data errors
```

#### 2) ストレージプール「mirpool」のディスク c2d10 をオフラインにします。

```
# zpool offline mirpool c2d10
```

### 3) オフライン後に、ストレージプールの状態を確認します。

対象のディスクが「OFFLINE」と表示されること、およびストレージプールの STATE が「DEGRADED」と表示されることを確認します。

```
# zpool status mirpool
pool: mirpool
state: DEGRADED
status: One or more devices has been taken offline by the administrator.
        Sufficient replicas exist for the pool to continue functioning in a
        degraded state.
action: Online the device using 'zpool online' or replace the device with
        'zpool replace'.
scan: resilvered 1.57G in 1m07s with 0 errors on Mon Mar 18 16:02:30 2019

config:

    NAME                                STATE      READ WRITE CKSUM
    mirpool                              DEGRADED  0     0     0
      mirror-0                            DEGRADED  0     0     0
        c2d3                              ONLINE    0     0     0
        c2d4                              ONLINE    0     0     0
        c2d10                             OFFLINE   0     0     0

errors: No known data errors
```

#### 《参考》 デバイスを一時的にオフラインにする

-t オプションを使用すると、デバイスは一時的にオフラインになります。

ここでは例として、ディスク c2d10 を一時的にオフラインにするコマンド実行例を示します。

```
# zpool offline -t mirpool c2d10
```

- ▶ 一時的にオフラインにした場合は、システムを再起動すると自動的にオンライン状態に戻ります。

### 1.7.2. デバイスのオンライン(online)

オフラインのデバイスは、zpool online コマンドでオンラインにできます。デバイスがオンラインになると、ストレージプールに書き込まれたすべてのデータは、新しく使用可能になったデバイスに再同期化されます。

#### 1) デバイスの状態を確認します。

```
# zpool status mirpool
pool: mirpool
state: DEGRADED
status: One or more devices has been taken offline by the administrator.
        Sufficient replicas exist for the pool to continue functioning in a
        degraded state.
action: Online the device using 'zpool online' or replace the device with
        'zpool replace'.
scan: resilvered 1.57G in 1m07s with 0 errors on Mon Mar 18 16:02:30 2019

config:

    NAME                                STATE      READ WRITE CKSUM
    mirpool                              DEGRADED   0     0     0
      mirror-0                            DEGRADED   0     0     0
        c2d3                              ONLINE     0     0     0
        c2d4                              ONLINE     0     0     0
        c2d10                             OFFLINE    0     0     0

errors: No known data errors
```

#### 2) ストレージプール「mirpool」のディスク c2d10 をオンラインにします。

```
# zpool online mirpool c2d10
```

- ▶ 「warning: device 'c2d10' onlined, but remains in degraded state」というメッセージが出力される場合があります。再同期化が完了するまで、デバイスが低下した状態のままであることを示しています。

### 3) オンライン後に、デバイスの状態を確認します。

```
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: resilvered 21K in 1s with 0 errors on Mon Mar 18 16:09:13 2019

config:

    NAME                STATE          READ WRITE CKSUM
    mirpool              ONLINE         0     0     0
      mirror-0
        c2d3            ONLINE         0     0     0
        c2d4            ONLINE         0     0     0
        c2d10           ONLINE         0     0     0

errors: No known data errors
```

## 1.8. エラーのクリア (clear)

ストレージプール内のデバイスに障害が発生した場合、zpool status の出力にエラーが表示されます。デバイスのエラーが一時的な場合は、エラーを無視してクリアすることができます。クリア後もデバイスのエラーが出力される場合は、デバイスの交換が必要です。エラー出力は自動的に消えないため、デバイスの交換後は手動でエラー出力をクリアします。

### 1) ストレージプール「mirpool」のエラーをクリアします。

```
# zpool clear mirpool
```

#### 《参考》 デバイスを指定してエラーのクリアを実行する

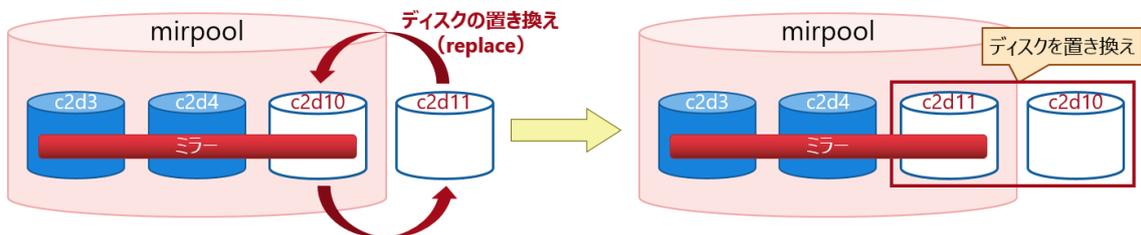
デバイスを指定してクリアを実行します。

```
# zpool clear mirpool c2d10
```

- ▶ 指定したデバイスに関連付けられたエラーのみ、クリアされます。

## 1.9. デバイスの置き換え(replace)

zpool replace コマンドで、ストレージプール内のデバイスを物理的に置き換えることができます。



ストレージプール内のデバイスを置き換える場合は、次のことを考慮してください。

- ストレージプールのプロパティ autoreplace を on に設定した場合、ストレージプールに以前属していたデバイスと物理的に同じ位置に新しいデバイスが検出されると、検出されたデバイスは自動的にフォーマットされ、zpool replace コマンドを使用せずに置き換えられます。ハードウェアの種類によっては、この機能を使用できない場合があります。
- 交換用デバイスは、ストレージプール内で最も容量の小さいディスクよりも大きい容量である必要があります。
- 置き換える前のデバイスよりも容量が大きいデバイスをストレージプールに追加しても、自動的に最大容量まで拡張されません。ストレージプールに置き換えたデバイスを最大容量まで拡張する場合は、ストレージプールの autoexpand プロパティを有効にします。デフォルトでは、autoexpand プロパティは無効です。
- 障害の発生したディスクがホットスペアに自動的に置き換えられた場合は、障害の発生したディスクが置き換えられたあと、スペアの切り離しが必要になることがあります。
- デバイスの置き換えでは、新しいデバイスへデータの再同期が行われるため、時間がかかります。zpool scrub コマンドを実行すると、新しいデバイスが動作可能なこと、およびデータが正しく書き込まれることを確認できます。

## 1) デバイスを確認します。

```
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: resilvered 0 in 1s with 0 errors on Mon Mar 18 16:21:03 2019

config:

    NAME                STATE          READ WRITE CKSUM
    mirpool              ONLINE         0     0     0
      mirror-0
        c2d3             ONLINE         0     0     0
        c2d4             ONLINE         0     0     0
        c2d10            ONLINE         0     0     0

errors: No known data errors
```

## 2) ディスク c2d3、c2d4、および c2d10 で構成されるストレージプール「mirpool」のうち、ディスク c2d10 を別ディスク c2d11 で置き換えます。

```
# zpool replace mirpool c2d10 c2d11
```

- ▶ 1つのデバイスで構成されるストレージプール内のデバイスを置き換える場合も同様に、置き換える前後のデバイスを指定します。

## 3) デバイスの置き換え後、ストレージプールの状態を確認し、デバイスが置き換えられたことを確認します。

同期処理が完了するまでは、「replacing-2」と表示されます。

```
# zpool status mirpool
pool: mirpool
state: DEGRADED
status: One or more devices are currently being resilvered.  The pool will
        continue to function in a degraded state.
action: Wait for the resilver to complete.
        Run 'zpool status -v' to see device specific details.
scan: resilver in progress since Mon Mar 18 16:43:57 2019
      1.57G scanned
      375M resilvered at 23.1M/s, 20.58% done, 55s to go
config:

      NAME                                STATE      READ WRITE CKSUM
mirpool                                DEGRADED    0     0     0
  mirror-0                              DEGRADED    0     0     0
    c2d3                                 ONLINE     0     0     0
    c2d4                                 ONLINE     0     0     0
    replacing-2                          DEGRADED    0     0     0
    c2d10                                ONLINE     0     0     0
    c2d11                                 DEGRADED    0     0     0 (resilvering)

errors: No known data errors
```

同期処理が完了すると、「replacing-2」の表示が消えます。

```
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: resilvered 1.57G in 1m19s with 0 errors on Mon Mar 18 16:45:16 2019
config:

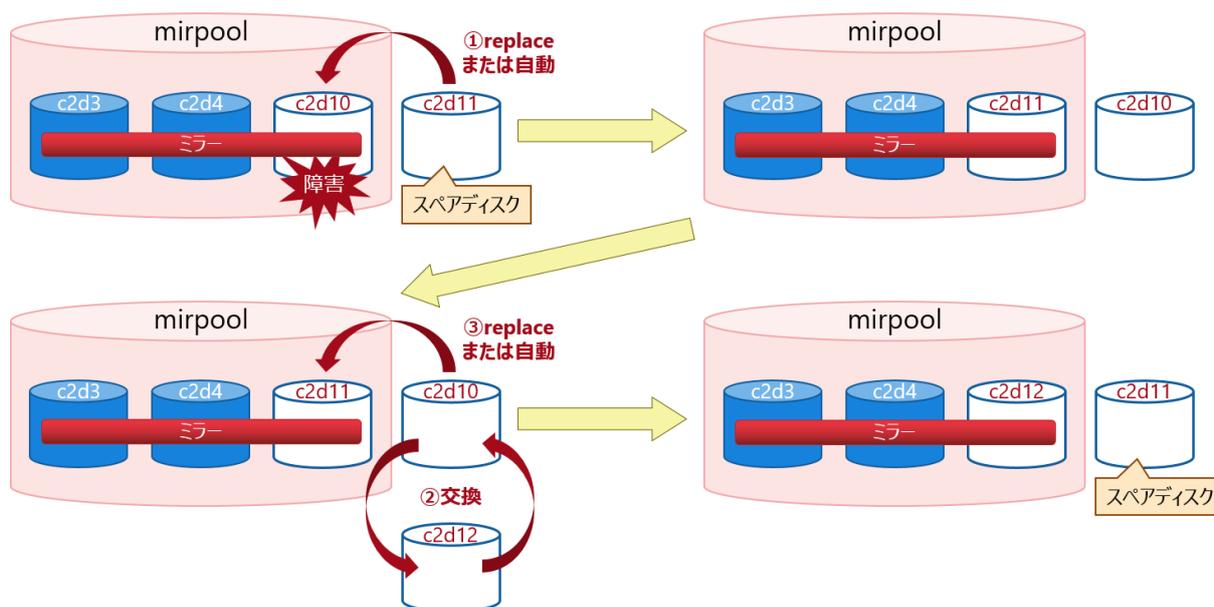
      NAME                                STATE      READ WRITE CKSUM
mirpool                                ONLINE     0     0     0
  mirror-0                              ONLINE     0     0     0
    c2d3                                 ONLINE     0     0     0
    c2d4                                 ONLINE     0     0     0
    c2d11                                 ONLINE     0     0     0

errors: No known data errors
```

## 1.10. ホットスペア (spare)

ホットスペア機能を利用すると、ストレージプールで障害が発生した場合、故障したデバイスをスペアディスクと置き換えることができます。オンラインのデバイスが故障した場合は、スペアディスクに自動的に置き換えられます。

ストレージプールの autoreplace プロパティを on に設定している場合、新しいデバイスが挿入されてオンライン処理が完了すると、使用中のスペアディスクは自動的に切り離され、未使用のスペアディスクに戻ります。



### 《注意》ホットスペア機能使用時の注意

- スペアディスクには、ストレージプール内で最も容量が大きいディスク以上の容量のディスクを設定してください。容量の小さなディスクを設定した場合、スペアディスクがアクティブになる（故障したデバイスと置き換わる、または手で置き換える）際にエラーとなります。
- 複数のストレージプールでホットスペア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。

### 1.10.1. ホットスペアディスクの登録

#### Point

「[1.9. デバイスの置き換え\(replace\)](#)」でディスク c2d10 を別ディスク c2d11 で置き換えている場合は、以下のコマンドを実行し、ディスク c2d10 に再度置き換えてください。

```
# zpool replace mirpool c2d11 c2d10
```

#### 1) スペアディスクを指定します。

- ストレージプール作成時にスペアディスクを指定する場合

```
# zpool create mirpool mirror c2d3 c2d4 c2d10 spare c2d11 c2d12
```

- ストレージプール作成後にスペアディスクを指定する場合

```
# zpool add mirpool spare c2d11 c2d12
```

#### 2) スペアディスクを確認します。

```
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: none requested
config:
```

NAME	STATE	READ	WRITE	CKSUM
mirpool	ONLINE	0	0	0
mirror-0	ONLINE	0	0	0
c2d3	ONLINE	0	0	0
c2d4	ONLINE	0	0	0
c2d10	ONLINE	0	0	0
spares				
c2d11	AVAIL			
c2d12	AVAIL			

```
errors: No known data errors
```

### 1.10.2. ホットスペアディスクのアクティブ化

スペアディスクをアクティブにする(置き換える)方法は、以下の2つです。

- 手動: zpool replace コマンドで置き換えます。
- 自動: FMA エージェントがデバイスのエラーを検知し、故障デバイスをスペアディスクに置き換えます。
  - ▶ FMA(Fault Management Architecture)は、ハードウェアに関する情報を監視・解析し、デバイスの自動切り離しなどを行う可用性を高める OS 標準機能です。

#### 1) ストレージプール内のホットスペアを、手動でアクティブにします。

```
# zpool replace mirpool c2d10 c2d11
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: resilvered 77.5K in 2s with 0 errors on Mon Mar 18 18:31:12 2019

config:

    NAME          STATE      READ WRITE CKSUM
    mirpool       ONLINE    0     0     0
      mirror-0    ONLINE    0     0     0
        c2d3      ONLINE    0     0     0
        c2d4      ONLINE    0     0     0
        spare-2   ONLINE    0     0     0
          c2d10   ONLINE    0     0     0
          c2d11   ONLINE    0     0     0
    spares
      c2d11       INUSE
      c2d12       AVAIL

errors: No known data errors
```

- ▶ アクティブになっているホットスペアのディスクは、STATE に「INUSE」が表示されます。

### 1.10.3. ホットスペアディスクの非アクティブ化

以下の 3 つの方法で、スペアディスクを非アクティブにできます。

- スペアディスクをストレージプールから削除することでホットスペアを取り消す (remove)
- 元のデバイスをスペアディスクと置き換える (replace)
- スペアディスクを恒久的に交換する (元のデバイスを detach する)

ここでは、ホットスペアに設定しているスペアディスクを削除する場合のコマンド実行例を、以下に示します。

```
# zpool remove mirpool c2d12
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: resilvered 77.5K in 2s with 0 errors on Mon Mar 18 18:31:12 2019

config:

    NAME          STATE      READ WRITE CKSUM
    mirpool       ONLINE    0     0     0
      mirror-0    ONLINE    0     0     0
        c2d3      ONLINE    0     0     0
        c2d4      ONLINE    0     0     0
        spare-2   ONLINE    0     0     0
        c2d10     ONLINE    0     0     0
        c2d11     ONLINE    0     0     0
    spares
      c2d11       INUSE

errors: No known data errors
```

## 《参考》 アクティブになっているスペアディスクを削除した場合

アクティブになっているスペアディスクは、削除できません。

```
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: resilvered 77.5K in 2s with 0 errors on Mon Mar 18 18:31:12 2019

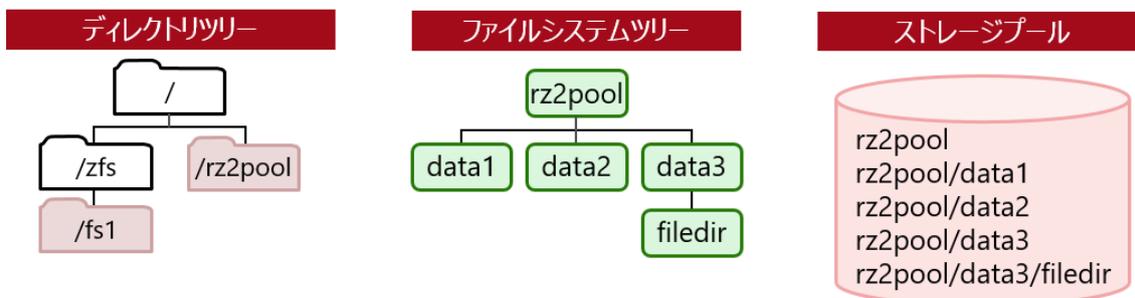
config:

