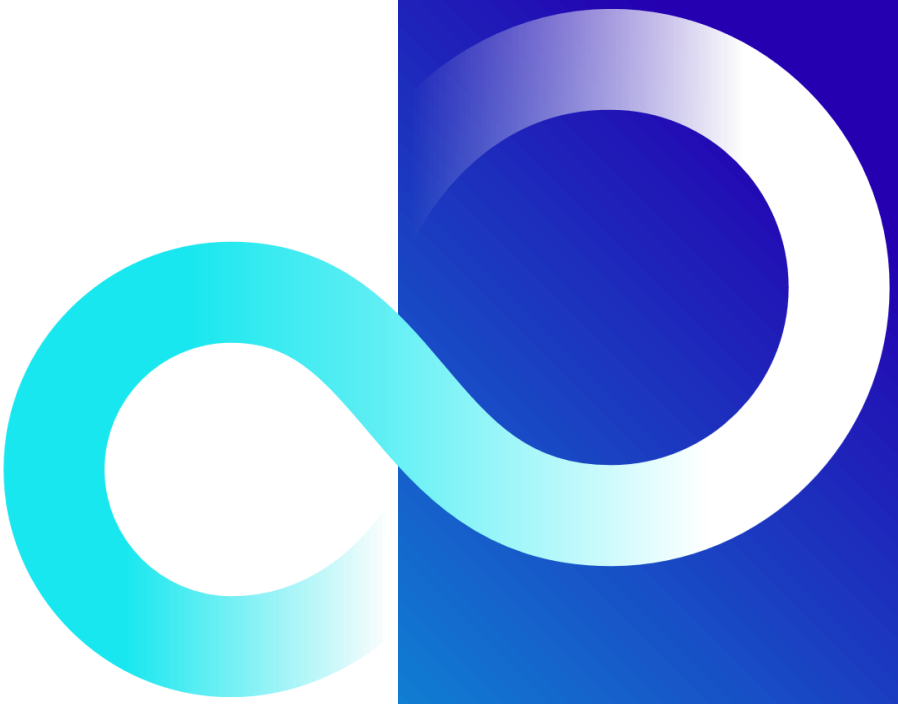
FUJITSU

ZFSデータ領域
バックアップガイド



2018年1月

第1.0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、SPARC64 ロゴおよびすべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している、同社の米国およびその他の国における商標または登録商標です。
- Oracle と Java は、Oracle Corporation およびその子会社、関連会社の米国およびその他の国における登録商標です。
- その他各種製品名は、各社の製品名称、商標または登録商標です。

はじめに

本書の内容

- SPARC/Solaris を使用される方を対象に、ZFS の機能を用いたユーザー領域（データ領域）の基本的なバックアップとバックアップ運用の自動化について、そのしくみと手順を説明しています。
- ZFS の詳細については、以下の URL をご参照ください。
  - Oracle Solaris 11 ZFS を使ってみよう
    http://www.fujitsu.com/jp/sparc-technical/document/solaris/index.html/#solaris11zfs
- Oracle Solaris 11 の詳細については、以下の URL をご参照ください。
  - Oracle Solaris 11.3 Information Library
    https://docs.oracle.com/cd/E62101_01/

対象読者

- Oracle Solaris、RAID、ZFS の基礎知識を有している方
- ZFS を使用したバックアップ運用を検討されている方

留意事項

- 本書は Oracle Solaris 11.3 の機能を基に作成しています。
- 本書に記載の設定値（ホスト名、IP アドレスなど）は参考例です。実際のシステム環境に応じて読み替えてください。
- 本書は、ユーザー領域のバックアップ、リストアを想定しています。システム領域のバックアップ、リストアは対象外です。

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

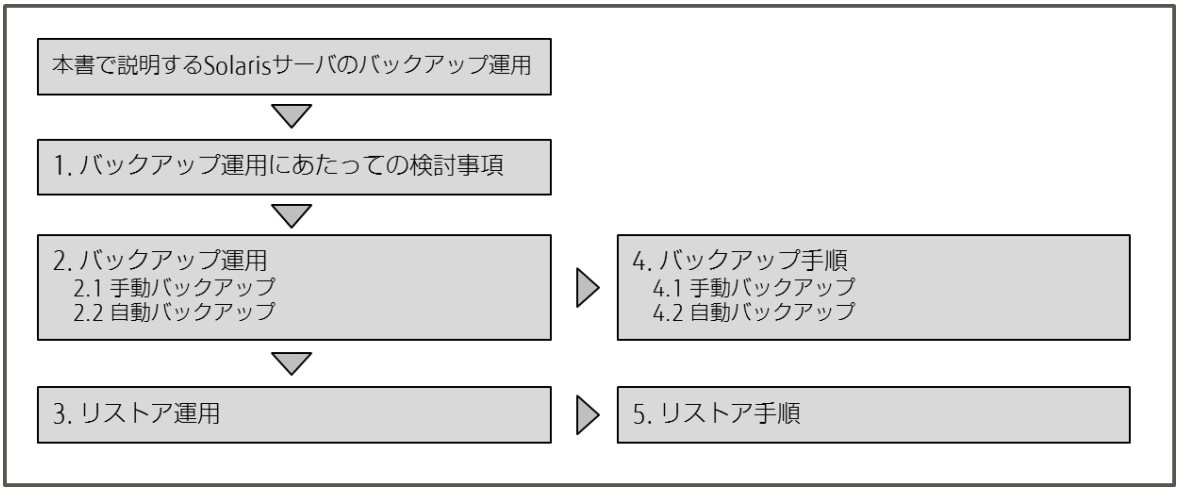
検証環境

- 本書に記載している操作の実行例には、以下の検証環境を使用しています。

		環境
サーバ機種		SPARC M10-1
	CPU	SPARC64 X+（3.2GHz） 1CPU（16core）
	メモリ	64 GB
	内蔵ストレージ	容量：600 GB 回転数：10000 rpm
OS		Oracle Solaris 11.3
SRU		SRU 16061
ESF		5.1

本書の全体のながれ

- 本書では、ZFS の機能で実現できる Solaris サーバのバックアップ／リストアの方式とその手順を紹介しています。本書で紹介するバックアップ／リストア方式が、システムのバックアップ要件を満たすかご確認のうえ、読み進めてください。
- 1 章～3 章でバックアップおよびリストアの方式を考えるにあたっての基礎知識や検討事項を説明し、4 章～5 章では 2 章～3 章で説明した運用の具体的な実行手順を説明しています。



目次

本書で説明する Solaris サーバのバックアップ運用

本書では、ZFS の機能を使用した Solaris サーバのバックアップ運用について説明します。
説明対象と内容を確認し、運用環境への適用が可能か判断のうえ、読み進めてください。