    NAME            STATE             READ WRITE CKSUM
    mirpool          ONLINE            0     0     0
      mirror-0      ONLINE            0     0     0
        c2d3         ONLINE            0     0     0
        c2d4         ONLINE            0     0     0
        spare-2      ONLINE            0     0     0
          c2d10      ONLINE            0     0     0
          c2d11      ONLINE            0     0     0
    spares
      c2d11          INUSE

errors: No known data errors
# zpool remove mirpool c2d11
cannot remove device(s): cannot remove this type of vdev
```

## 2. ファイルシステムの構築

本章では、ファイルシステムの作成方法、および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。



### 2.1. ファイルシステムの作成

#### 1) ファイルシステム名を指定し、ファイルシステムを作成します。

ここでは、「`rz2pool/data1`」と「`rz2pool/data2`」という名前のファイルシステムを作成します。

```
# zfs create rz2pool/data1
# zfs create rz2pool/data2
```

▶ 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。

#### 2) 作成したファイルシステムを確認します。

```
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool             203K  29.3G  33.0K  /rz2pool
rz2pool/data1       31.0K  29.3G  31.0K  /rz2pool/data1
rz2pool/data2       31.0K  29.3G  31.0K  /rz2pool/data2
```

#### 3) `-p` オプションを使用し、中間のデータセットを同時に作成します。

ここでは、「`rz2pool/data3/filedir`」という名前のファイルシステムと、存在しない中間のデータセット「`rz2pool/data3`」を同時に作成します。

```
# zfs create -p rz2pool/data3/filedir
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool             285K  29.3G  34.0K  /rz2pool
rz2pool/data1       31.0K  29.3G  31.0K  /rz2pool/data1
rz2pool/data2       31.0K  29.3G  31.0K  /rz2pool/data2
rz2pool/data3       62.9K  29.3G  32.0K  /rz2pool/data3
rz2pool/data3/filedir 31.0K  29.3G  31.0K  /rz2pool/data3/filedir
```

### 《参考》マウントポイントを指定したファイルシステムの作成

ここでは例として、マウントポイントに「/zfs/fs4」を指定し、「rz2pool/data4」という名前のファイルシステムを作成します。

```
# 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
```

### 《参考》ZFS ボリュームの作成

ZFS ボリュームとは、ブロックデバイスを表すデータセットです。

ZFS ボリュームは、/dev/zvol/{dsk,rdsk}/<プール名>ディレクトリのデバイスとして識別されます。

ここでは例として、2 GB の ZFS ボリューム「rz2pool/data4/user1」を作成します。

```
# zfs create -p -V 2G rz2pool/data4/user1
# zfs list -r rz2pool
NAME                                USED  AVAIL  REFER  MOUNTPOINT
~(省略)~
rz2pool/data4                       2.06G  27.2G  30.9K  /zfs/fs4
rz2pool/data4/user1                 2.06G  29.3G  15.9K  -
```

## 2.2. ファイルシステム名の変更

- 1) 作成したファイルシステム「rz2pool/data3/filedir」の名前を、「rz2pool/data3/files」に変更します。

```
# zfs rename rz2pool/data3/filedir rz2pool/data3/files
```

- 2) 変更したファイルシステム名を確認します。

```
# zfs list -r rz2pool
NAME                                USED  AVAIL  REFER  MOUNTPOINT
~(省略)~
rz2pool/data3/files                 30.9K  29.3G  30.9K  /rz2pool/data3/files
```

## 2.3. プロパティの設定

作成したファイルシステムに各種のプロパティを設定することで、ファイルシステムの動作を制御します。設定可能なプロパティの中には、親に設定したプロパティが配下のすべてのファイルシステムに継承されるものがあります。

### 2.3.1. マウントポイントの変更

初期設定では、すべてのファイルシステムは、OS 起動時にサービス管理機能(SMF)の `svc:/system/filesystem/local:default` サービスによって自動マウントされます。

`zfs set` コマンドで `mountpoint` プロパティを指定することで、マウントポイントを変更できます。また、`mountpoint` プロパティは、上位データセットから継承されます。

レガシーマウントを利用すると、従来のように `/etc/vfstab` ファイルや `mount/umount` コマンドで、ファイルシステムのマウント/アンマウントを管理できます。レガシーマウントを利用する場合は、`mountpoint` プロパティに `legacy` を設定します。

ここでは例として、「`rz2pool/data3`」ファイルシステムのマウントポイントを「`/zfs/fs1`」に設定する手順を示します。

#### 1) マウントポイントプロパティの設定値を確認します。

```
# zfs get -r mountpoint rz2pool
NAME                PROPERTY  VALUE                SOURCE
rz2pool             mountpoint /rz2pool             default
~(省略)~
rz2pool/data3       mountpoint /rz2pool/data3      default
rz2pool/data3/files mountpoint /rz2pool/data3/files default
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool             2.06G  27.2G  35.9K  /rz2pool
~(省略)~
rz2pool/data3       63.8K  3.00G  32.9K  /rz2pool/data3
rz2pool/data3/files 30.9K  3.00G  30.9K  /rz2pool/data3/files
```

#### 2) 「`rz2pool/data3`」のマウントポイントプロパティを「`/zfs/fs1`」に設定します。

```
# zfs set mountpoint=/zfs/fs1 rz2pool/data3
```

- ▶ `mountpoint` のプロパティは、自動で継承されます。
- ▶ `mountpoint` プロパティを変更すると、ファイルシステムが古いマウントポイントから自動的にマウント解除され、新しいマウントポイントに再マウントされます。

### 3) 設定したマウントポイントプロパティの設定値を確認します。

```
# zfs get -r mountpoint rz2pool
NAME                PROPERTY  VALUE                SOURCE
rz2pool             mountpoint /rz2pool            default
~(省略)~
rz2pool/data3       mountpoint /zfs/fs1            local
rz2pool/data3/files mountpoint /zfs/fs1/files      inherited from rz2pool/data3
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool             2.06G 27.2G 35.9K  /rz2pool
~(省略)~
rz2pool/data3       81.7K 3.00G 32.9K  /zfs/fs1
rz2pool/data3/files 48.8K 3.00G 30.9K  /zfs/fs1/files
```

#### 2.3.2. ファイルシステムの共有と共有解除

Oracle Solaris 11.1 以降の ZFS におけるファイルシステムの共有は、share.nfs プロパティを設定することで実現します。Oracle Solaris 10 のように/etc/dfs/dfstab ファイルを編集しても、ファイルシステムの共有は設定されません。

なお、Solaris 11 11/11 の場合は、手順が異なります。11/11 の手順については、「[《参考》 Oracle Solaris 11 11/11 のファイルシステムの共有と共有解除](#)」を参照してください。

#### 1) ファイルシステムの share.nfs プロパティを確認します。

デフォルトは「off」（無効）です。

```
# zfs get share.nfs rz2pool/data2
NAME                PROPERTY  VALUE  SOURCE
rz2pool/data2       share.nfs  off    default
```

#### 2) 共有設定されているファイルシステムがないことを確認します。

```
# share
IPC$                smb      -      Remote IPC
```

- ▶ ファイルシステムの共有の状態は、share コマンドで確認できます。
- ▶ 共有されているファイルシステムが存在しない場合は、何も出力されません。

#### 3) ファイルシステムの共有を設定します。

share.nfs プロパティを有効にします。

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

- 4) `share.nfs` プロパティが「on」(有効)であることを確認します。

```
# zfs get share.nfs rz2pool/data2
NAME          PROPERTY  VALUE  SOURCE
rz2pool/data2 share.nfs  on     local
```

- 5) 共有設定されているファイルシステムを確認します。

```
# share
IPC$          smb      -       Remote IPC
rz2pool_data2 /rz2pool/data2 nfs      sec=sys,rw
```

- 6) ファイルシステムの共有を一時的に解除します。

```
# zfs unshare rz2pool/data2
```

▶ `-a` オプションを使用すると、共有されているすべてのファイルシステムの共有を解除します。

- 7) `share.nfs` プロパティの状態を確認し、共有を一時的に解除したあとも共有設定が有効であることを確認します。

```
# zfs get share.nfs rz2pool/data2
NAME          PROPERTY  VALUE  SOURCE
rz2pool/data2 share.nfs  on     local
```

- 8) 共有されているファイルシステムを確認します。

一時的に共有を解除しているため、何も出力されません。

```
# share
IPC$          smb      -       Remote IPC
```

- 9) ファイルシステムの共有を再開します。

```
# zfs share rz2pool/data2
```

- 10) 共有されているファイルシステムを確認します。

共有を再開したため、該当のファイルシステムが出力されます。

```
# share
IPC$          smb      -       Remote IPC
rz2pool_data2 /rz2pool/data2 nfs      sec=sys,rw
```

### 《参考》 Oracle Solaris 11 11/11 のファイルシステムの共有と共有解除

Oracle Solaris 11 11/11 の ZFS でファイルシステムを共有するには、共有を作成(zfs set share)し、NFS 共有を公開(share nfs=on)します。

(1) 現在のファイルシステムの sharenfs プロパティを確認します。

```
# zfs get share.nfs rz2pool/data2
NAME          PROPERTY  VALUE  SOURCE
rz2pool/data2 share.nfs  off    local
```

▶ sharenfs プロパティは、別名 (share.nfs) で表示されます。

(2) 共有されているファイルシステムが存在しないことを確認します。

```
# share
IPC$          smb      -        Remote IPC
```

▶ ファイルシステムの共有の状態は、share コマンドで確認ができます。

▶ 共有されているファイルシステムが存在しない場合は、項目のみ表示されます。

(3) 共有を作成します。

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

▶ 「share=name」は、共有名を指定します。

▶ 「path」は、共有するファイルシステムまたは NFS 共有のパスを指定します。

▶ 「prot」は、NFS または SMB などの共有プロトコルを指定します。

(4) 作成した共有を公開します。

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

(5) sharenfs プロパティを確認して、共有が公開されたことを確認します。

```
# zfs get -r sharenfs rz2pool/data2
NAME          PROPERTY  VALUE  SOURCE
rz2pool/data2 share.nfs  on     local
rz2pool/data2%d2 share.nfs  on     inherited from rz2pool/data2
```

▶ sharenfs プロパティは、別名 (share.nfs) で表示されます。

(6) 共有されているファイルシステムを確認します。

```
# share
IPC$          smb      -        Remote IPC
d2            /rz2pool/data2  nfs      sec=sys, rw
```

(7) ファイルシステムの共有を解除します。

```
# zfs unshare rz2pool/data2
# share
IPC$          smb      -      Remote IPC
```

- ▶ -a オプションを使用すると、共有されているすべてのファイルシステムの共有を解除します。
- ▶ ファイルシステムの共有の状態は、share コマンドで確認できます。

(8) ファイルシステムの共有を再開します。

```
# zfs share rz2pool/data2
# share
IPC$          smb      -      Remote IPC
d2            /rz2pool/data2  nfs    sec=sys, rw
```

### 2.3.3. ファイルシステムの使用可能領域の上限設定 (quota プロパティ)

1) 「rz2pool/data3」の quota プロパティを確認します。

```
# zfs get quota rz2pool/data3
NAME          PROPERTY  VALUE  SOURCE
rz2pool/data3 quota     none   default
```

2) 「rz2pool/data3」が使用できる容量の上限を 3 GB に設定します。

```
# zfs set quota=3G rz2pool/data3
```

- ▶ 設定を解除する場合は、「quota=none」と設定します。
- ▶ ファイルシステムで消費されている容量がシステムに反映されるまでにタイムラグがあるため、quota プロパティに設定した値を超えてデータを書き込める場合があります。

3) 設定した quota プロパティを確認します。

```
# zfs get quota rz2pool/data3
NAME          PROPERTY  VALUE  SOURCE
rz2pool/data3 quota     3G     local
```

- ▶ -r オプションを使用すると、子孫のプロパティも確認できます。

SOURCE の列は、プロパティ値の設定元を表示します。表示される値の意味を、以下に示します。

SOURCE の値	説明
default	プロパティのデフォルト値が使用されている状態
inherited from <i>dataset-name</i>	<i>dataset-name</i> に指定されているデータセットから継承されている状態
local	zfs set コマンドで、データセットに明示的に設定している状態
temporary	プロパティ値は、zfs mount -o オプションで設定され、マウントされている間だけ有効となる状態
-	読み取り専用プロパティであり、値は ZFS により生成される

### 2.3.4. ユーザー／グループ割り当て制限

割り当て制限は、ユーザーごと、およびグループごとに設定できます。割り当て制限が設定されていないすべてのユーザー／グループに設定する方法、および特定のユーザー／グループごとに設定する方法があります。割り当て制限の設定を、以下に示します。

プロパティ	説明
defaultuserquota=size   none	デフォルトのユーザー割り当て制限
defaultgroupquota=size   none	デフォルトのグループ割り当て制限
userquota@user= size   none   default	特定のユーザー割り当て制限
groupquota@group= size   none   default	特定のグループ割り当て制限

- ▶ none は、割り当て制限を解除し、割り当て制限なしにします。
- ▶ default は、割り当て制限のデフォルト値に設定します。

設定例を以下に示します。

- デフォルトのグループ割り当て制限を設定する場合

```
# zfs set defaultuserquota=25G rz2pool/data3
# zfs get -r defaultuserquota rz2pool
NAME                PROPERTY            VALUE  SOURCE
rz2pool             defaultuserquota    none   -
rz2pool/data1       defaultuserquota    none   -
~(省略)~
rz2pool/data3       defaultuserquota    25G   -
rz2pool/data3/files defaultuserquota    none   -
~(省略)~
```

- 特定のグループ割り当て制限を設定する場合

本書では、事前に追加したグループ「group01」に割り当て制限を設定します。

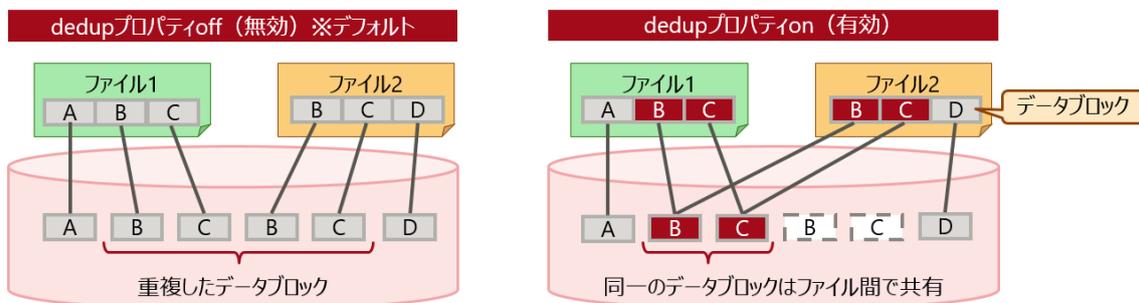
```
# zfs set groupquota@group01=500M rz2pool/data3
# zfs get -r groupquota@group01 rz2pool
NAME                PROPERTY            VALUE  SOURCE
rz2pool             groupquota@group01  none   default
rz2pool/data1       groupquota@group01  none   default
~(省略)~
rz2pool/data3       groupquota@group01  500M   local
rz2pool/data3/files groupquota@group01  none   default
~(省略)~
```

- ▶ グループを追加する方法については、Oracle 社のマニュアルを参照してください。  
『Managing User Accounts and User Environments in Oracle Solaris 11.4』(Oracle 社)  
[https://docs.oracle.com/cd/E37838\\_01/html/E60996/addgroup-1.html#scrolltoc](https://docs.oracle.com/cd/E37838_01/html/E60996/addgroup-1.html#scrolltoc)

### 2.3.5. データ重複の検知・排除

dedup プロパティの設定により、ブロック単位でデータの重複検知や排除が可能です。

重複排除を利用すると、格納されているブロックと同一内容のブロックはオンラインで除去され、データ重複の無駄をブロックレベルで省き、ストレージ利用効率を向上できます。



#### 1) 重複排除を設定するファイルシステムを確認します。

```
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool             2.06G 27.2G 37.9K  /rz2pool
rz2pool/data1       50.8K 27.2G 30.9K  /rz2pool/data1
rz2pool/data2       53.3K 27.2G 32.4K  /rz2pool/data2
~(省略)~
```

#### 2) 現在の使用容量と重複排除率を確認します。

```
# zpool list rz2pool
NAME    SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
rz2pool 89.5G 3.29M  89.5G  0%  1.00x  ONLINE  -
```

#### 3) ファイルシステムの重複排除の設定を有効にします。

```
# zfs set dedup=on rz2pool/data1
# zfs set dedup=on rz2pool/data2
```

#### 4) 重複排除の設定を確認します。

VALUE が「on」になっていることを確認します。

```
# zfs get dedup rz2pool/data1
NAME                PROPERTY  VALUE  SOURCE
rz2pool/data1      dedup     on     local
# zfs get dedup rz2pool/data2
NAME                PROPERTY  VALUE  SOURCE
rz2pool/data2      dedup     on     local
```

## 5) 重複排除を設定した各ファイルシステムに、同一のファイルを配置します。

- i) 「/rz2pool/data1」に格納したファイルを確認します。

本手順では、例として Oracle Solaris 11.4 の Text Installer の ISO イメージを格納します。

```
# ls -l /rz2pool/data1
total 1876415
-rw-r--r--  1 root    root    1018736640 May  8 16:46 sol-11_4-text-sparc.iso
# du -h /rz2pool/data1/sol-11_4-text-sparc.iso
916M /rz2pool/data1/sol-11_4-text-sparc.iso
```

- ii) 「/rz2pool/data2」に格納したファイルを確認します。

```
# ls -l /rz2pool/data2
total 1876415
-rw-r--r--  1 root    root    1018736640 May  8 16:51 sol-11_4-text-sparc.iso
# du -h /rz2pool/data2/sol-11_4-text-sparc.iso
916M /rz2pool/data2/sol-11_4-text-sparc.iso
```

## 6) 重複排除率を確認します。

ストレージプール単位で共通ブロックを共有するため、重複排除率はストレージプール単位で確認します。DEDUP には、どれだけの重複除去が行われたかを表す重複排除率が表示されます。

```
# zpool list rz2pool
NAME      SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
rz2pool  89.5G  2.70G  86.8G   3%  2.00x  ONLINE  -
```

- ▶ 「/rz2pool」には約 900 MB のファイルが 2 つ存在しますが、使用容量は 5.4 GB (raidz2-0 構成のため)ではなく約 2.7 GB になっており、重複排除機能の効果が確認できます。
- ▶ DEDUP 列の値が「2.00x」と表示されていることから、約半分の容量になっていることが確認できます。

### 2.3.6. データの暗号化

encryption プロパティを設定することで、ファイルシステムを暗号化できます。

暗号化とは、機密保護のためにデータをエンコードするプロセスで、エンコードされたデータにアクセスするには鍵が必要になります。

ZFS 暗号化には、以下の特長があります。

- 鍵の変更や再入力などの暗号化操作は、オンラインで実行可能です。
- ZFS のファイルシステム単位で暗号化ポリシーを設定できます。しかし、暗号化ポリシーの変更はできません。
- ZFS 暗号化は、配下のファイルシステムに継承できます。

## 1) 暗号化するファイルシステムを新規作成します。

パスワードは、任意の文字列で、8文字以上で設定する必要があります。

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

## 2) 暗号化したファイルシステムを確認します。

VALUE が「on」になっていることを確認します。

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

## 3) 暗号化されたファイルシステムのスナップショットを作成します。

```
# zfs snapshot rz2pool/data5@encryption
```

## 4) 作成したスナップショットを確認します。

```
# zfs list -r -t snapshot rz2pool
NAME                               USED  AVAIL  REFER  MOUNTPOINT
rz2pool/data5@encryption           0    -    33.0K  -
```

## 5) 暗号化されたファイルシステムのクローンを作成します。

暗号化が設定されているクローンを作成する場合には、新たにパスワードを設定する必要があります。

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

## 6) 作成したクローンを確認します。

```
# zfs list -r rz2pool
NAME                               USED  AVAIL  REFER  MOUNTPOINT
rz2pool                           2.06G  27.2G  37.9K  /rz2pool
rz2pool/data1                     50.8K  27.2G  30.9K  /rz2pool/data1
rz2pool/data2                     53.3K  27.2G  32.4K  /rz2pool/data2
～(省)～
rz2pool/data5                     32.9K  27.2G  32.9K  /rz2pool/data5
rz2pool/data5-clone               18.9K  27.2G  32.9K  /rz2pool/data5-clone
```

### 2.3.7. プロパティの継承

割り当て (quota) と予約 (reservation) を除き、設定可能なすべてのプロパティは上位データセットから値を継承します。明示的な値が上位データセットに設定されていない場合は、デフォルト値が設定されます。

ここでは例として、いったん `rz2pool/data3/files` に圧縮 (compression) 「on」を設定したあと、上位データセットからの設定 (default) を継承する手順を示します。

- 1) プロパティを設定し、「`rz2pool/data3/files`」の `compression` 設定値を「on」に設定します。

```
# zfs set compression=on rz2pool/data3/files
```

- 2) 「`rz2pool`」の `compression` 設定値が「on」になっていることを確認します。

```
# zfs get -r compression rz2pool
NAME                PROPERTY    VALUE    SOURCE
rz2pool             compression off      default
～(省略)～
rz2pool/data3       compression off      default
rz2pool/data3/files compression  on       local
```

▶ `-r` オプションを使用すると、指定したデータセット配下に再帰的に実行されます。

- 3) `compression` 設定を上位データセットから継承します。

```
# zfs inherit compression rz2pool/data3/files
```

- 4) 上位データセットからの継承後、「`rz2pool`」の `compression` 設定値を確認します。

設定値が「off」になっていることを確認してください。

```
# zfs get -r compression rz2pool
NAME                PROPERTY    VALUE    SOURCE
rz2pool             compression off      default
～(省略)～
rz2pool/data3       compression off      default
rz2pool/data3/files compression  off      default
```

## 2.4. ファイルシステムの削除

### Point

「[《参考》マウントポイントを指定したファイルシステムの作成](#)」と「[《参考》ZFS ボリュームの作成](#)」を実施していない場合、手順 1)は不要です。

#### 1) 作成したファイルシステム「rz2pool/data4」を削除します。

```
# zfs destroy -r rz2pool/data4
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool             1.79G 27.2G 36.0K  /rz2pool
rz2pool/data1       916M 27.2G 916M  /rz2pool/data1
rz2pool/data2       916M 27.2G 916M  /rz2pool/data2
rz2pool/data3       63.9K 3.00G 33.0K  /zfs/fs1
rz2pool/data3/files 31.0K 3.00G 31.0K  /zfs/fs1/files
rz2pool/data5       33.0K 27.2G 33.0K  /rz2pool/data5
rz2pool/data5-clone 19.0K 27.2G 33.0K  /rz2pool/data5-clone
```

▶ `-r` オプションを使用すると、指定したデータセット配下のデータセットを再帰的に削除できます。

#### 2) 作成したファイルシステム「rz2pool/data5」を削除します。

```
# zfs destroy -R rz2pool/data5
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool             1.79G 27.2G 36.0K  /rz2pool
rz2pool/data1       916M 27.2G 916M  /rz2pool/data1
rz2pool/data2       916M 27.2G 916M  /rz2pool/data2
rz2pool/data3       63.9K 3.00G 33.0K  /zfs/fs1
rz2pool/data3/files 31.0K 3.00G 31.0K  /zfs/fs1/files
```

▶ `-R` オプションを使用すると、指定したデータセットのクローンファイルシステムを含む、すべての依存関係を再帰的に削除できます。

## 2.5. ファイルシステムのマウント／アンマウント

ZFS のファイルシステムは、ファイルシステムが作成される時、またはシステムが起動するときに、自動的にマウントされます。

マウントポイントを変更するとき、または手動でファイルシステムをマウント／アンマウントするときは、zfs mount コマンドを使用します。

zfs mount コマンドを引数なしで実行すると、ZFS が現在マウントしているファイルシステムをすべて表示します。ただし、レガシーマウントで管理されているマウントポイントは表示されません。

### 1) 現在のファイルシステムのマウント状態を確認します。

```
# zfs mount
~(省略)~
rz2pool                /rz2pool
rz2pool/data1         /rz2pool/data1
rz2pool/data2         /rz2pool/data2
rz2pool/data3         /zfs/fs1
rz2pool/data3/files   /zfs/fs1/files
~(省略)~
```

### 2) 「rz2pool/data3」をアンマウントします。以下のどちらかを実行します。

```
# zfs unmount rz2pool/data3
```

```
# zfs unmount /zfs/fs1
```

- ▶ `-a` オプションを使用すると、/(root)ファイルシステムを除き、ZFS が管理しているファイルシステムをすべてアンマウントできます。
- ▶ アンマウントは、ファイルシステム名またはマウントポイントを指定して実行できます。

### 3) アンマウント実行後、ファイルシステムのマウント状態を確認します。

```
# zfs mount
~(省略)~
rz2pool                /rz2pool
rz2pool/data1         /rz2pool/data1
rz2pool/data2         /rz2pool/data2
~(省略)~
```

### 4) 「rz2pool/data3」をマウントします。

```
# zfs mount rz2pool/data3
```

## 5) マウント実行後、ファイルシステムのマウント状態を確認します。

```
# zfs mount
~(省略)~
rz2pool                /rz2pool
rz2pool/data1          /rz2pool/data1
rz2pool/data2          /rz2pool/data2
rzpool                 /rzpool
rz2pool/data3        /zfs/fs1
```

- ▶ `-a` オプションを使用すると、ZFS が管理しているファイルシステムをすべてマウントします。レガシーマウントで管理されているファイルシステムは、マウントされません。
- ▶ `-O` オプションを使用すると、空ではないディレクトリへ強制的にマウントします。

## 《参考》レガシーマウントの設定

「rz2pool/data3」ファイルシステムのマウントポイントを「legacy」に設定する場合の手順を、以下に示します。

## (1) マウントポイントプロパティの設定値を確認します。

```
# zfs get -r mountpoint rz2pool
NAME                PROPERTY  VALUE                SOURCE
rz2pool             mountpoint /rz2pool            default
~(省略)~
rz2pool/data3       mountpoint /zfs/fs1            local
rz2pool/data3/files mountpoint /zfs/fs1/files     inherited from rz2pool/data3
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool             1.79G  27.2G  33.0K  /rz2pool
~(省略)~
rz2pool/data3       63.9K  3.00G  33.0K  /zfs/fs1
rz2pool/data3/files 31.0K  3.00G  31.0K  /zfs/fs1/files
```

## (2) マウントポイントプロパティを「legacy」に設定します。

```
# zfs set mountpoint=legacy rz2pool/data3
```

(3) 設定したマウントポイントプロパティの設定値を確認します。

```
# zfs get -r mountpoint rz2pool
NAME                PROPERTY  VALUE          SOURCE
rz2pool             mountpoint /rz2pool      default
~(省略)~
rz2pool/data3       mountpoint legacy         local
rz2pool/data3/files mountpoint legacy         inherited from rz2pool/data3
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool             1.79G  27.2G  33.0K  /rz2pool
~(省略)~
rz2pool/data3       63.9K  3.00G  33.0K  legacy
rz2pool/data3/files 31.0K  3.00G  31.0K  legacy
```

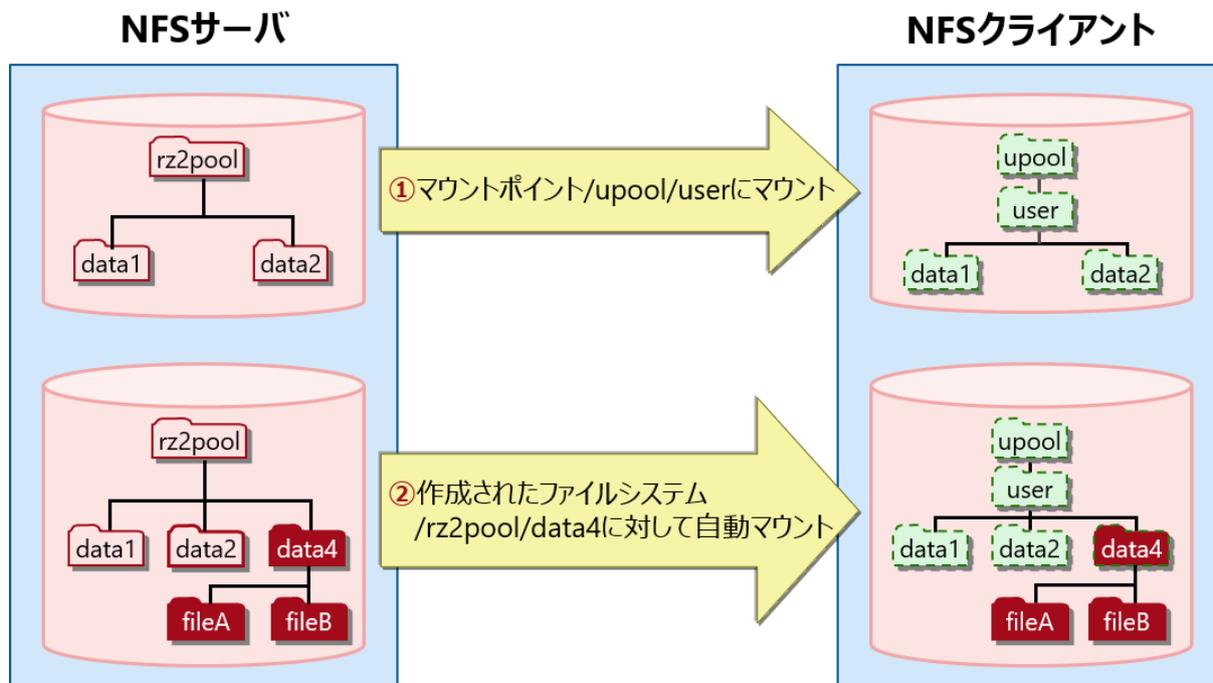
▶ legacy を解除する場合、手順(2)のマウントポイントプロパティの設定で、マウントポイントを再指定してください。

#### 《参考》レガシーファイルシステムをマウント／アンマウントする

```
# mount -F zfs rz2pool/data3 /mnt
# umount /mnt
```

## 2.6. ファイルシステムミラーマウント

ファイルシステムミラーマウントは、マウント済みのファイルシステム配下に作成されたファイルシステムを、NFS クライアントが自動的に認識する機能です。



### 1) NFS サーバで、共有するファイルシステムを確認します。

```
NFS-Server# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool             1.79G 27.2G 33.0K  /rz2pool
rz2pool/data1       916M 27.2G 916M   /rz2pool/data1
rz2pool/data2       916M 27.2G 916M   /rz2pool/data2
rz2pool/data3       63.9K 3.00G 33.0K  /zfs/fs1
rz2pool/data3/files 31.0K 3.00G 31.0K  /zfs/fs1/files
```

### 2) NFS サーバで、該当のファイルシステムを共有設定します。

```
NFS-Server# zfs set share.nfs=on rz2pool
NFS-Server# share
IPC$                smb      -      Remote IPC
rz2pool /rz2pool         nfs     sec=sys, rw
rz2pool_data1 /rz2pool/data1  nfs     sec=sys, rw
rz2pool_data2 /rz2pool/data2  nfs     sec=sys, rw
rz2pool_data3 /zfs/fs1        nfs     sec=sys, rw
rz2pool_data3_files /zfs/fs1/files  nfs     sec=sys, rw
```

### 3) NFS クライアントで、NFS マウントを実行します(マウントポイント:/upool/user)。

```
NFS-Client# zfs unmount upool/user
NFS-Client# mount -F nfs 192.168.10.xxx:/rz2pool /upool/user
```

## 4) NFS クライアントで、マウントしたファイルシステムに存在するファイルを確認します。

```
NFS-Client# ls /upool/user  
data1 data2
```

## 5) NFS サーバで、ファイルシステムを新規作成します。

```
NFS-Server# zfs create rz2pool/data4  
NFS-Server# zfs create rz2pool/data4/fileA  
NFS-Server# zfs create rz2pool/data4/fileB  
NFS-Server# zfs list -r rz2pool  
NAME                USED  AVAIL  REFER  MOUNTPOINT  
rz2pool             1.79G 27.2G 35.0K  /rz2pool  
rz2pool/data1       916M 27.2G 916M   /rz2pool/data1  
rz2pool/data2       916M 27.2G 916M   /rz2pool/data2  
rz2pool/data3       63.9K 3.00G 33.0K  /zfs/fs1  
rz2pool/data3/files 31.0K 3.00G 31.0K  /zfs/fs1/files  
rz2pool/data4       94.9K 27.2G 33.0K  /rz2pool/data4  
rz2pool/data4/fileA 31.0K 27.2G 31.0K  /rz2pool/data4/fileA  
rz2pool/data4/fileB 31.0K 27.2G 31.0K  /rz2pool/data4/fileB
```

## 6) NFS クライアントで、マウントしたファイルシステムを確認します。

NFS サーバで新規作成したファイルシステムを、NFS クライアントが自動認識していることを確認します。