バックアップ対象

本書では、ユーザー領域（データ領域、ストレージプール）のバックアップ運用を説明の対象としています。

《参考》システム領域（ルートプール）のバックアップ運用

Oracle Solaris 11 のシステム領域のバックアップ／リストアには ZFS の機能を使用します。手順については、以下のドキュメントをご参照ください。

- 『Oracle Solaris 11 を使ってみよう（構築・運用手順書）』-「システムボリュームのバックアップ／リストア」
  http://www.fujitsu.com/jp/sparc-technical/document/solaris/index.html#os

バックアップ／リストアスクリプトを使用し、Solaris のコマンドを意識せずに簡単にシステムのリカバリーが行える、Oracle Solaris 11 システムリカバリーツールを使用することもできます。

- 『Oracle Solaris 11 システムリカバリーツール 解説資料』
  http://www.fujitsu.com/jp/sparc-technical/tools/system-recovery/index.html

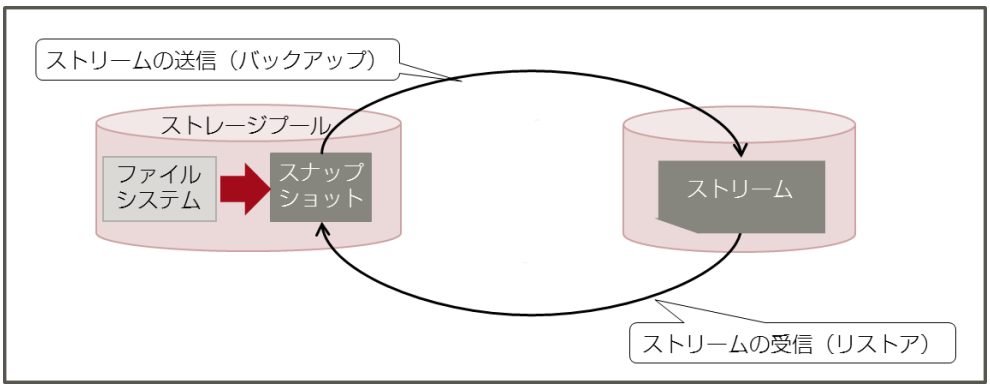
バックアップ要件と対応状況（ユーザー領域）

バックアップ運用に関する要件と、ZFS での対応可否を示します。
本書では、主にZFSとOSの標準機能などを組み合わせて実現できることについて、そのしくみと手順を説明します（☆マーク）。

要件		対応可否	備考
基本機能			
	バックアップ	○	ストレージプール内のスナップショットを作成し、スナップショットからストリームを作成して送信（☆） ☛ 「2.1.1 スナップショットの作成」、「2.1.2 バックアップの取得」
	差分バックアップ	○	2 つのスナップショットの差分ストリームを送信（☆） ☛ 「2.1.3 差分バックアップの取得」
	リストア	○	バックアップ時に送信したストリームを受信（☆） ☛ 「3 リストア運用」
対象（範囲）			
	ストレージプール	○	データセット（主にファイルシステム）を指定可能（☆） 階層がある場合、再帰的な指定が可能（☆） ☛ 「1.2 バックアップ範囲の検討」
	ファイルシステム	○	

要件		対応可否	備考
	ディスク	△	通常はストレージプール・ファイルシステム単位でバックアップを取得。ただし、単一のディスクでストレージプールを構成している場合は、OS の標準コマンド（dd コマンドなど）でバックアップ可能
	ファイル	○	OS の標準コマンド（cp コマンド、tar コマンドなど）を使用
バックアップデータ処理			
	重複排除	○	バックアップ時のオプションの指定によって、重複排除機能の利用が可能（☆） ☛ 「4.1.2 バックアップの取得」
	圧縮	○	ZFS のバックアップと OS 標準のファイル圧縮機能（gzip コマンドなど）を併用 ☛ 「4.1.2 バックアップの取得」
運用管理			
	スケジュール管理（定期的な自動実行）	○	タイムスライダーサービスでスナップショットを自動作成（☆） バックアップ実行のスクリプトを作成してタスクスケジューラ（cron）に登録することで、バックアップを自動実行（☆） ☛ 「1.1 バックアップ取得間隔および世代数の検討」 ☛ 「2 バックアップ運用」
	世代管理	○	世代別にバックアップファイルを保管し、古いバックアップを順次削除（☆） バックアップ実行のスクリプトに組み込んで、タスクスケジューラ（cron）に登録することで自動実行（☆） ☛ 「1.1 バックアップ取得間隔および世代数の検討」 ☛ 「2 バックアップ運用」
	進捗確認	○	バックアップ／リストア時のサブコマンドの指定によって、バックアップ／リストア実行中の進捗表示機能の利用が可能 ☛ 「4.1.2 バックアップの取得」 ☛ 「5.1.2 フルバックアップからのリストア」、「5.1.3 差分バックアップからのリストア」
	可用性	△	OS を停止させずに、スナップショットの作成とバックアップの取得が可能。ただし、リストア実行時には、対象のファイルシステムとその配下のファイルシステムへのアクセス不可
	操作性	△	コマンドラインでの操作
その他			
	サーバ負荷	△	バックアップ・リストア実行中に、若干の負荷が発生
	費用	○	OS の標準機能であるため追加費用なし

ZFS でのバックアップ／リストア（基本機能）



※　スナップショットとは、ファイルシステム（ボリュームを含む）の読み取り専用コピーのことです。スナップショットは瞬時に作成され、スナップショット作成直後はストレージプール内の領域を使用しません。

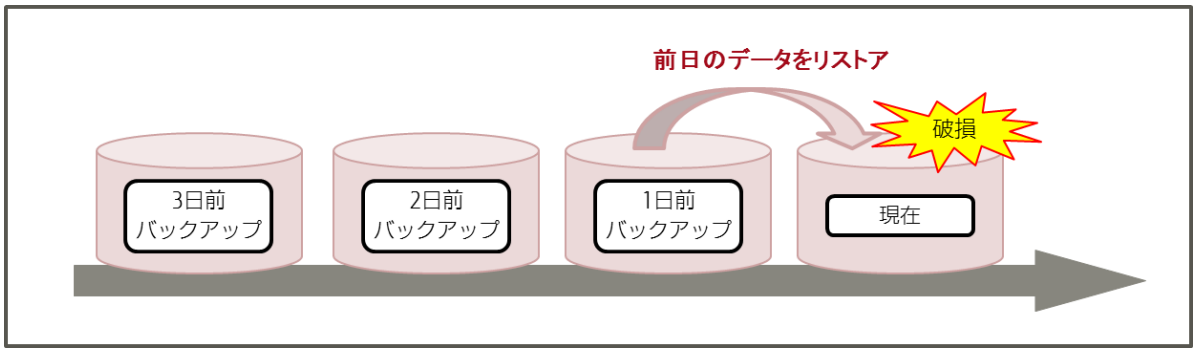
1. バックアップ運用にあたっての検討事項

Solaris サーバでは、OS 標準のファイルシステムである ZFS の機能を使用してバックアップ運用が実現できます。さらに、タスクスケジューラ（cron）との組み合わせにより、バックアップを自動化することも可能です。