```
NFS-Client# ls /upool/user  
data1 data2 data4  
NFS-Client# ls /upool/user/data4  
fileA fileB
```

## 7) NFS サーバで、ファイルシステムを削除します。

```
NFS-Server# zfs destroy -r rz2pool/data4  
NFS-Server# zfs list -r rz2pool  
NAME                USED  AVAIL  REFER  MOUNTPOINT  
rz2pool             1.79G 27.2G 35.0K  /rz2pool  
rz2pool/data1       916M 27.2G 916M   /rz2pool/data1  
rz2pool/data2       916M 27.2G 916M   /rz2pool/data2  
rz2pool/data3       63.9K 3.00G 33.0K  /zfs/fs1  
rz2pool/data3/files 31.0K 3.00G 31.0K  /zfs/fs1/files
```

## 8) NFS クライアントで、マウントしたファイルシステムを確認します。

NFS サーバで削除したファイルシステムを、NFS クライアントが自動認識していることを確認します。

```
NFS-Client# ls /upool/user  
data1 data2
```

### 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方法を説明します。

#### 3.1. スナップショットの作成

スナップショットとは、ファイルシステムまたは ZFS ボリュームのコピーで、読み取り専用です。スナップショットは、瞬時に作成できます。

スナップショット名は、「ファイルシステム名@任意のスナップショット名」で指定します。スナップショット名の命名方法は、「[1.3. ストレージプール\(ユーザー領域\)の作成](#)」に示した規則に従います。例を以下に示します。

```
filesystem@snapname
```

```
volume@snapname
```

ZFS スナップショットには、以下の特長があります。

- スナップショットは、システムの再起動後も残ります。
- スナップショットの理論上の最大数は、 $2^{64}$  (2 の 64 乗) です。
- スナップショットは、作成元のファイルシステムと同じストレージプールの領域を消費します。

ZFS スナップショットの留意事項を、以下に示します。

- ボリュームのスナップショットには直接アクセスできません。スナップショットの複製、バックアップ、ロールバックなどは実行できます。
- スナップショットには、変更できるプロパティはありません。また、データセットのプロパティは、スナップショットに適用できません。
- データセットのスナップショットが存在する場合、データセットは破棄できません。

#### 1) 「rz2pool/data1」のスナップショットを、「now」という名前で作成します。

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

#### 《参考》 再帰的なスナップショットの作成

「rz2pool/data3」配下のデータセットすべてのスナップショットを、「now」という名前で作成します。

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

- ▶ `-r` オプションを使用すると、指定したデータセット配下のスナップショットを再帰的に作成します。

## 2) スナップショットを確認します。

スナップショットのみを表示します。

```
# zfs list -r -t snapshot rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool/data1@now    0     -    916M   -
rz2pool/data3@now    0     -    33.0K  -
rz2pool/data3/files@now 0     -    31.0K  -
```

▶ 表示するファイルタイプ「snapshot」を-t オプションで指定すると、スナップショットのみを表示します。

## 3) 指定したスナップショットを削除します。

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

### 《参考》再帰的なスナップショットの削除

指定したデータセット配下のスナップショットをすべて削除します。

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

▶ -r オプションを使用すると、指定したデータセット配下のスナップショットを再帰的に削除します。

## 4) スナップショットを確認します。

```
# zfs list -r -t snapshot rz2pool
no datasets available
```

### 《参考》スナップショット名の変更

スナップショットの名前を「now」から「before」に変更します。以下のどちらかを実行します。

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

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

名前が変更されたことを確認します。

```
# zfs list -r -t snapshot rz2pool
NAME                               USED  AVAIL  REFER  MOUNTPOINT
rz2pool/data1@before                0     -    916M  -
rz2pool/data3@now                   0     -    33.0K  -
rz2pool/data3/files@now              0     -    31.0K  -
```

-r オプションを使用すると、指定したデータセット配下のスナップショット名を再帰的に変更します。

```
# zfs rename -r rz2pool/data3@now before
# zfs list -r -t snapshot rz2pool
NAME                               USED  AVAIL  REFER  MOUNTPOINT
rz2pool/data1@before                0     -    916M  -
rz2pool/data3@before                0     -    33.0K  -
rz2pool/data3/files@before          0     -    31.0K  -
```

ファイルシステムのスナップショットには、ファイルシステムを含むルートの.zfs/snapshot ディレクトリからアクセスできます。例えば、rz2pool/data1 が/rz2pool/data1 にマウントされている場合は、rz2pool/data1@before スナップショットのデータに/rz2pool/data1/.zfs/snapshot/before ディレクトリからアクセスできます。

```
# ls /rz2pool/data1/.zfs/snapshot/before
sol-11_4-text-sparc.iso
```

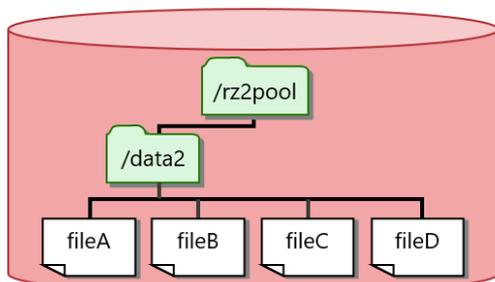
- ▶ データは、読み取り専用でアクセスできます。
- ▶ .zfs ディレクトリは ls コマンドで確認できないため、スナップショットのデータはフルパスで指定する必要があります。

### 3.2. スナップショットの差分表示

zfs diff コマンドで、2 つのスナップショット間の差分を確認できます。

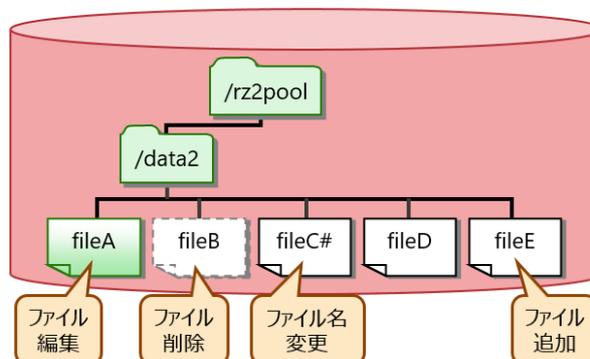
✓ スナップショットA（データ更新前）

rz2pool/data2@snap1



✓ スナップショットB（データ更新後）

rz2pool/data2@snap2



1) rz2pool のファイルシステムを確認します。

```
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool              2.06G 27.2G  34.9K  /rz2pool
rz2pool/data1        48.8K 27.2G  30.9K  /rz2pool/data1
rz2pool/data2      30.9K 27.2G  30.9K  /rz2pool/data2
~(省略)~
```

2) ファイルを作成します。

```
# touch /rz2pool/data2/fileA
# touch /rz2pool/data2/fileB
# touch /rz2pool/data2/fileC
# touch /rz2pool/data2/fileD
```

▶ 「2.3.5. データ重複の検知・排除」で/rz2pool/data2 にファイルを格納している場合は、削除してください。

3) ファイルを確認します。

初期状態は 4 つのファイルが存在する状態です。

```
# ls -l /rz2pool/data2
total 4
-rw-r--r--  1 root   root           0 Mar 19 20:01 fileA
-rw-r--r--  1 root   root           0 Mar 19 20:01 fileB
-rw-r--r--  1 root   root           0 Mar 19 20:01 fileC
-rw-r--r--  1 root   root           0 Mar 19 20:01 fileD
```

4) スナップショットを作成します。

```
# zfs snapshot rz2pool/data2@snap1
```

## 5) スナップショットを確認します。

```
# zfs list -r -t snapshot rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool/data2@snap1 20.0K - 31.0K -
```

## 6) 各ファイルにさまざまな変更を加えます。

```
# echo ABC >> /rz2pool/data2/fileA      ←ファイル編集
# rm /rz2pool/data2/fileB                ←ファイル削除
# mv /rz2pool/data2/fileC /rz2pool/data2/fileC# ←ファイル名変更
# touch /rz2pool/data2/fileE             ←ファイル追加
```

## 7) ファイルの更新後、再度スナップショットを作成します。

```
# zfs snapshot rz2pool/data2@snap2
```

## 8) 作成したスナップショットを確認します。

```
# zfs list -r -t snapshot rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool/data2@snap1 20.0K - 31.0K -
rz2pool/data2@snap2 0      - 33.0K -
```

## 9) スナップショットを比較し、差分を確認します。

```
# zfs diff rz2pool/data2@snap1 rz2pool/data2@snap2
M    /rz2pool/data2/
M    /rz2pool/data2/fileA
-    /rz2pool/data2/fileB
R    /rz2pool/data2/fileC -> /rz2pool/data2/fileC#
+    /rz2pool/data2/fileE
```

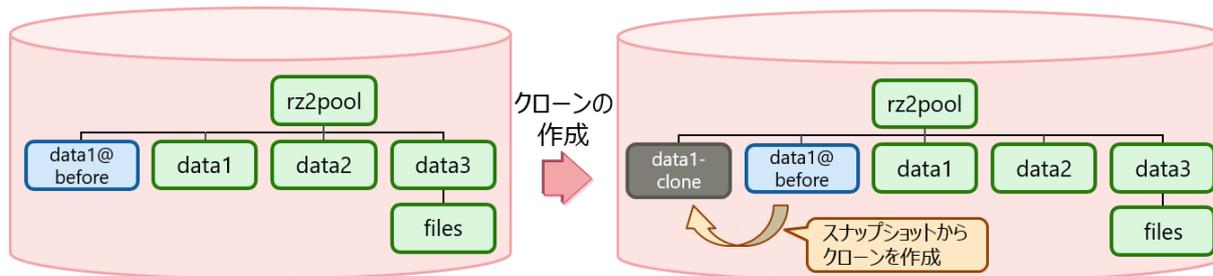
-r オプションを使用すると、指定したスナップショットとすべての子スナップショットの差分を表示できます。差分の表記の意味を、以下に示します。

差分の表記	説明
M	ファイルまたはディレクトリの変更を示す
R	ファイルまたはディレクトリの名前が変更されている
-	ファイルまたはディレクトリが、古いスナップショットにのみ存在する
+	ファイルまたはディレクトリが、新しいスナップショットにのみ存在する

### 3.3. クローンの作成

クローンは、スナップショットから作成するデータセットの複製で、書き込み可能なボリュームまたはファイルシステムです。スナップショットと同様に、クローンは瞬間的に作成され、最初は追加のディスク領域を消費しません。また、クローンのスナップショットを作成することもできます。

クローンは、データセット階層内の別の場所に作成されますが、クローンが存在する間は元のスナップショットを破棄できません。また、クローンには、作成元のデータセットのプロパティは継承されません。



- 1) 「rz2pool/data1」のスナップショットを、「before」という名前で作成します。

```
# zfs snapshot rz2pool/data1@before
# zfs list -r -t snapshot rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool/data1@before  0    -      916M   -
rz2pool/data2@snap1  20.0K -      31.0K   -
rz2pool/data2@snap2  0    -      33.0K   -
```

▶ すでに「rz2pool/data1@before」が存在する場合は、本手順は不要です。

- 2) 「rz2pool/data1@before」のクローンを、「rz2pool/data1-clone」という名前で作成します。

```
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool             2.06G 27.2G 34.9K  /rz2pool
rz2pool/data1       31.9K 27.2G 30.9K  /rz2pool/data1
~(省略)~
# zfs clone rz2pool/data1@before rz2pool/data1-clone
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool             2.06G 27.2G 35.9K  /rz2pool
rz2pool/data1       31.9K 27.2G 30.9K  /rz2pool/data1
rz2pool/data1-clone 17.9K 27.2G 30.9K  /rz2pool/data1-clone
~(省略)~
```

### 《参考》クローンの依存関係の確認

依存関係を確認します。

クローンの場合、VALUE に依存しているスナップショット名が表示されます。

```
# zfs get origin rz2pool/data1-clone
NAME                PROPERTY  VALUE                SOURCE
rz2pool/data1-clone origin    rz2pool/data1@before -
```

通常のファイルシステムの場合、VALUE には何も表示されません。

```
# zfs get origin rz2pool/data1
NAME                PROPERTY  VALUE  SOURCE
rz2pool/data1      origin    -      -
```

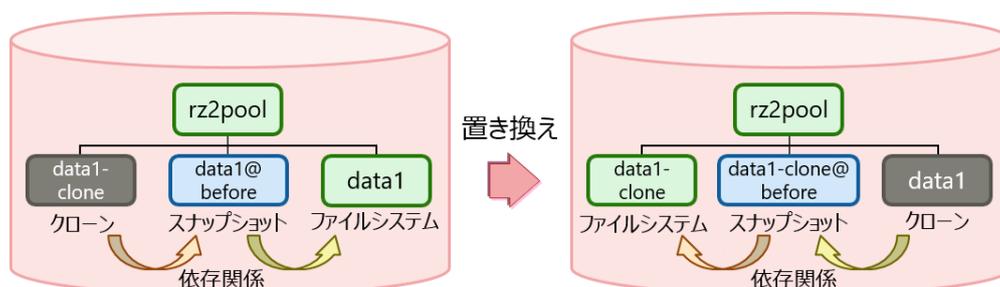
### 《参考》クローンの破棄

クローン「rz2pool/data1-clone」を破棄します。

```
# zfs destroy rz2pool/data1-clone
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool             2.06G 27.2G 34.9K  /rz2pool
rz2pool/data1       31.9K 27.2G 30.9K  /rz2pool/data1
~(省略)~
```

### 3.4. ファイルシステムの置き換え

アクティブなファイルシステムと、アクティブなファイルシステムのクローンの関係を置き換える (promote) ことができます。置き換え後は、元のファイルシステムがクローンとなり、依存関係が逆転します。



#### 1) 「rz2pool/data1-clone」と「rz2pool/data1」の依存関係を確認します。

「rz2pool/data1-clone」は、依存しているスナップショット名が VALUE に表示されます。

```
# zfs get origin rz2pool/data1-clone
NAME                PROPERTY  VALUE                SOURCE
rz2pool/data1-clone origin    rz2pool/data1@before -
```

「rz2pool/data1」は依存関係がないため、VALUE に何も表示されません。

```
# zfs get origin rz2pool/data1
NAME                PROPERTY  VALUE  SOURCE
rz2pool/data1      origin    -      -
```

#### 2) クローン「rz2pool/data1-clone」を置き換えます。

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

#### 3) 置き換え後の依存関係を確認します。

promote によって依存関係が置換されたため、VALUE には何も表示されません。

```
# zfs get origin rz2pool/data1-clone
NAME                PROPERTY  VALUE  SOURCE
rz2pool/data1-clone origin    -      -
```

promote によって依存関係が置換されたため、VALUE にはスナップショット名が表示されます。

```
# zfs get origin rz2pool/data1
NAME                PROPERTY  VALUE                SOURCE
rz2pool/data1      origin    rz2pool/data1-clone@before -
```

**《参考》 クローン作成元ファイルシステムの名前変更**

```
# zfs rename rz2pool/data1 rz2pool/data1.old
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool             2.06G 27.2G 35.9K  /rz2pool
rz2pool/data1-clone 48.8K 27.2G 30.9K  /rz2pool/data1-clone
rz2pool/data1.old 17.9K 27.2G 30.9K  /rz2pool/data1.old
~(省略)~
```

**《参考》 クローンの名前変更**

```
# zfs rename rz2pool/data1-clone rz2pool/data1
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool             2.06G 27.2G 35.9K  /rz2pool
rz2pool/data1     48.8K 27.2G 30.9K  /rz2pool/data1
rz2pool/data1.old  17.9K 27.2G 30.9K  /rz2pool/data1.old
~(省略)~
```

**《参考》 クローン作成元のファイルシステムの削除**

置き換えをすることで、クローン作成元のファイルシステムの削除も可能です。

```
# zfs destroy rz2pool/data1.old
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool             2.06G 27.2G 34.9K  /rz2pool
rz2pool/data1       48.8K 27.2G 30.9K  /rz2pool/data1
~(省略)~
```

### 3.5. スナップショットからのロールバック

ファイルシステムのロールバックを行うと、スナップショットを作成した時点よりあとに加えられたすべての変更を破棄し、スナップショットを作成した時点の状態にファイルシステムを戻すことができます。

ロールバックを行うには、以下の条件を満たす必要があります。

- 最新のスナップショット以外のスナップショットにロールバックすることはできません。最新より古いスナップショットにロールバックするには、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。
- 対象のスナップショットより新しいスナップショットからクローンが作成されている場合は、`-R` オプションを指定することで、クローンも破棄できます。

#### 1) ファイルを作成します。

```
# touch /rz2pool/data1/file01.txt
# ls /rz2pool/data1
file01.txt
```

- ▶ 「[2.3.5. データ重複の検知・排除](#)」で `/rz2pool/data1` にファイルを格納している場合は、削除してください。

#### 2) 「`/rz2pool/data1`」のスナップショットを、「`after`」という名前で作成します。

```
# zfs snapshot rz2pool/data1@after
# zfs list -r -t snapshot rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool/data1@before 1023  -      916M  -
rz2pool/data1@after  0     -      31.0K  -
~(省略)~
```

#### 3) 「`/rz2pool/data1@before`」をロールバックします。

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

- ▶ `-r` オプションを使用すると、指定したスナップショットを最新としてロールバックします。`/rz2pool/data1@before` 以降に作成したスナップショットは、削除されます。
- ▶ 指定したファイルシステムのみ、ロールバックで戻ります。指定したファイルシステムの配下に作成したファイルシステムは、戻りません。指定したファイルシステムの配下に作成したファイルシステムを戻す必要がある場合、個々にロールバックします。

## 4) ロールバック後の状態を確認します。

「file01.txt」がなくなっていることを確認します。

```
# ls /rz2pool/data1  
#
```

▶ 「[2.3.5. データ重複の検知・排除](#)」で/rz2pool/data1 にファイルを格納している場合は、格納したファイルが表示されます。

「rz2pool/data1@after」が削除されていることを確認します。

```
# zfs list -r -t snapshot rz2pool  
NAME                USED  AVAIL  REFER  MOUNTPOINT  
rz2pool/data1@before 1023   -    916M   -  
～(省略)～
```

### 3.6. スナップショットの保護

スナップショットは、不用意に削除されることがないように保護できます。保護したスナップショットは、破棄されません。

また、スナップショットの保護と `zfs destroy -d` コマンドを使用することで、クローンが存在するスナップショットを遅延破棄することが可能です。この場合、保護解除とクローン削除を実行したタイミングで、スナップショットが削除されます。

#### 1) スナップショットを作成します。

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

#### 2) スナップショットを確認します。

```
# zfs list -r -t snapshot rz2pool
NAME                               USED  AVAIL  REFER  MOUNTPOINT
rz2pool/data1@before               1023   -    916M   -
rz2pool/data1@hold_test             0     -    916M   -
rz2pool/data2@snap1                 20.0K  -    31.0K  -
rz2pool/data2@snap2                 19.0K  -    33.0K  -
```

#### 3) 作成したスナップショットを保護します。

```
# zfs hold mytag rz2pool/data1@hold_test
```

▶ `rz2pool/data1@hold_test` に、保持タグ「mytag」を付与します。

#### 4) 保護したスナップショットを確認します。

`hold` コマンドで設定した保持タグの情報が表示されることを確認します。

```
# zfs holds rz2pool/data1@hold_test
NAME                               TAG      TIMESTAMP
rz2pool/data1@hold_test           mytag   Tue Mar 19 19:01:49 2019
```

※ 保護されたスナップショットを削除しようとすると、削除できないというメッセージが表示されます。

```
# zfs destroy rz2pool/data1@hold_test
cannot destroy 'rz2pool/data1@hold_test': snapshot is busy
```

▶ 保護しているスナップショットは削除できません。削除するには、保護を解除する必要があります。

### 《参考》 保護されているスナップショットの削除

(1) スナップショットの保護を解除します。

```
# zfs release mytag rz2pool/data1@hold_test
```

(2) 保護の解除後、スナップショットを削除します。

```
# zfs destroy rz2pool/data1@hold_test
```

(3) スナップショットが削除されていることを確認します。

```
# zfs list -r -t snapshot rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool/data1@before 1023   -    916M   -
rz2pool/data2@snap1 20.0K   -    31.0K   -
rz2pool/data2@snap2 19.0K   -    33.0K   -
```

### 《参考》 スナップショットの遅延破棄

スナップショットの保護が設定されている状態で、遅延破棄(zfs destroy -d)を実行すると、スナップショットの保護が解除され、クローンを削除したタイミングでスナップショットが削除されます。

(1) クローンを作成します。

「rz2pool/data1@hold\_test」のクローンを「rz2pool/data1\_clone」という名前で作成します。

```
# zfs clone rz2pool/data1@hold_test rz2pool/data1_clone
```

(2) 作成したクローンを確認します。

```
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool             919M  27.2G  36.0K  /rz2pool
rz2pool/data1       916M  27.2G  916M   /rz2pool/data1
rz2pool/data1_clone 18.0K  27.2G  916M   /rz2pool/data1_clone
~(省略)~
```

(3) スナップショット「rz2pool/data1@hold\_test」を保護します。

```
# zfs hold mytag rz2pool/data1@hold_test
```

(4) 保護したスナップショットを確認します。

```
# zfs holds rz2pool/data1@hold_test
NAME                TAG    TIMESTAMP
rz2pool/data1@hold_test mytag  Tue Mar 19 19:06:46 2019-
```

(5) -d オプションを使用し、保護されたスナップショットを削除します。

```
# zfs destroy -d rz2pool/data1@hold_test
```

▶ 実際には保護の解除とクローンの削除が実行されるまで削除されません。

(6) 削除したスナップショットの defer\_destroy プロパティを確認します。

遅延破棄を実行すると、defer\_destroy プロパティが「on」になります。

```
# zfs get defer_destroy rz2pool/data1@hold_test
NAME                PROPERTY      VALUE  SOURCE
rz2pool/data1@hold_test  defer_destroy  on     -
```

(7) スナップショットが削除されていないことを確認します。

```
# zfs list -t snapshot rz2pool/data1@hold_test
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool/data1@hold_test    0    -    916M  -
```

(8) 保護を解除します。

```
# zfs release mytag rz2pool/data1@hold_test
```

(9) スナップショットが削除されていないことを確認します。

```
# zfs list -t snapshot rz2pool/data1@hold_test
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool/data1@hold_test    0    -    916M  -
```

▶ スナップショットの保護を解除しただけでは、削除されません。

(10) クローンを削除します。

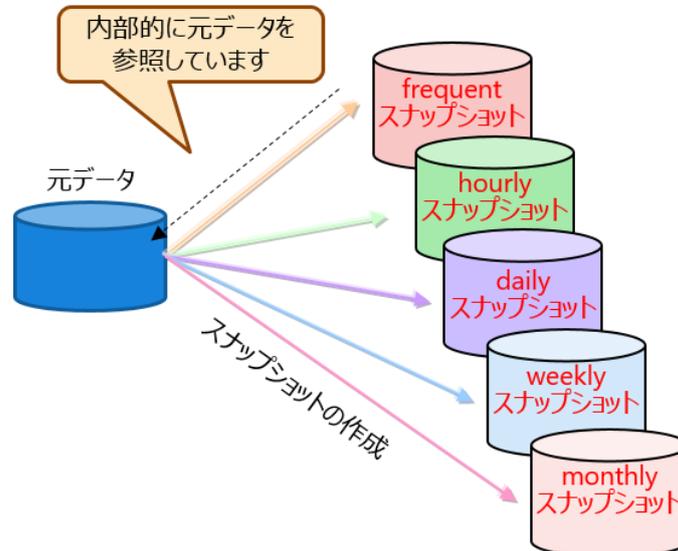
```
# zfs destroy rz2pool/data1_clone
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool             919M  27.2G  35.0K  /rz2pool
rz2pool/data1       916M  27.2G  916M   /rz2pool/data1
rz2pool/data2       71.9K  27.2G  33.0K  /rz2pool/data2
~(省略)~
```

(11) スナップショットが削除されたことを確認します。

```
# zfs list -t snapshot rz2pool/data1@hold_test
cannot open 'rz2pool/data1@hold_test': snapshot does not exist
```

### 3.7. ZFS 自動スナップショット

タイムスライダーサービスを設定すると、自動的にスナップショットを取得できます。ZFS 自動スナップショットを使用するには、パッケージ(time-slider)の追加インストールが必要です。



タイムスライダーサービスの種類、および取得タイミングを以下に示します。

タイムスライダーサービスの種類	取得タイミング
frequent スナップショット	15 分ごと
hourly スナップショット	毎時
daily スナップショット	毎日
weekly スナップショット	毎週
monthly スナップショット	毎月

タイムスライダーサービスは、GNOME 関連のサービスが必要です。本書では、双方のサービスが含まれるパッケージグループである「solaris-desktop」をインストールします。

本書では、BE (Boot Environment) 機能を用いて、パッケージをインストールする手順を説明します。BE に関する詳細は、以下をご参照ください。

- 「Oracle Solaris 11 を使ってみよう」

<https://www.fujitsu.com/jp/sparc/technical/document/solaris/#os>

#### 1) 新規 BE「be01」を作成し、グループパッケージ「solaris-desktop」をインストールします。

```
# beadm create be01
# beadm list
BE Name      Flags Mountpoint Space  Policy Created
-----
be01         -      -          95.07M static 2019-03-22 14:43
solaris      NR     /          4.15G static 2019-03-01 15:16
```

- ▶ BE 名は、任意に指定可能です。

## 2) 新規作成した BE「be01」を、マウントポイント「/mnt」にマウントします。

```
# beadm mount be01 /mnt
# beadm list
BE Name      Flags Mountpoint Space  Policy Created
-----
be01         -    /mnt      95.14M static 2019-03-22 14:43
solaris      NR    /         4.15G static 2019-03-01 15:16
```

## 3) グループパッケージ「solaris-desktop」をインストールします。

```
# pkg -R /mnt install solaris-desktop
Packages to install: 285
Services to change: 16

DOWNLOAD                                PKGS      FILES    XFER (MB)  SPEED
Completed                                285/285   56307/56307 810.1/810.1 2.5M/s

PHASE                                     ITEMS
Installing new actions                    68854/68854
Updating package state database           Done
Updating package cache                    0/0
Updating image state                      Done
Creating fast lookup database            Done
Updating package cache                    1/1
```

▶ グループパッケージ「solaris-desktop」には、自動スナップショットのパッケージが含まれています。

## 4) 「solaris-desktop」パッケージを確認します。

インストールが完了すると、IFO パラメータに「i--」と表示されます。

```
# pkg -R /mnt list solaris-desktop
NAME (PUBLISHER)                                VERSION                                IFO
group/system/solaris-desktop                    11.4-11.4.6.0.1.4.0                  i--
```

## 5) BE をアクティベートして切り替えたあと、再起動します。

be01 の Active の表示が「R」と表示されることを確認します。

```
# bootadm update-archive -R /mnt
# beadm umount be01
# beadm list
BE Name      Flags Mountpoint Space  Policy Created
-----
be01         -      /          2.49G static 2019-03-22 14:43
solaris      NR     /          4.15G static 2019-03-01 15:16
# beadm activate be01
# beadm list
BE Name      Flags Mountpoint Space  Policy Created
-----
be01         R      -          6.55G static 2019-03-22 14:43
solaris      N     /          96.10M static 2019-03-01 15:16
# shutdown -y -g0 -i6
```

▶ 次回の起動時に、BE が切り替わります。

## 6) 再起動後、BE が切り替わっていることを確認します。

```
# beadm list
BE Name      Flags Mountpoint Space  Policy Created
-----
be01         NR     /          6.57G static 2019-03-22 14:43
solaris      -     -          574.96M static 2019-03-01 15:16
```

## 7) タイムスライダーサービスが存在することを確認します。

```
# svcs time-slider
STATE      STIME      FMRI
disabled   14:55:28  svc:/application/time-slider:default
```

## 8) 自動スナップショットサービスが存在することを確認します。

```
# svcs -a | grep auto-snapshot
disabled   14:55:30  svc:/system/filesystem/zfs/auto-snapshot:daily
disabled   14:55:30  svc:/system/filesystem/zfs/auto-snapshot:frequent
disabled   14:55:30  svc:/system/filesystem/zfs/auto-snapshot:hourly
disabled   14:55:30  svc:/system/filesystem/zfs/auto-snapshot:monthly
disabled   14:55:30  svc:/system/filesystem/zfs/auto-snapshot:weekly
```

## 9) 自動スナップショットを有効化します。

ここでは例として、ファイルシステム「rz2pool/data1」の自動スナップショットを有効化します。

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

### 10) 自動スナップショットサービスを有効化します。

自動スナップショットサービスが「disabled」の状態の場合、タイムスライダーサービスが起動するタイミングで「offline」から「online」の状態になります。

ここでは例として、「frequent」と「hourly」を有効化します。

```
# svcadm enable svc:/system/filesystem/zfs/auto-snapshot:frequent
# svcadm enable svc:/system/filesystem/zfs/auto-snapshot:hourly
# svcs -a | grep auto-snapshot
disabled      14:55:30 svc:/system/filesystem/zfs/auto-snapshot:daily
disabled      14:55:30 svc:/system/filesystem/zfs/auto-snapshot:monthly
disabled      14:55:30 svc:/system/filesystem/zfs/auto-snapshot:weekly
offline       15:01:46 svc:/system/filesystem/zfs/auto-snapshot:frequent
offline       15:01:51 svc:/system/filesystem/zfs/auto-snapshot:hourly
```

- ▶ 「frequent」を有効にすると、15 分ごとにスナップショットが自動作成されます。
- ▶ 「hourly」を有効にすると、1 時間ごとにスナップショットが自動作成されます。

### 11) タイムスライダーサービスを起動します。

```
# svcadm enable time-slider
# svcs time-slider
STATE      STIME      FMRI
online     15:02:18  svc:/application/time-slider:default
```

### 12) タイムスライダーサービスの起動後、自動スナップショットサービスを確認します。

「offline」状態のサービスのみ、「online」になります。

```
# svcs -a | grep auto-snapshot
disabled      14:55:30 svc:/system/filesystem/zfs/auto-snapshot:daily
disabled      14:55:30 svc:/system/filesystem/zfs/auto-snapshot:monthly
disabled      14:55:30 svc:/system/filesystem/zfs/auto-snapshot:weekly
online        15:01:46 svc:/system/filesystem/zfs/auto-snapshot:frequent
online        15:01:51 svc:/system/filesystem/zfs/auto-snapshot:hourly
```

### 13) 自動で作成されるスナップショットを確認します。

有効にした自動スナップショットサービスの実行間隔で、スナップショットが作成されていることを確認します。

```
# zfs list -r -t snapshot rz2pool
NAME                                                    USED  AVAIL  REFER  MOUNTPOINT
rz2pool/data1@zfs-auto-snap_hourly-2019-05-08-20h08    0      -    916M  -
rz2pool/data1@zfs-auto-snap_frequent-2019-05-08-20h38  0      -    916M  -
rz2pool/data1@before                                    1023   -    916M  -
rz2pool/data2@snap1                                    20.0K   -    31.0K  -
rz2pool/data2@snap2                                    0      -    33.0K  -
```

- ▶ 常に最新のスナップショットのみ保持されます。

## 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。zfs send コマンドでスナップショットからストリームを作成し、ファイルへ送信することで、ファイルシステムやボリュームをバックアップできます。

また、zfs send コマンドで作成・送信されたストリームを zfs receive コマンドで受信することで、ファイルシステムやボリュームをリストアすることができます。

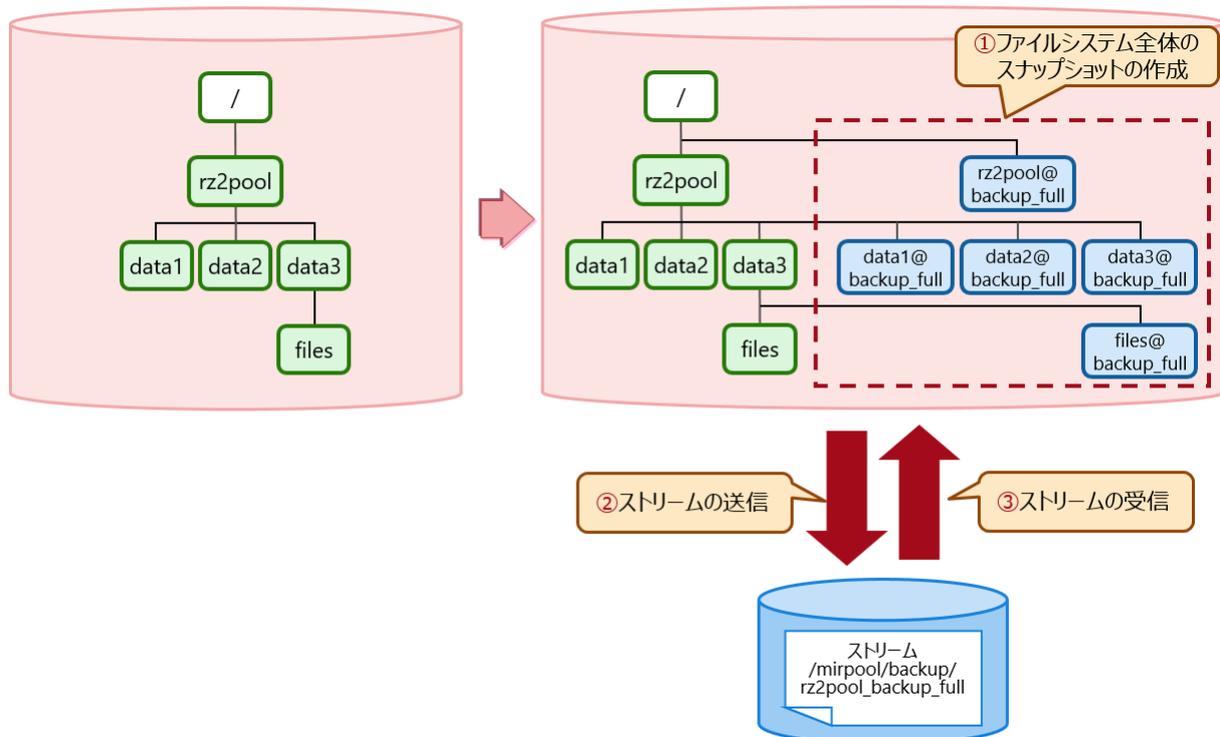
本書では、ユーザー領域のバックアップ／リストアの方法を説明します。システム領域のバックアップ／リストアについては、以下を参照してください。

- 「Oracle Solaris 11 を使ってみよう」

<https://www.fujitsu.com/jp/sparc/technical/document/solaris/#os>

### 4.1. 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。



#### 4.1.1. 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を作成します。

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

- ▶ `-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。

##### 2) スナップショットが作成されたことを確認します。