1.1. バックアップ取得間隔および世代数の検討

目標とする保障期間や精度に応じてバックアップの取得間隔、世代数を検討します。データ損失時点から、より直近のデータをリストアする必要がある場合はスナップショット機能の利用も併せて検討します。
ZFS には、バックアップの世代管理やスケジュール管理（定期的な自動実行）の機能は備わっていません。
本書では、シェルスクリプトとタスクスケジューラを組み合わせて、バックアップの世代管理やスケジュール管理を実装する方法を紹介します。
→「2 バックアップ運用」、「4 バックアップ手順」

バックアップ運用イメージ



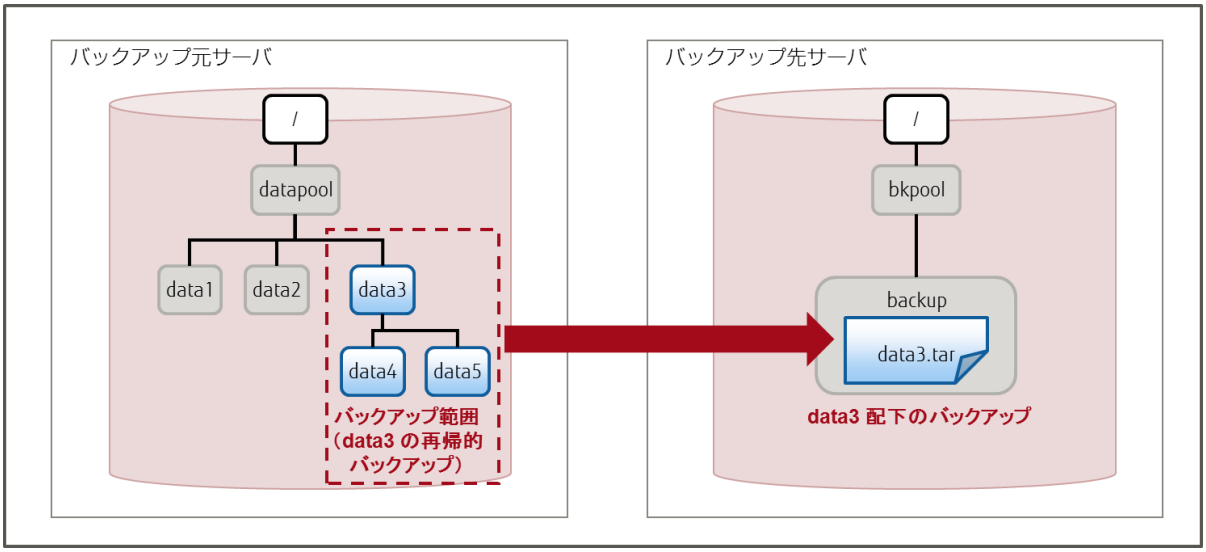
☛ 上記の例では 1 日間隔でバックアップを取得し、3 世代のバックアップを保持しています。

☛ 論理的なデータ損失（誤操作によるデータ削除など）の場合、自動スナップショットの機能を有効にすることで 15 分前までのロールバックが可能です。自動スナップショットの利用方法については「2.2.1 スナップショットの自動化」を参照してください。

1.2. バックアップ範囲の検討

ZFS では単一のファイルシステムのバックアップのほか、下位の階層のファイルシステムを含めた再帰的なバックアップが可能です。リストアはバックアップ時に取得したファイルシステム単位で行うため、リストアのタイミングが合わないファイルシステムが含まれないようバックアップ対象を設定してください。

ZFS にはファイル単位のバックアップ／リストア機能は備わっていません。ファイル単位の操作が必要な場合は、OS 標準コマンド（cp コマンド、tar コマンドなど）によるバックアップまたは別途バックアップソフトウェアの導入を検討してください。



- ファイルシステムのバックアップはアーカイブ形式（tar、cpio）などで出力できます。出力されるバックアップはストリーム形式のため、個別のファイルを対象としたバックアップやリストアはできません。
- サーバの故障に備えて、バックアップの配置先は外部装置にすることを推奨します。
- 後述するスナップショットの機能を用いることで、バックアップ元ファイルシステムに故障がない場合に限り、ファイル単位のロールバックが可能です。

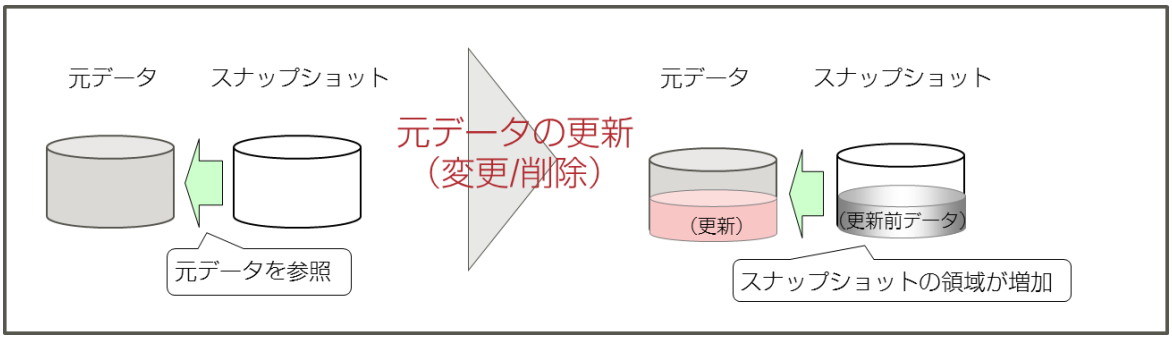
2. バックアップ運用

ZFS のバックアップ運用を手動で行う方法と、自動化する場合の実装方法についてそれぞれ説明します。

2.1. 手動バックアップ

2.1.1. スナップショットの作成

スナップショットを作成しておくことで、誤ってデータを変更したり削除したりした場合でも、即座に元の状態へ戻すことができます。スナップショット作成元のファイルシステム内のデータが更新（変更／削除）されると、スナップショットの領域に更新対象のファイルの元データが退避されます。



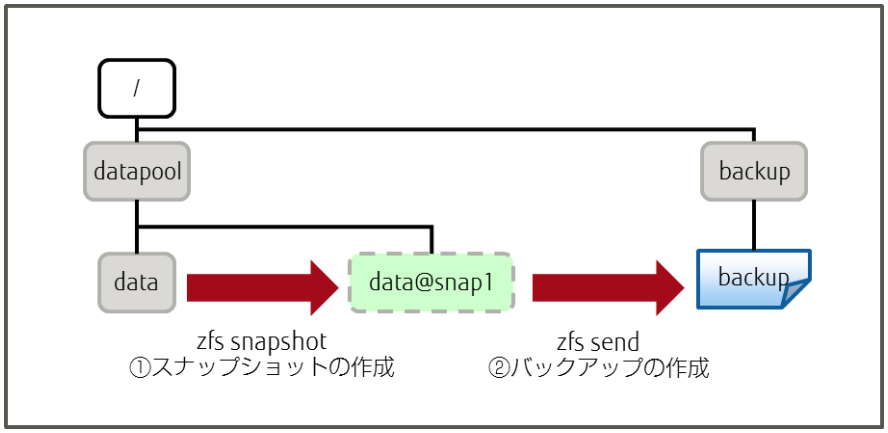
スナップショットの作成には zfs snapshot コマンドを使用します。スナップショットは対象のファイルシステムと同じ階層上に作成されます。スナップショットは同一ファイルシステム上に複数作成できますが、更新前データが退避されるのは最新のスナップショットに対してのみです。

☞ スナップショットは誤ったデータ変更などの論理故障からロールバックするための機能です。ディスクの物理故障などによるデータ損失に備えて、外部装置へのバックアップも併せて検討してください。

2.1.2. バックアップの取得

ZFS は、スナップショットからストリームの作成および送信を行うことでバックアップを取得します。なお、ストリームの作成時や送信時、OS を停止する必要はありません。
バックアップの取得には zfs send コマンドを使用します。バックアップはスナップショット作成時点のデータで取得されるため、必ずバックアップの直前にスナップショットを作成するようにしてください。

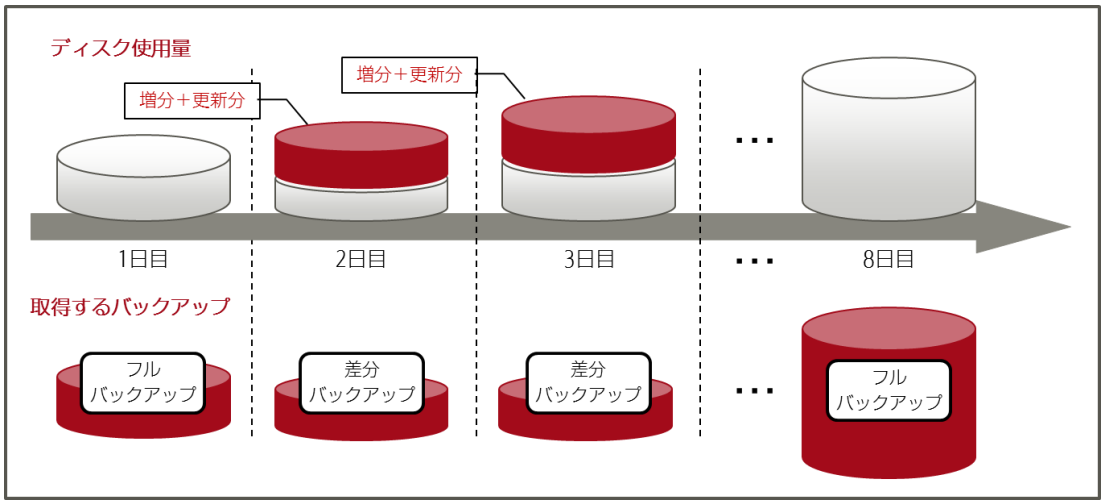
ZFS のバックアップのながれ（フルバックアップ）



2.1.3. 差分バックアップの取得

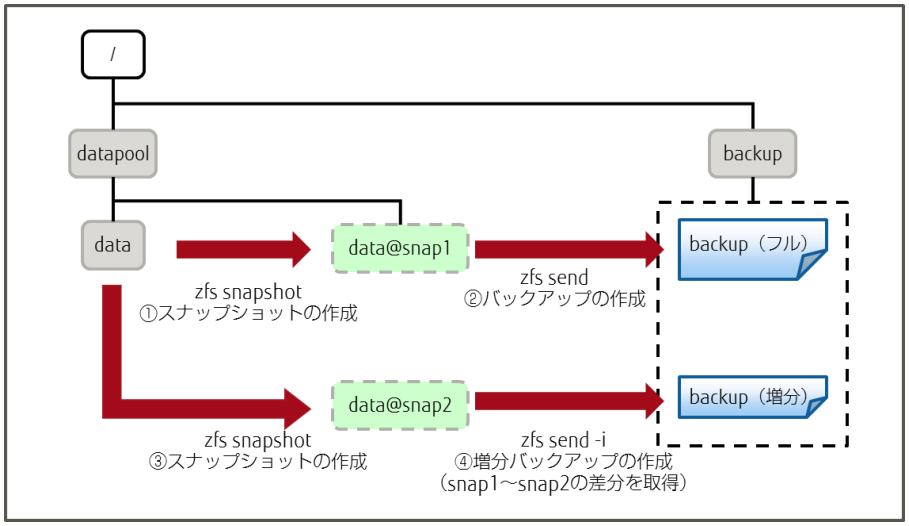
ZFS はフルバックアップのほか、差分バックアップが可能です。バックアップ対象のファイル数が多い場合やサイズが大きい場合、毎回フルバックアップを取得しているとバックアップが長時間化する可能性があります。バックアップ時間を短縮するため、例えば日次のバックアップを差分バックアップで行い、週次でフルバックアップを行うなどの運用方法を検討します。

差分バックアップの運用イメージ



ZFS の差分バックアップは、前回バックアップを取得した際のスナップショットと新規に作成したスナップショットの内容を比較し、更新された部分のみバックアップします。

ZFS のバックアップのながれ（差分バックアップ）



☞ 差分バックアップからリストアする場合は、最初にフルバックアップからリストアし、その後取得したすべての差分バックアップを適用する必要があります。詳細は「3.3 差分バックアップからのリストア」を参照してください。

2.2. 自動バックアップ

2.2.1. スナップショットの自動化

タイムスライダー（Time-slider）サービスを使用することで、一定期間ごとにスナップショットを自動的に作成できます。使用するディスク容量は増加しますが、バックアップを自動化したり、こまめにスナップショットを作成したりする際に便利です。タイムスライダーサービスを有効化したファイルシステムに対して、以下の作成タイミングと保持数を指定できます。