```
# zfs list -r -t snapshot rz2pool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
rz2pool@backup_full	0	-	34.0K	-
rz2pool/data1@zfs-auto-snap_frequent-2019-05-13-17h43	18.0K	-	916M	-
rz2pool/data1@zfs-auto-snap_frequent-2019-05-15-17h58	1023	-	916M	-
rz2pool/data1@before	1023	-	916M	-
rz2pool/data1@backup_full	0	-	916M	-
rz2pool/data2@snap1	20.0K	-	31.0K	-
rz2pool/data2@snap2	19.0K	-	33.0K	-
rz2pool/data2@backup_full	0	-	33.0K	-
rz2pool/data3@backup_full	0	-	33.0K	-
rz2pool/data3/files@backup_full	0	-	31.0K	-

##### 3) ストリームの送信先を作成します。

ストリームの送信先は任意です。本書では、送信先に `/mirpool/backup` を作成します。

```
# zfs create mirpool/backup
```

```
# zfs list -r mirpool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
mirpool	313K	29.7G	33K	/mirpool
mirpool/backup	31K	29.7G	31K	/mirpool/backup

#### 4) スナップショットからストリームを作成し、送信します。

```
# zfs send -Rv rz2pool@backup_full > /mirpool/backup/rz2pool_backup_full
  estimating full stream rz2pool@backup_full (size = 41.9K)
  estimating full stream rz2pool/data1@zfs-auto-snap_frequent-2019-05-13-17h43 (size =
920M)
  estimating incremental rz2pool/data1@zfs-auto-snap_frequent-2019-05-13-17h43 to
rz2pool/data1@zfs-auto-snap_frequent-2019-05-15-17h58 (size = 9.03K)
  estimating incremental rz2pool/data1@zfs-auto-snap_frequent-2019-05-15-17h58 to
rz2pool/data1@before (size = 513)
  estimating incremental rz2pool/data1@before to rz2pool/data1@backup_full (size =
513)
  estimating full stream rz2pool/data2@snap1 (size = 41.1K)
  estimating incremental rz2pool/data2@snap1 to rz2pool/data2@snap2 (size = 11.0K)
  estimating incremental rz2pool/data2@snap2 to rz2pool/data2@backup_full (size =
9.53K)
  estimating full stream rz2pool/data3@backup_full (size = 41.3K)
  estimating full stream rz2pool/data3/files@backup_full (size = 39.6K)
estimated total stream size: 920M
```

▶ `-R` オプションで指定し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。

#### 《参考》 ストリームを1つのファイルに保存する

1つのファイルに保存する場合は、`gzip` コマンドで圧縮します。

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

#### 4.1.2. リストア

ここでは例として、ファイルシステム「rz2pool」を削除し、バックアップからリストアできることを確認します。

##### Point

「[《参考》保護されているスナップショットの削除](#)」を実施していない場合は、以下のコマンドを実行し、スナップショットの保護を解除してください。

```
# zfs release mytag rz2pool/data1@hold_test
```

##### 1) ファイルシステムを削除します。

```
# zfs destroy -r rz2pool
# zfs list -r rz2pool
NAME      USED  AVAIL  REFER  MOUNTPOINT
rz2pool   918M  28.6G  31.0K  /rz2pool
```

▶ `-r` オプションを指定し、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を同時に削除します。

##### 2) ストリームを受信します。

```
# zfs receive -F -d -v rz2pool < /mirpool/backup/rz2pool_backup_full
receiving full stream of rz2pool@backup_full into rz2pool@backup_full
received 41.3KB stream in 3 seconds (13.8KB/sec)
receiving full stream of rz2pool/data1@zfs-auto-snap_frequent-2019-05-13-17h43 into
rz2pool/data1@zfs-auto-snap_frequent-2019-05-13-17h43
received 917MB stream in 70 seconds (13.1MB/sec)
receiving incremental stream of rz2pool/data1@zfs-auto-snap_frequent-2019-05-15-17h58
into rz2pool/data1@zfs-auto-snap_frequent-2019-05-15-17h58
received 2.30KB stream in 2 seconds (1.15KB/sec)
receiving incremental stream of rz2pool/data1@before into rz2pool/data1@before
received 248B stream in 1 seconds (248B/sec)
receiving incremental stream of rz2pool/data1@backup_full into
rz2pool/data1@backup_full
received 248B stream in 3 seconds (82B/sec)
receiving full stream of rz2pool/data2@snap1 into rz2pool/data2@snap1
received 41.1KB stream in 1 seconds (41.1KB/sec)
receiving incremental stream of rz2pool/data2@snap2 into rz2pool/data2@snap2
received 5.49KB stream in 1 seconds (5.49KB/sec)
receiving incremental stream of rz2pool/data2@backup_full into
rz2pool/data2@backup_full
received 3.55KB stream in 2 seconds (1.77KB/sec)
receiving full stream of rz2pool/data3@backup_full into rz2pool/data3@backup_full
received 41.5KB stream in 2 seconds (20.7KB/sec)
receiving full stream of rz2pool/data3/files@backup_full into
rz2pool/data3/files@backup_full
received 39.6KB stream in 1 seconds (39.6KB/sec)
```

- ▶ -F オプションを使用すると、ファイルシステムが最新のスナップショットから強制的にロールバックされます。
- ▶ -d オプションを使用すると、送信側で使用している名前でスナップショットを復元できます。-d オプションを指定せずに、-R オプションを指定して作成したストリームを受信すると、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。
- ▶ -v オプションを使用すると、ストリームおよび受信操作の所要時間に関する詳細情報を出力します。
- ▶ receive コマンドは、短縮形の recv を使用できます。

### 《参考》 gzip コマンドで圧縮されたストリームを受信する

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

### 3) ファイルシステムが復元されていることを確認します。

```
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool              918M  28.6G  34.0K  /rz2pool
rz2pool/data1        916M  28.6G   916M  /rz2pool/data1
rz2pool/data2        71.6K  28.6G  32.6K  /rz2pool/data2
rz2pool/data3        63.9K  3.00G  33.0K  /zfs/fs1
rz2pool/data3/files  31.0K  3.00G  31.0K  /zfs/fs1/files
```

### 4) スナップショットを削除します。

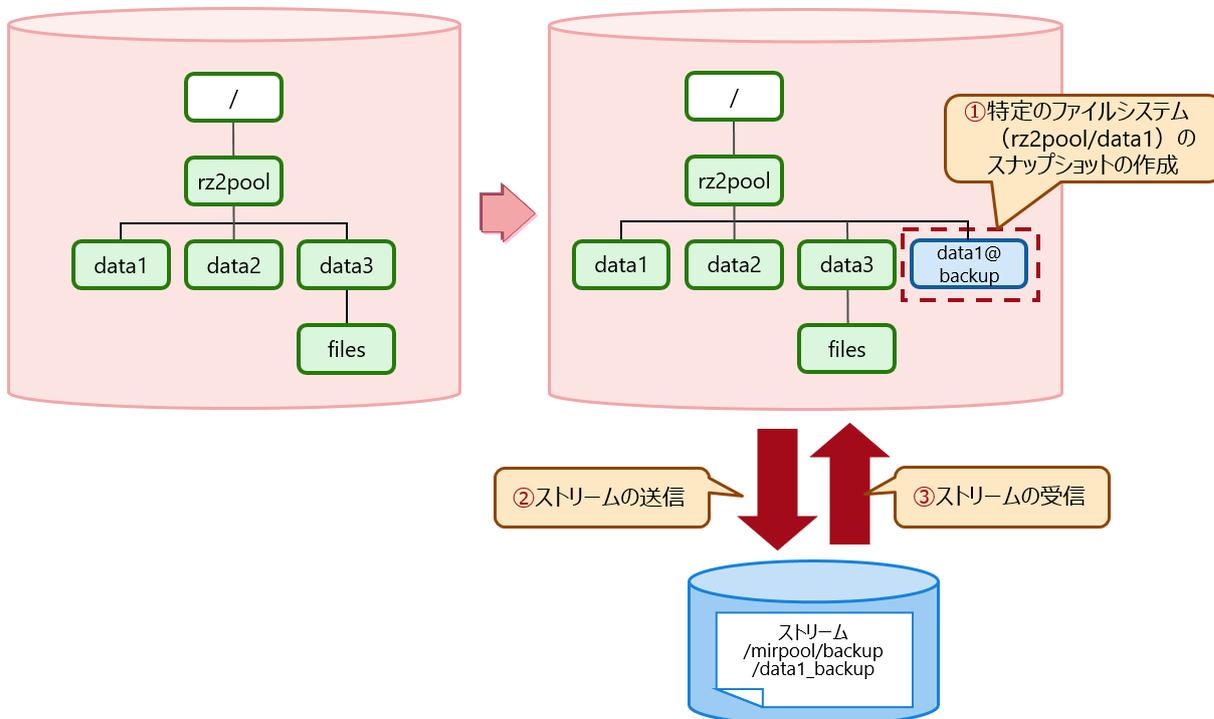
```
# zfs destroy -r rz2pool@backup_full
```

### 5) スナップショットが削除されていることを確認します。

```
# zfs list -r -t snapshot rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool/data1@zfs-auto-snap_frequent-2019-05-13-17h43  18.0K  -    916M  -
rz2pool/data1@zfs-auto-snap_frequent-2019-05-15-17h58  1023   -    916M  -
rz2pool/data1@before  1023   -    916M  -
rz2pool/data2@snap1  20.0K  -   31.0K  -
rz2pool/data2@snap2  19.0K  -   32.6K  -
```

## 4.2. 特定のファイルシステムのバックアップ／リストア

ストレージプール内の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。



### 4.2.1. バックアップ

- 1) 特定のファイルシステムのスナップショットを作成します。

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

- 2) スナップショットが作成されたことを確認します。

```
# zfs list -r -t snapshot rz2pool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
rz2pool/data1@zfs-auto-snap_frequent-2019-05-13-17h43	18.0K	-	916M	-
rz2pool/data1@zfs-auto-snap_frequent-2019-05-15-17h58	1023	-	916M	-
rz2pool/data1@before	1023	-	916M	-
<b>rz2pool/data1@backup</b>	0	-	916M	-
rz2pool/data2@snap1	20.0K	-	31.0K	-
rz2pool/data2@snap2	19.0K	-	32.6K	-

- 3) スナップショットからストリームを作成し、送信します。

```
# zfs send -v rz2pool/data1@backup > /mirpool/backup/data1_backup
  estimating full stream rz2pool/data1@backup (size = 920M)
  estimated total stream size: 920M
```

#### 4.2.2. リストア

ここでは例として、ファイルシステム「rz2pool/data1」を削除し、バックアップからリストアできることを確認します。

##### 1) ファイルシステムを削除します。

「rz2pool/data1」が表示されないことを確認します。

```
# zfs destroy rz2pool/data1
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool              1.07M 29.3G 32.0K  /rz2pool
rz2pool/data2        71.6K 29.3G 32.6K  /rz2pool/data2
rz2pool/data3        63.9K 3.00G 33.0K  /zfs/fs1
rz2pool/data3/files  31.0K 3.00G 31.0K  /zfs/fs1/files
```

##### 2) ストリームを受信します。

```
# zfs receive -v rz2pool/data1 < /mirpool/backup/data1_backup
receiving full stream of rz2pool/data1@backup into rz2pool/data1@backup
received 917MB stream in 50 seconds (18.3MB/sec)
```

##### 3) ファイルシステムが復元されていることを確認します。

```
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool              917M 28.6G 33.0K  /rz2pool
rz2pool/data1      916M 28.6G  916M  /rz2pool/data1
rz2pool/data2        71.6K 28.6G 32.6K  /rz2pool/data2
rz2pool/data3        63.9K 3.00G 33.0K  /zfs/fs1
rz2pool/data3/files  31.0K 3.00G 31.0K  /zfs/fs1/files
```

##### 4) スナップショットを削除します。

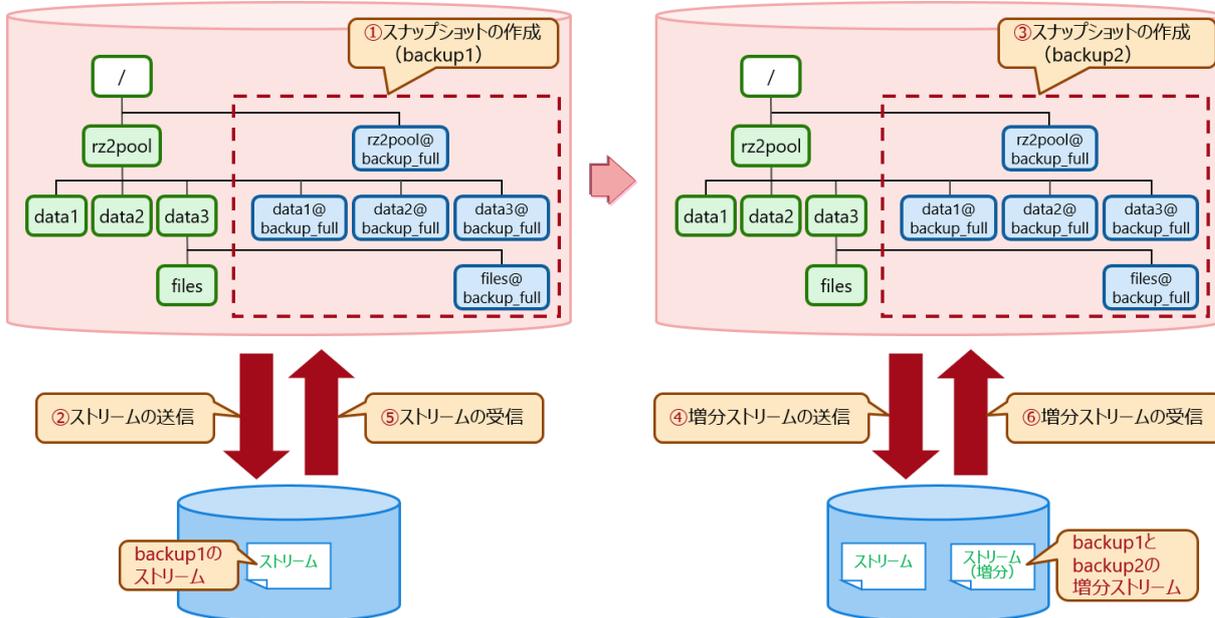
```
# zfs destroy rz2pool/data1@backup
```