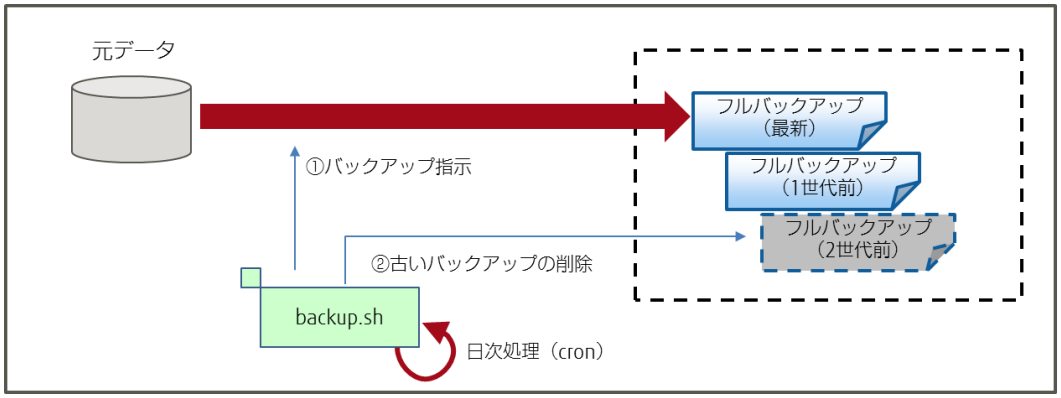
機能名	作成タイミング	保持可能な数（上限）
frequent snapshots	15 分ごと	4
hourly snapshots	毎時	24
daily snapshots	毎日	31
weekly snapshots	毎週	7
monthly snapshots	毎月	12

☛ タイムスライダーサービスでは、保持可能な数を超えたスナップショットは自動的に削除されます。

☛ 最初の作成タイミングはサービスを有効化した時点です。2 回目以降の作成タイミングで対象のファイルシステムへの更新が行われていなかった場合、スナップショットは作成されません。

2.2.2. フルバックアップ運用の自動化

ZFS では、バックアップのスケジュール管理および世代管理の自動化は、シェルスクリプトとタスクスケジューラの組み合わせで実装します。本書では、フルバックアップの取得および保存世代数を超えたバックアップの削除を行うシェルスクリプトを実装し、cron に登録して定期的に実行することでフルバックアップ運用の自動化を行います。

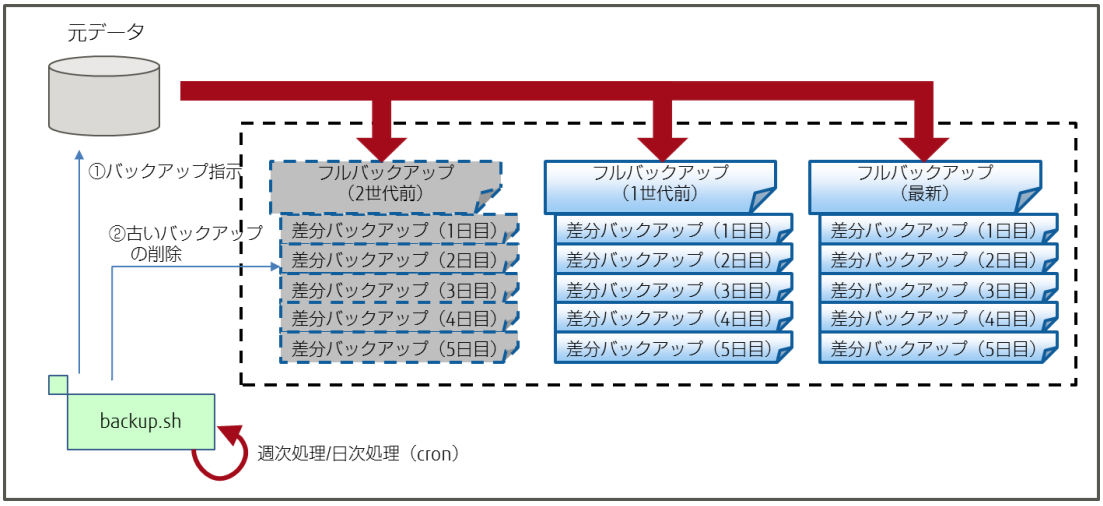


☛ 夜間バッチなどへの組み込みを行う場合は cron に登録せず、ご利用のジョブスケジューラへ登録してください。

☛ シェルスクリプトのサンプルは「4.2.2 フルバックアップ運用の自動化」を参照してください。

2.2.3. 差分バックアップ運用の自動化

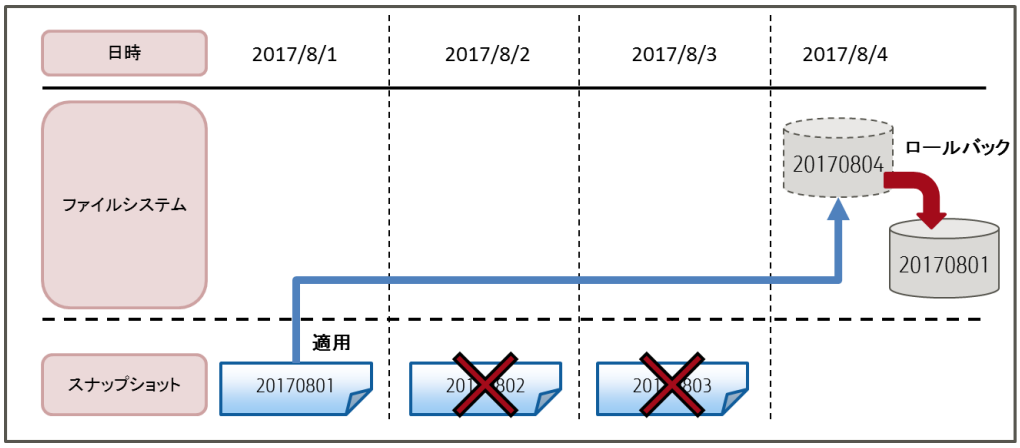
差分バックアップの自動化は、フルバックアップ運用の自動化と同様、シェルスクリプトとタスクスケジューラの組み合わせで実装します。本書では、フルバックアップおよび差分バックアップの取得および保存世代数を超えたバックアップの削除を行うシェルスクリプトを実装し、cron に登録して定期的に実行することで差分バックアップ運用の自動化を行います。



3. リストア運用

3.1. スナップショットからのロールバック

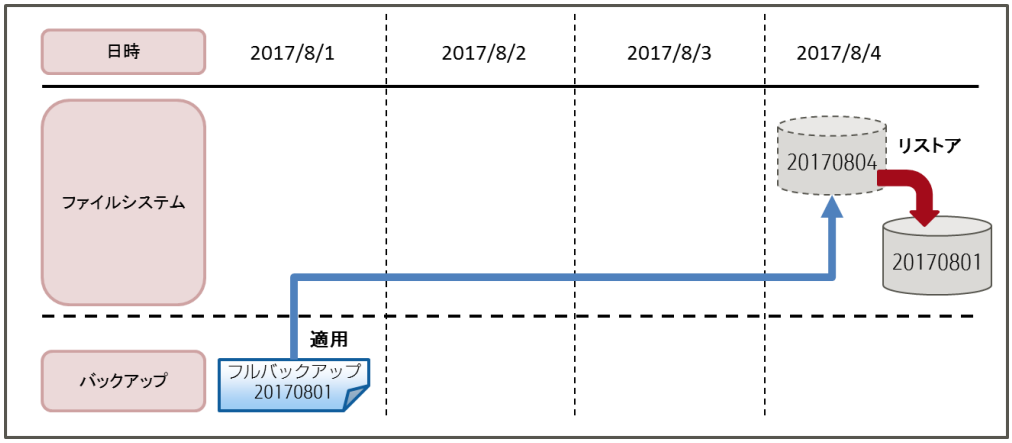
特定のスナップショット作成時よりあとに加えられたすべての変更を破棄し、ファイルシステムをスナップショット作成時の状態に戻します。ロールバック対象のファイルシステム上に複数回スナップショットを作成していた場合は、ロールバック時点までのスナップショットを削除する必要があります。



☛ ロールバックは、最新のスナップショットに対してのみ実行できます。上記の例では、「20170801」を最新のスナップショットにするために、ロールバック実行前に「20170802」と「20170803」のスナップショットを削除します。

3.2. フルバックアップからのリストア

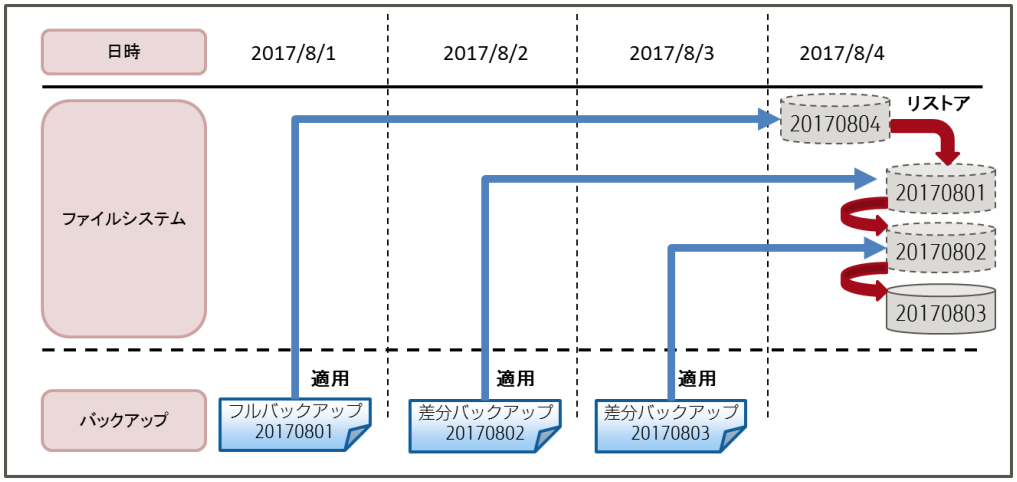
取得したフルバックアップからストリーム受信することで、リストアを行います。OS を停止する必要はありませんが、リストア中は対象のファイルシステムと下位のファイルシステムへのアクセスができません。リストア対象のファイルシステム上にスナップショットを作成していた場合は、すべてのスナップショットを手動で削除する必要があります。



☛ リストア対象のファイルシステム上にスナップショットが存在していると、リストアを実行できません。

3.3. 差分バックアップからのリストア

フルバックアップからリストア後、差分バックアップからのリストアを順次適用することでリストアを行います。OS を停止する必要はありませんが、リストア中は対象ファイルシステムと下位のファイルシステムへのアクセスができません。リストア対象のファイルシステム上にスナップショットを作成していた場合は、すべてのスナップショットを手動で削除する必要があります。



☛ リストア対象のファイルシステム上にスナップショットが存在していると、リストアを実行できません。

4. バックアップ手順

本章では「2 バックアップ運用」で説明した運用例の、具体的な設定方法を説明します。

4.1. 手動バックアップ

データが格納されたファイルシステムのバックアップを手動で取得します。本手順ではバックアップ先をほかのサーバ上の、NFS マウントした共有ディレクトリとします。

項目	内容
バックアップ対象のファイルシステム	data/share
バックアップ先	192.168.1.2:/backup ※バックアップサーバのディレクトリ
バックアップ先マウントポイント	/backup ※バックアップサーバを NFS マウントしたディレクトリ

4.1.1. スナップショットの作成

1) スナップショットの作成

i) バックアップ対象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 -r data/share@20170801
```

☛ “@”以降の文字列は作成するスナップショットの名称を指定します。

☛ バックアップ元のファイルシステムに下位のファイルシステムが存在する場合、-r オプションを指定すると再帰的にスナップショットを作成できます。

ii) スナップショットが作成されたことを確認します。

```
# zfs list -t snapshot
NAME                    USED  AVAIL  REFER  MOUNTPOINT
data/share@20170801        0      -   105M  -
```

☛ スナップショットの容量は「USED」の値で確認できます。スナップショットの作成時点では容量は増えません。

4.1.2. バックアップの取得

1) バックアップの取得

スナップショットからバックアップを取得します。バックアップはスナップショット作成時点のデータから取得されるため、必ずバックアップの直前にスナップショットを作成するようにしてください。

```
# zfs send -R -v data/share@20170801 | pv | gzip > /backup/bk_20170801.gz
sending full stream to data/share@20170801
estimated stream size: 105M  ←出力されるストリームファイルのサイズ（圧縮前）
 105MB 0:00:01 [79.9MB/s] [  <=>                                             ]
 ↑作成途中のストリームファイルのサイズ、経過時間、データの転送速度
```

- ☛ zfs send コマンドはスナップショットのストリーム形式を出力するコマンドです。上記の例では、ストリーム出力を gzip で圧縮して保存しています。
- ☛ zfs send コマンドに-D オプションを付与することで、重複排除機能を使用したバックアップが可能です。
- ☛ バックアップ元のファイルシステムに下位のファイルシステムが存在する場合、-R オプションを指定すると再帰的にバックアップを取得できます。
- ☛ pv コマンドを使用することで、バックアップ実行中の進捗状況（ファイルサイズ、経過時間）とデータの転送速度を表示できます。

4.1.3. 差分バックアップの取得

1) スナップショットの作成

i) バックアップ対象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 -r data/share@20170802
```

ii) スナップショットが作成されたことを確認します。

```
# zfs list -t snapshot
NAME                       USED  AVAIL  REFER  MOUNTPOINT
data/share@20170801         31K      -   105M  -
data/share@20170802           0      -   115M  -
```