##### 5) スナップショットが削除されていることを確認します。

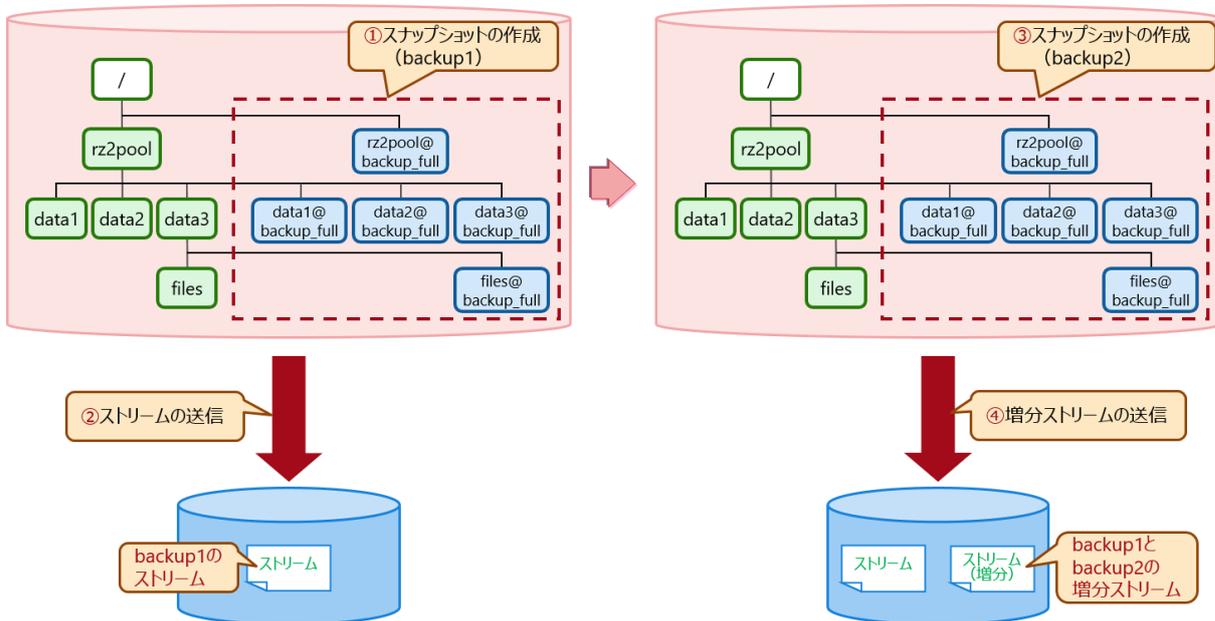
```
# zfs list -r -t snapshot rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool/data2@snap1 20.0K  -    31.0K  -
rz2pool/data2@snap2 19.0K  -    32.6K  -
```

### 4.3. 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／リストアします。



#### 4.3.1. 増分バックアップ



1) ファイルシステムのスナップショット「backup1」を作成します。

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

## 2) スナップショット「backup1」が作成されたことを確認します。

```
# zfs list -r -t snapshot rz2pool
NAME                               USED  AVAIL  REFER  MOUNTPOINT
rz2pool@backup1                    0     -    33.0K  -
rz2pool/data1@backup1              0     -    916M  -
rz2pool/data2@snap1               20.0K  -    31.0K  -
rz2pool/data2@snap2               19.0K  -    32.6K  -
rz2pool/data2@backup1              0     -    32.6K  -
rz2pool/data3@backup1              0     -    33.0K  -
rz2pool/data3/files@backup1        0     -    31.0K  -
```

## 3) スナップショット「backup1」からストリームを作成し、送信します。

```
# zfs send -Rv rz2pool@backup1 > /mirpool/backup/rz2pool_backup1
estimating full stream rz2pool@backup1 (size = 41.3K)
estimating full stream rz2pool/data1@backup1 (size = 920M)
estimating full stream rz2pool/data3@backup1 (size = 41.3K)
estimating full stream rz2pool/data3/files@backup1 (size = 39.6K)
estimating full stream rz2pool/data2@snap1 (size = 41.1K)
estimating incremental rz2pool/data2@snap1 to rz2pool/data2@snap2 (size = 11.0K)
estimating incremental rz2pool/data2@snap2 to rz2pool/data2@backup1 (size = 9.53K)
estimated total stream size: 920M
```

## 4) 増分データを確認するため、ファイルを作成します。

```
# touch /rz2pool/file01.txt
# touch /rz2pool/data1/file02.txt
# touch /zfs/fs1/files/file03.txt
# rm /rz2pool/data1/sol-11_4-text-sparc.iso
# ls -l /rz2pool
total 7
drwxr-xr-x  2 root  root           3 May 20 16:25 data1
drwxr-xr-x  2 root  root           6 May  8 19:27 data2
-rw-r--r--  1 root  root           0 May 20 16:22 file01.txt
# ls -l /rz2pool/data1
total 1
-rw-r--r--  1 root  root           0 May 20 16:23 file02.txt
# ls -l /zfs/fs1/files
total 1
-rw-r--r--  1 root  root           0 May 20 16:23 file03.txt
```

## 5) ファイルシステムのスナップショット「backup2」を作成します。

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

## 6) スナップショット「backup2」が作成されたことを確認します。

```
# zfs list -r -t snapshot rz2pool
NAME                               USED  AVAIL  REFER  MOUNTPOINT
rz2pool@backup1                    19.0K  -    33.0K  -
rz2pool@backup2                      0     -    34.0K  -
rz2pool/data1@backup1               916M  -    916M  -
rz2pool/data1@backup2                  0     -    31.0K  -
rz2pool/data2@snap1                 20.0K  -    31.0K  -
rz2pool/data2@snap2                 19.0K  -    32.6K  -
rz2pool/data2@backup1                 0     -    32.6K  -
rz2pool/data2@backup2                  0     -    32.6K  -
rz2pool/data3@backup1                 0     -    33.0K  -
rz2pool/data3@backup2                  0     -    33.0K  -
rz2pool/data3/files@backup1          19.0K  -    31.0K  -
rz2pool/data3/files@backup2           0     -    31.0K  -
```

## 7) スナップショット「backup1」と「backup2」の増分ストリームを作成し、送信します。

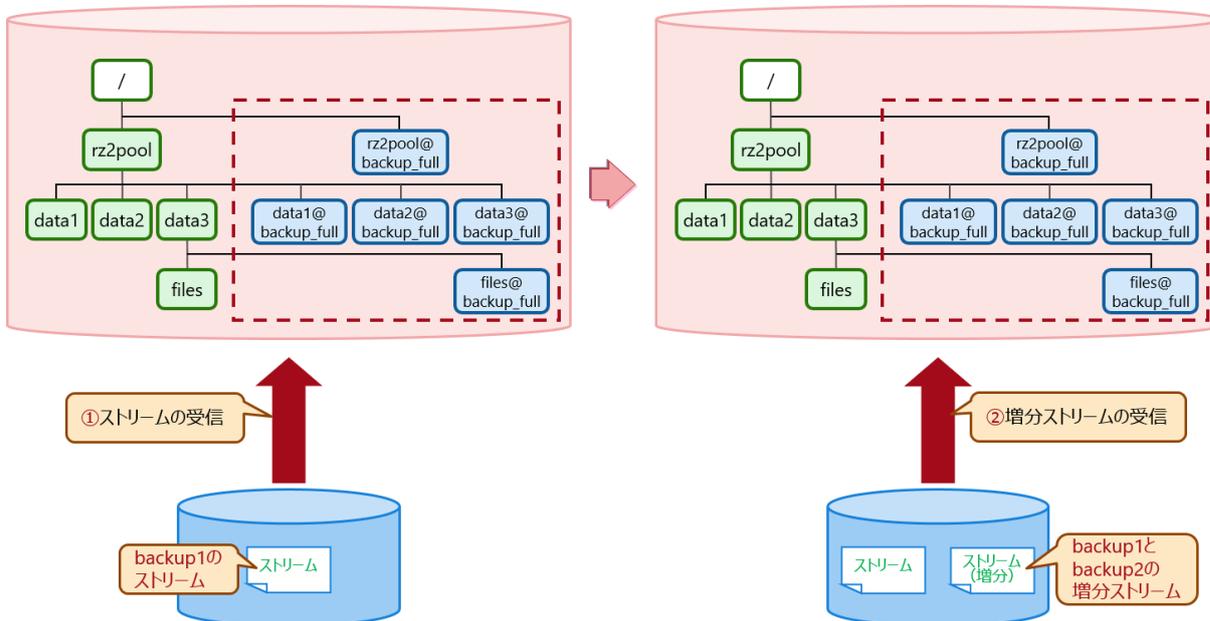
```
# zfs send -Rv -i rz2pool@backup1 rz2pool@backup2 > /mirpool/backup/rz2pool_backup2
  estimating incremental rz2pool@backup1 to rz2pool@backup2 (size = 10.4K)
  estimating incremental rz2pool/data1@backup1 to rz2pool/data1@backup2 (size = 10.0K)
  estimating incremental rz2pool/data3@backup1 to rz2pool/data3@backup2 (size = 0)
  estimating incremental rz2pool/data3/files@backup1 to rz2pool/data3/files@backup2
(size = 9.89K)
  estimating incremental rz2pool/data2@backup1 to rz2pool/data2@backup2 (size = 0)
estimated total stream size: 30.3K
```

- ▶ `-i` オプションを使用すると、最初の snapshot1 から 2 番目の snapshot2 への増分ストリームを生成します。増分ソースである最初のスナップショット「rz2pool@backup1」は、「@」の後ろの部分「backup1」のみの指定も可能です。

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

### 4.3.2. リストア

ここでは例として、ファイルシステム「rz2pool」を削除し、最初のバックアップ「backup1」と増分バックアップ「backup2」から段階的にリストアできることを確認します。



#### 1) ファイルシステムを削除します。

```
# zfs destroy -r rz2pool
# zfs list -r rz2pool
NAME      USED  AVAIL  REFER  MOUNTPOINT
rz2pool  866K  29.3G  31.0K  /rz2pool
```

#### 2) スナップショット「backup1」のストリームを受信します。

```
# zfs receive -F -d -v rz2pool < /mirpool/backup/rz2pool_backup1
receiving full stream of rz2pool@backup1 into rz2pool@backup1
received 41.2KB stream in 2 seconds (20.6KB/sec)
receiving full stream of rz2pool/data1@backup1 into rz2pool/data1@backup1
received 917MB stream in 52 seconds (17.6MB/sec)
receiving full stream of rz2pool/data3@backup1 into rz2pool/data3@backup1
received 41.5KB stream in 2 seconds (20.7KB/sec)
receiving full stream of rz2pool/data3/files@backup1 into rz2pool/data3/files@backup1
received 39.6KB stream in 1 seconds (39.6KB/sec)
receiving full stream of rz2pool/data2@snap1 into rz2pool/data2@snap1
received 41.1KB stream in 1 seconds (41.1KB/sec)
receiving incremental stream of rz2pool/data2@snap2 into rz2pool/data2@snap2
received 5.49KB stream in 2 seconds (2.75KB/sec)
receiving incremental stream of rz2pool/data2@backup1 into rz2pool/data2@backup1
received 3.55KB stream in 1 seconds (3.55KB/sec)
```

### 3) ファイルシステムが復元されていることを確認します。

この時点では、「sol-11\_4-text-sparc.iso」は復元されていますが、「file01.txt」～「file03.txt」は復元されていません。

```
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool             917M  28.6G  33.0K  /rz2pool
rz2pool/data1       916M  28.6G   916M  /rz2pool/data1
rz2pool/data2       71.6K  28.6G  32.6K  /rz2pool/data2
rz2pool/data3       63.9K  3.00G  33.0K  /zfs/fs1
rz2pool/data3/files 31.0K  3.00G  31.0K  /zfs/fs1/files
# ls -l /rz2pool
total 6
drwxr-xr-x  2 root   root           3 May  8 16:45 data1
drwxr-xr-x  2 root   root           6 May  8 19:27 data2
# ls -l /rz2pool/data1
total 1876307
-rw-r--r--  1 root   root    1018736640 May  8 16:46 sol-11_4-text-sparc.iso
# ls -l /zfs/fs1/files
total 0
```

### 4) スナップショット「backup1」と「backup2」の増分ストリームを受信します。

```
# zfs receive -F -d -v rz2pool < /mirpool/backup/rz2pool_backup2
receiving incremental stream of rz2pool@backup2 into rz2pool@backup2
received 3.75KB stream in 1 seconds (3.75KB/sec)
receiving incremental stream of rz2pool/data1@backup2 into rz2pool/data1@backup2
received 3.68KB stream in 1 seconds (3.68KB/sec)
receiving incremental stream of rz2pool/data3@backup2 into rz2pool/data3@backup2
received 160B stream in 2 seconds (80B/sec)
receiving incremental stream of rz2pool/data3/files@backup2 into
rz2pool/data3/files@backup2
received 2.95KB stream in 2 seconds (1.47KB/sec)
receiving incremental stream of rz2pool/data2@backup2 into rz2pool/data2@backup2
received 160B stream in 2 seconds (80B/sec)
```

5) 「file01.txt」～「file03.txt」が復元され、「sol-11\_4-text-sparc.iso」がなくなっていることを確認します。

```
# zfs list -r rz2pool
NAME                USED  AVAIL  REFER  MOUNTPOINT
rz2pool             918M  28.6G  34.0K  /rz2pool
rz2pool/data1       916M  28.6G  31.0K  /rz2pool/data1
rz2pool/data2       72.6K  28.6G  32.6K  /rz2pool/data2
rz2pool/data3       83.9K  3.00G  33.0K  /zfs/fs1
rz2pool/data3/files 50.0K  3.00G  31.0K  /zfs/fs1/files
# ls -l /rz2pool
total 7
drwxr-xr-x  2 root    root          3 May 20 16:25 data1
drwxr-xr-x  2 root    root          6 May  8 19:27 data2
-rw-r--r--  1 root    root           0 May 20 16:22 file01.txt
# ls -l /rz2pool/data1
total 1
-rw-r--r--  1 root    root           0 May 20 16:23 file02.txt
# ls -l /zfs/fs1/files
total 1
-rw-r--r--  1 root    root           0 May 20 16:23 file03.txt
```

#### 《参考》ファイルシステム「rz2pool」削除後に増分ストリームのみ受信した場合

最新のスナップショットが増分ソースと一致しないことを示すエラーが表示され、リストアに失敗します。

```
# zfs receive -F -d -v rz2pool < /mirpool/backup/rz2pool_backup2
receiving incremental stream of rz2pool@backup2 into rz2pool@backup2
cannot receive: most recent snapshot of rz2pool does not
match incremental source
```

#### 《参考》事前にファイルサイズを確認する

zfs send -nv オプションで、事前にファイルサイズを確認できます。

```
# zfs send -nv rz2pool@backup1
  estimating full stream rz2pool@backup1 (size = 41.3K)
estimated total stream size: 41.3K
```

▶ ファイルは作成されません。

**《参考》 ストリーム送信時の進捗表示**

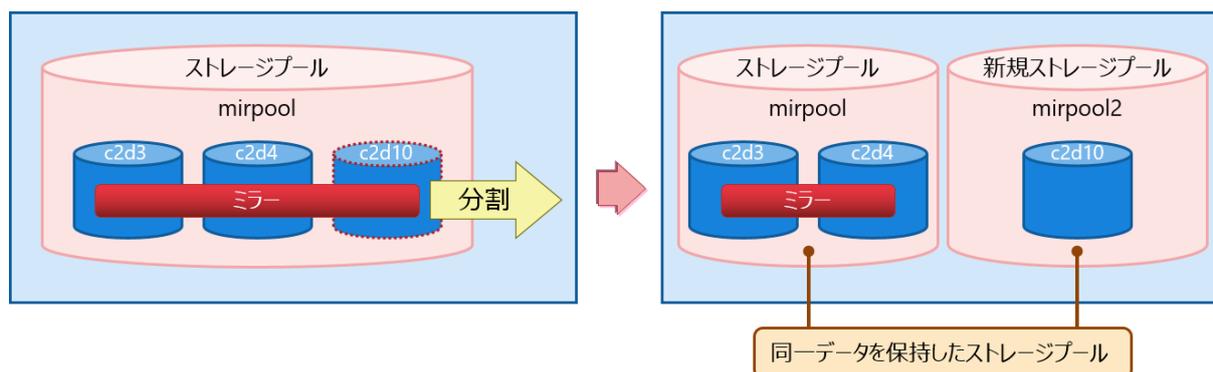
pv コマンドで、zfs send コマンドの進捗状況を確認します。

```
# zfs send -Rv rz2pool@backup1 | pv | gzip > /mirpool/backup/rz2pool_backup1.gz
estimating full stream rz2pool@backup1 (size = 41.3K)
estimating full stream rz2pool/data2@snap1 (size = 41.1K)
estimating incremental rz2pool/data2@snap1 to rz2pool/data2@snap2 (size = 11.0K)
estimating incremental rz2pool/data2@snap2 to rz2pool/data2@backup1 (size =
9.53K)
estimating full stream rz2pool/data3@backup1 0.00 B 0:00:01 [0.00 B/s] [<=>
(size = 41.3K) ]
estimating full stream rz2pool/data3/files@backup1 0.00 B 0:00:02 [0.00 B/s]
[(size = 39.6K) ]
estimating full stream rz2pool/data1@backup1 (size = 920M)
estimated total stream size: 920M
203MiB 0:00:26 [14.9MiB/s] [ <=> ]
```