2) 差分バックアップの取得

前回スナップショットと今回スナップショットを指定して、差分バックアップを取得します。

```
# zfs send -R -v -i data/share@20170801 data/share@20170802 | gzip >
/backup/bk_20170802.gz
sending @20170801 to data/share@20170802
estimated stream size: 10.0M
 10MB 0:00:00 [21.5MB/s] [ <=>                                               ]
```

4.2. 自動バックアップ

4.2.1. スナップショットの自動化

1) タイムスライダーサービスの確認（サービスがインストールされていない場合）

```
# svcs time-slider
svcs: Pattern 'time-slider' doesn't match any instances
STATE          STIME    FMRI
```

☞ 上記はサービスがインストールされていない場合です。インストール済みの場合は、手順 3 へ進みます。

2) タイムスライダーサービスのインストール（サービスがインストールされていない場合）

```
# pkg install pkg:/desktop/time-slider
   インストールするパッケージ:         65
                  変更するサービス:         10
                  ブート環境の作成: いいえ
バックアップブート環境の作成: いいえ

【中略】

パッケージキャッシュを更新しています           1/1
```

☞ パッケージのインストールには、事前にリポジトリの設定が必要です。

3) タイムスライダーサービスと自動スナップショットの設定の確認

```
# svcs time-slider
STATE          STIME    FMRI
disabled       20:36:47 svc:/application/time-slider:default
# svcs -a | grep auto-snapshot
disabled       20:36:47 svc:/system/filesystem/zfs/auto-snapshot:monthly
disabled       20:36:47 svc:/system/filesystem/zfs/auto-snapshot:weekly
disabled       20:36:47 svc:/system/filesystem/zfs/auto-snapshot:daily
disabled       20:36:47 svc:/system/filesystem/zfs/auto-snapshot:hourly
disabled       20:36:48 svc:/system/filesystem/zfs/auto-snapshot:frequent
```

☞ 上記の例では、サービスがインストール済みで、自動スナップショットがすべて無効になっています。

4) 自動スナップショットの有効化と確認

```
# zfs set com.sun:auto-snapshot=true data/share
```

5) 自動スナップショットサービスの有効化

使用する動作タイミングのサービスを有効化します。以下の例では、「frequent」と「hourly」を有効化しています。

```
# svcadm enable svc:/system/filesystem/zfs/auto-snapshot:frequent
# svcadm enable svc:/system/filesystem/zfs/auto-snapshot:hourly
#
# svcs -a | grep auto-snapshot
disabled       20:36:47 svc:/system/filesystem/zfs/auto-snapshot:monthly
disabled       20:36:47 svc:/system/filesystem/zfs/auto-snapshot:weekly
disabled       20:36:47 svc:/system/filesystem/zfs/auto-snapshot:daily
offline        20:36:47 svc:/system/filesystem/zfs/auto-snapshot:hourly
offline        20:36:48 svc:/system/filesystem/zfs/auto-snapshot:frequent
```

☛ 「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。

6) タイムスライダーサービスの起動

```
# svcadm enable time-slider
# svcs time-slider
STATE          STIME    FMRI
online         20:40:46 svc:/application/time-slider:default
```

7) スナップショットの確認

```
# zfs list -t snapshot | grep data/share
data/share@zfs-auto-snap_hourly-2017-08-01-17h23                0      -   305M  -
data/share@zfs-auto-snap_frequent-2017-08-01-17h38              0      -   305M  -
data/share/groupA@zfs-auto-snap_hourly-2017-08-01-17h23         0      -  50.0M  -
data/share/groupA@zfs-auto-snap_frequent-2017-08-01-17h38       0      -  50.0M  -
data/share/groupB@zfs-auto-snap_hourly-2017-08-01-17h23         0      -    31K  -
data/share/groupB@zfs-auto-snap_frequent-2017-08-01-17h38       0      -    31K  -
```

☛ 自動スナップショットを有効化したファイルシステムに下位のファイルシステムが存在する場合は、下位のファイルシステムに対して自動スナップショットの設定が継承されます。

4.2.2. フルバックアップ運用の自動化

データが格納された ZFS を日次バックアップし、3 世代保持するよう設定を行います。本書では以下のように設定します。また、本手順は root 権限を持つユーザーで実施してください。

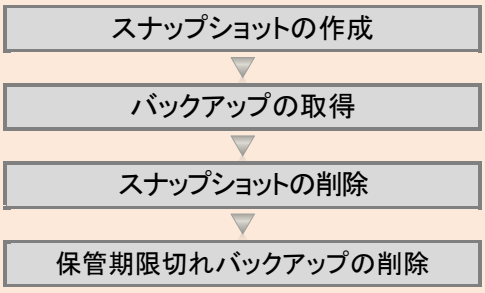
項目	内容
バックアップ対象ファイルシステム	data/share
バックアップ先	192.168.1.2:/backup ※バックアップサーバのディレクトリ
バックアップ先マウントポイント	/backup ※バックアップサーバを NFS マウントしたディレクトリ
世代数	3 世代
バックアップ実行タイミング	平日 5:30
タスクスケジューラ	cron を利用
バックアップスクリプト	/shell/full_backup.sh

☛ バックアップ先は外部装置上にすることを推奨します。

1) バックアップスクリプトの作成

Point

- **バックアップスクリプトの処理フロー**



バックアップスクリプトを作成します。

```
# vi /shell/full_backup.sh
```

日次バックアップを取得するスクリプトの作成例を以下に示します。

```
#!/bin/sh
#########################################################################
### 環境に合わせて変更してください
BkSource=data/share             # バックアップ元ファイルシステム
BkDest=/backup                  # バックアップ先ディレクトリ
Generation=3                    # 世代数
#########################################################################

Snap=`date +"%Y%m%d"`           # スナップショット名 yyyymmdd(年月日)
BkFile=bk_${Snap}.gz            # バックアップファイル名

### スナップショットの作成
zfs snapshot -r ${BkSource}@${Snap}

if [ $? -ne 0 ];then
  echo "スナップショットの作成に失敗しました。"
  return 1
else
  echo "スナップショット ${BkSource}@${Snap} を作成しました。"
fi

### バックアップの取得
zfs send -R ${BkSource}@${Snap} | gzip > ${BkDest}/${BkFile}

if [ $? -ne 0 ];then
  echo "バックアップの取得に失敗しました。"
  return 1
else
  echo "バックアップ ${BkDest}/${BkFile} を作成しました。"
fi

### スナップショットの削除
zfs destroy -r ${BkSource}@${Snap}

if [ $? -ne 0 ];then
  echo "スナップショット${BkSource}@${Snap} の削除に失敗しました。"
  return 1
else
  echo "スナップショット ${BkSource}@${Snap} を削除しました。"
fi

### 保管期限切れバックアップの削除
search=`echo ${BkFile} | sed s/${Snap}/*/g`
BkList+=(`ls -r ${BkDest}/${search}`)
index=0
```

```
for bk in "${BkList[@]}"
do
  if [ ${index} -ge ${Generation} ];then
    rm -f $bk
    if [ $? -eq 0 ];then
      echo "保管期限切れバックアップ ${bk} を削除しました。"
    else
      echo "保管期限切れバックアップ ${bk} の削除に失敗しました。"
      return 1
    fi
  fi
  index=`expr $index + 1`
done

echo "バックアップは正常終了しました。"
return 0
```