## 5. ミラーディスクの分割

ミラーを構成するストレージプール内の一部のディスクを分割し、分割元と同じデータの新規ストレージプールを作成できます。分割したディスクをインポートすることで元のストレージプールと同じ構成で新規ストレージプールを作成できるため、バックアップとして利用できます。

※ ディスク分割後もデータの冗長性を保つには、3 本以上のミラーディスクに対してミラーディスクの分割を実施します。



### 1) ストレージプール「mirpool」の状態を確認します。

```
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: resilvered 130M in 6s with 0 errors on Wed May 8 15:25:44 2019

config:

    NAME      STATE    READ WRITE CKSUM
    mirpool   ONLINE   0     0     0
      mirror-0 ONLINE   0     0     0
        c2d3   ONLINE   0     0     0
        c2d4   ONLINE   0     0     0
        c2d10  ONLINE   0     0     0
    spares
      c2d11   AVAIL

errors: No known data errors
```

▶ mirpool は 3 面ミラー構成になっている状態です。

## 2) ファイルシステム「mirpool/dataset/test」を作成します。

```
# zfs list -r mirpool
NAME                USED  AVAIL  REFER  MOUNTPOINT
mirpool              1.57G 25.7G 1.57G  /mirpool
mirpool/backup       3.58G 25.7G 3.58G  /mirpool/backup
# zfs destroy mirpool/backup
# zfs create -p mirpool/dataset/test
```

## 3) ストレージプール「mirpool」のファイルシステムを確認します。

```
# zfs list -r mirpool
NAME                USED  AVAIL  REFER  MOUNTPOINT
mirpool              1.57G 27.7G 1.57G  /mirpool
mirpool/dataset      63K  27.7G  32K  /mirpool/dataset
mirpool/dataset/test 31K  27.7G  31K  /mirpool/dataset/test
```

## 4) ミラーディスクを分割します。

ストレージプール「mirpool」からディスク c2d10 を分割し、新規ストレージプール「mirpool2」を作成します。

```
# zpool split mirpool mirpool2 c2d10
```

- ▶ ディスクを指定しない場合は、ミラー構成の最後のディスクが分割されます。

## 5) ディスク分割後のストレージプール「mirpool」の状態を確認します。

ストレージプール「mirpool」のディスク c2d10 が分割されていることを確認します。

```
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: resilvered 130M in 6s with 0 errors on Wed May  8 15:25:44 2019

config:

    NAME        STATE        READ  WRITE  CKSUM
    mirpool     ONLINE      0     0     0
    mirror-0    ONLINE      0     0     0
    c2d3        ONLINE      0     0     0
    c2d4        ONLINE      0     0     0
    spares
    c2d11       AVAIL

errors: No known data errors
```

## 6) 分割で作成した新規ストレージプールを確認します。

分割されたストレージプールは、import 可能なストレージプールとして表示されます。

```
# zpool import
pool: mirpool2
   id: 12381640452406804919
  state: ONLINE
action: The pool can be imported using its name or numeric identifier.
config:

        mirpool2  ONLINE
           c2d10  ONLINE
```

## 7) 分割で作成した新規ストレージプール「mirpool2」をインポートします。

```
# zpool import mirpool2
```

## 8) 新規ストレージプールの状態を確認します。

ストレージプール「mirpool2」のデバイス構成を確認し、「mirpool2」がインポートされていることを確認します。

```
# zpool status mirpool2
pool: mirpool2
  state: ONLINE
   scan: scrub repaired 0 in 1s with 0 errors on Tue Mar 19 16:21:16 2019

config:

        NAME      STATE      READ WRITE CKSUM
        mirpool2  ONLINE      0     0     0
           c2d10  ONLINE      0     0     0

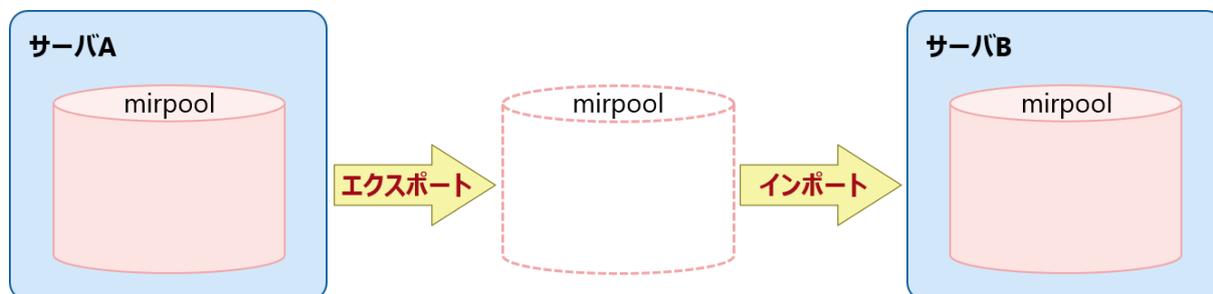
errors: No known data errors
```

## 9) 新規ストレージプール「mirpool2」のファイルシステムを確認します。

[手順 3](#))で確認したストレージプール「mirpool」のファイルシステム構成と同一であることを確認します。

```
# zfs list -r mirpool2
NAME                                USED  AVAIL  REFER  MOUNTPOINT
mirpool2                            1.57G 27.7G  1.57G  /mirpool2
mirpool2/dataset                    63K   27.7G   32K   /mirpool2/dataset
mirpool2/dataset/test                31K   27.7G   31K   /mirpool2/dataset/test
```

## 6. ストレージプールの移行



### 6.1. エクスポート(export)

ストレージプールをエクスポートし、別のサーバにインポートできます。この操作により、ストレージプールに関するすべての情報を別のサーバに移行できます。

- 1) ストレージプール「mirpool」をエクスポートします。

```
# zpool export mirpool
```

- 2) エクスポート状態を確認します。

エクスポートされたストレージプールは、システムから認識されなくなります。

```
# zpool list mirpool  
cannot open 'mirpool': no such pool
```

- ▶ ストレージプールの中にマウントされたファイルシステムがある場合は、マウントを解除してからエクスポートを実行します。マウントの解除に失敗した場合は、`-f` オプションを使用して強制的にマウントを解除できます。ただし、ZFS ボリュームがストレージプール内で使用中の場合は、`-f` オプションを使用してもストレージプールをエクスポートできません。

## 6.2. インポート(import)

ストレージプールをエクスポート後に物理デバイスを取り外し、インポート先のサーバに接続します。ZFSでは、一部のデバイスだけで使用できる状態になることもあります。サーバ間でストレージプールを移動するときは、すべての物理デバイスを移動してください。

デバイス名は、元のサーバ上でのデバイス名である必要はありません。デバイス名が変更されても、ストレージプールを構成するデバイスとして自動的に検出されます。

各ストレージプールは、名前および一意の数値識別子で識別されます。インポート可能な複数のストレージプールが同じ名前の場合は、数値識別子で識別します。

### 1) インポート可能なストレージプールを確認します。

インポート可能なストレージプールが表示されます。

```
# 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
      c2d11
```

## 2) ストレージプール「mirpool」をインポートします。

```
# zpool import mirpool
# zpool status mirpool
pool: mirpool
state: ONLINE
  scan: resilvered 130M in 6s with 0 errors on Wed May  8 15:25:44 2019

config:

    NAME        STATE      READ WRITE CKSUM
    mirpool     ONLINE    0     0     0
      mirror-0  ONLINE    0     0     0
        c2d3    ONLINE    0     0     0
        c2d4    ONLINE    0     0     0
    spares
      c2d11     AVAIL

errors: No known data errors
# zpool import
no pools available to import
```

### 《参考》 別名を付けたストレージプールのインポート

インポートするストレージプール名が既存のストレージプール名と重複する場合は、別の名前でインポートできます。

ストレージプール「mirpool」を「example\_mirpool」としてインポートする場合のコマンド実行例を、以下に示します。

```
# zpool import mirpool example_mirpool
# zpool status mirpool
cannot open 'mirpool': no such pool
# zpool status example_mirpool
pool: example_mirpool
state: ONLINE
scan: resilvered 130M in 6s with 0 errors on Wed May 8 15:25:44 2019

config:

    NAME                STATE          READ WRITE CKSUM
    example_mirpool     ONLINE         0     0     0
      mirror-0          ONLINE         0     0     0
        c2d3             ONLINE         0     0     0
        c2d4             ONLINE         0     0     0
    spares
      c2d11              AVAIL

errors: No known data errors
```

- ▶ 明示的にエクスポートしていないストレージプールは、誤って別のシステムにインポートされることを防ぐため、インポートできません。明示的にエクスポートしていないストレージプールをインポートする場合は、`-f` オプションを使用します。

### 《参考》 ID で指定してストレージプールをインポート

```
# zpool import
pool: mirpool
id: 3833992751035069071
state: ONLINE
action: The pool can be imported using its name or numeric identifier.
config:
~(省略)~
# zpool import 3833992751035069071
# zpool status mirpool
pool: mirpool
state: ONLINE
scan: resilvered 77.5K in 2s with 0 errors on Mon Mar 18 18:31:12 2019
~(省略)~
```

## 7. ZFS データの完全性のチェック

ZFS には、UFS の fsck に相当する機能は存在しません。代わりに、「スクラブ」という機能が用意されています。スクラブは、ストレージプールに含まれるすべてのデータにアクセスすることで、すべてのブロックが読み取り可能であることを確認します。

### 1) スクラブを実行します。

rpool をスクラブします。

```
# zpool scrub rpool
```

### 2) スクラブの状態を表示します。

```
# zpool status -v rpool
pool: rpool
state: ONLINE
scan: scrub in progress since Wed May  8 20:31:59 2019
      2.55G scanned out of 17.9G at 41.2M/s, 6m21s to go
      0 repaired, 14.25% done
config:

      NAME      STATE      READ WRITE CKSUM
      rpool     ONLINE    0     0     0
      mirror-0  ONLINE    0     0     0
      c2d0s0    ONLINE    0     0     0
      c2d1s0    ONLINE    0     0     0
      c2d2s0    ONLINE    0     0     0

errors: No known data errors
```

- ▶ `-v` オプションを使用すると、詳細なデータエラー情報が表示され、前回のスクラブ完了後の、すべてのデータエラーが出力されます。

### 《参考》 スクラブを中止する

実行中のスクラブを中止するには、`-s` オプションを指定します。

```
# zpool scrub -s rpool
```

## 8. トラブルシューティング

### 8.1. ストレージプールの入出力統計情報

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

#### 1) ストレージプール「rz2pool」の統計情報を表示します。

以下の例では、統計情報を取得する間隔（秒）と回数を「5 5」で指定しています。

```
# zpool iostat rz2pool 5 5
```

pool	capacity		operations		bandwidth	
	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

- ▶ 間隔（秒）だけ指定して回数を指定しないと、指定した間隔で表示され続け、[Ctrl]+[C]キーを押すと停止します。間隔（秒）と回数を指定しない場合は、コマンド実行時の状況が表示されます。

#### 2) `-v` オプションを指定し、仮想デバイス全体のレイアウトと入出力統計を表示します。

以下の例では、統計情報を取得する間隔（秒）と回数を「5 5」で指定しています。

```
# zpool iostat -v rz2pool 5 5
```

pool	capacity		operations		bandwidth	
	alloc	free	read	write	read	write
rz2pool	11.5G	78.0G	0	35	0	2.80M
raidz2-0	11.5G	78.0G	0	35	0	2.80M
c2d7	-	-	0	30	0	2.80M
c2d8	-	-	0	30	0	2.80M
c2d9	-	-	0	30	0	2.80M

～(省略)～

- ▶ 間隔（秒）だけ指定して回数を指定しないと、指定した間隔で表示され続け、[Ctrl]+[C]キーを押すと停止します。間隔（秒）と回数を指定しない場合は、コマンド実行時の状況が表示されます。

## 8.2. ZFS 操作の監視

Oracle Solaris 11.3 以降から、ファイルシステムおよびストレージプールの操作を監視できます。

監視可能な操作は以下のとおりです。

ファイルシステムの操作 (zfs コマンド)	ストレージプールの操作 (zpool コマンド)
<ul style="list-style-type: none"> <li>ZFS データ送受信 (send/receive)</li> <li>スナップショット破棄 (destroy)</li> </ul>	<ul style="list-style-type: none"> <li>ファイルシステム検証 (scrub)</li> <li>プール再同期 (resilver)</li> </ul>

### 1) スクラブを実行します。

rpool をスクラブします。

```
# zpool scrub rpool
```

### 2) scrub 実行中のプールの進捗状況を表示します。

ファイルシステム検証 (scrub) の進捗状況を表示します。以下の例では、「-t scrub」で対象の操作を指定し、「5」で進捗状況を取得する間隔 (秒) を指定しています。

```
# zpool monitor -t scrub 5
POOL          PROVIDER  PCTDONE  TOTAL  SPEED  TIMELEFT
rpool         scrub     9.5      17.9G  62.6M  4m25s
rpool         scrub    11.1     17.9G  62.4M  4m21s
rpool         scrub    12.8     17.9G  62.3M  4m16s
rpool         scrub    14.1     17.9G  60.7M  4m19s
rpool         scrub    14.8     17.9G  56.9M  4m34s
:
```

- ▶ 進捗状況は、[Ctrl]+[C]キーを押すか、または該当の処理が完了すると停止します。間隔 (秒) を指定しない場合は、コマンド実行時の状況が表示されます。
- ▶ スクラブを中止する場合は、「[《参考》スクラブを中止する](#)」を実施してください。

### 8.3. ZFS コマンドの履歴表示

正常に実行された ZFS コマンド(zfs および zpool)は、自動的に記録されています。この機能により、ユーザーやサポート担当者は実行された ZFS コマンドを特定し、トラブルシューティングなどに役立てることができます。

zpool history コマンドを使用した例を、以下に示します。

```
# zpool history -il rz2pool
History for 'rz2pool':
2019-05-07.16:41:27 zpool create rz2pool raidz c2d7 c2d8 c2d9 [user root on
guest2:global]
2019-05-07.16:41:27 [internal pool create txg:5] pool spa 44; zfs spa 44; zpl 6; uts
guest2 5.11 11.4.6.4.0 sun4v [user root on guest2]
2019-05-07.16:41:27 [internal property set txg:5] $share2=3 dataset = 18 [user root on
guest2]
2019-05-07.16:56:25 [internal pool load txg:64] pool load rz2pool; active ub txg=63
ts=1557215137 hostid=0x84f91f25 [user root on guest2]
2019-05-07.17:27:25 [internal pool load txg:439] pool load rz2pool; active ub txg=438
ts=1557217614 hostid=0x84f91f25 [user root on guest2]
:
```

- ▶ -i オプションにより、診断目的の内部イベント情報を表示します。上記の例では、[internal～、で始まる情報です。
- ▶ -l オプションにより、ユーザー名、ホスト名、および操作が実行されたゾーン名を表示します。上記の例では、[user root on guest2]が表示された情報です。
- ▶ オプション、およびストレージプール名は省略可能です。オプションを省略した場合は、実行コマンドの履歴のみ表示されます。ストレージプール名を省略した場合は、すべてのストレージプールの履歴が表示されます。

### 8.4. その他の問題や障害の識別・回復・解消

その他、下記のような問題や障害を識別し、回復・解消することができます。

- 損傷したデバイスの修復
- 破壊されたファイルまたはディレクトリの修復
- ZFS ストレージプール全体の損傷の修復
- 起動できないシステムの修復

詳細は、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/pdf/E75197.pdf](https://docs.oracle.com/cd/E75431_01/pdf/E75197.pdf)

-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

## 改版履歴

改版日	版数	改版内容
2012年11月	1.0版	新規作成
2015年5月	1.1版	Oracle Solaris 11.2 に対応
2016年5月	1.2版	Oracle Solaris 11.3 に対応
2019年6月	2.0版	Oracle Solaris 11.4 に対応