☛ バックアップを取得し、世代数を超えるバックアップを削除するスクリプトの例です。バックアップ先フォルダのファイル名の日付部分が古いファイルから順に削除されます。

☛ 処理に成功した場合は 0、失敗した場合は 1 の値を返します。

2) バックアップスクリプトの権限変更

バックアップスクリプトに実行権限を付与します。

```
# chmod 755 /shell/full_backup.sh
```

3) cron への登録

i) cron でバックアップスクリプトが自動的に実行されるよう設定します。バックアップは root ユーザーで実行するため、root ユーザーの crontab ファイルを編集します。

```
# crontab -e root
#ident  "%Z%%M% %I%     %E% SMI"
#
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# The root crontab should be used to perform accounting data collection.
#
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 [ -x /usr/lib/fs/nfs/nfsfind ] && /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
30 5 * * 1-5 /shell/full_backup.sh > /tmp/full_backup.log
```

☛ 上記の例では、平日（月～金曜日）の AM 5:30 に実行されるよう設定しています。また、実行時のログを/tmp 配下に出力するようにしています。

《参考》crontab ファイルのフィールド

- 左から順に以下のパラメーターを指定します。

時刻フィールド	値
分	0 - 59
時	0 - 23
日	1 - 31
月	1 - 12
曜日	0 - 6 (0 は日曜日)

- 各フィールドはスペースで区切ります。複数の値は「,」(カンマ)で区切るか、ハイフンを使用して範囲で指定します。
- すべての値を含むには、ワイルドカードとして「*」(アスタリスク)を指定します。

ii) crontab ファイルの内容を表示し、設定を確認します。

```
# crontab -l root
      :
30 5 * * 1-5 /shell/full_backup.sh > /tmp/full_backup.log
```

4.2.3. 差分バックアップ運用の自動化

データが格納された ZFS ファイルシステムを週次でフルバックアップし、平日は差分バックアップを取得するよう設定を行います。

本書では以下のように設定します。また、本手順は root 権限を持つユーザーで実施してください。

項目	内容
バックアップ対象ファイルシステム	data/share
バックアップ先	192.168.1.2:/backup ※バックアップサーバのディレクトリ
バックアップ先マウントポイント	/backup ※バックアップサーバを NFS マウントしたディレクトリ
世代数	フルバックアップ 3 世代
バックアップ実行タイミング	フルバックアップ 日曜 5:30 差分バックアップ 平日 5:30
タスクスケジューラ	cron を利用
バックアップスクリプト	/shell/inc_full_backup.sh （週次フルバックアップ） /shell/inc_backup.sh （日次差分バックアップ）

☛ バックアップ先は外部装置にすることを推奨します。

本書では、週次でフルバックアップを取得するスクリプトと、日次で差分バックアップを取得するスクリプトをそれぞれ作成します。

1) 週次フルバックアップスクリプトの作成

Point

- **週次フルバックアップスクリプトの処理フロー**

*1 :「4.2.2 フルバックアップ運用の自動化」で作成する日時のフルバックアップスクリプトでは、バックアップ取得後に削除しますが、差分バックアップが前提の場合は、スナップショット作成前に前回のスナップショットを削除します。これは、バックアップ時に作成したスナップショットは次回の差分バックアップに使用するためです。

週次でフルバックアップを取得するスクリプトを作成します。

```
# vi /shell/inc_full_backup.sh
```

週次でフルバックアップを取得するスクリプトの作成例を以下に示します。

```
#!/bin/sh
#########################################################################
### 環境に合わせて変更してください
BkSource=data/share             # バックアップ元ファイルシステム
BkDest=/backup                  # バックアップ先ディレクトリ
Generation=3                    # 世代数
#########################################################################

Snap=`date +"%Y%m%d"`_full      # スナップショット名 yyyymmdd(年月日)_full
BkFile=bk_${Snap}.gz            # バックアップファイル名

### 前回スナップショットの削除
SnList+=(`zfs list -t snapshot | egrep "${BkSource}@.*_inc|${BkSource}@.*_full" | cut -d'
' -f1`)
if [ ${#SnList[@]} -ne 0 ];then
  PrevSnap=${SnList[0]}
  zfs destroy -r ${PrevSnap}
  if [ $? -ne 0 ];then
    echo "スナップショット ${PrevSnap} の削除に失敗しました。"
    return 1
  else
    echo "スナップショット ${PrevSnap} を削除しました。"
  fi
fi

### スナップショットの作成
zfs snapshot -r ${BkSource}@${Snap}

if [ $? -ne 0 ];then
  echo "スナップショットの作成に失敗しました。"
  return 1
else
  echo "スナップショット ${BkSource}@${Snap} を作成しました。"
fi

### バックアップの取得
zfs send -R ${BkSource}@${Snap} | gzip > ${BkDest}/${BkFile}

if [ $? -ne 0 ];then
  echo "バックアップの取得に失敗しました。"
  return 1
else
  echo "バックアップ ${BkDest}/${BkFile} を作成しました。"
```

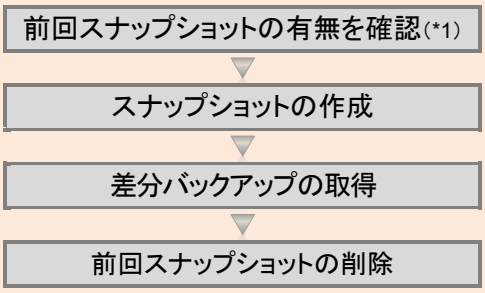
```
fi

### 保管期限切れバックアップの削除
search=`echo ${BkFile} | sed s/${Snap}/*/g`
BkList+=(`ls -r ${BkDest}/${search}`)
index=0
full_count=0

for bk in "${BkList[@]}"
do
  if [ ${full_count} -ge ${Generation} ];then
    rm -f $bk
    if [ $? -eq 0 ];then
      echo "保管期限切れバックアップ ${bk} を削除しました。"
    else
      echo "保管期限切れバックアップ ${bk} の削除に失敗しました。"
      return 1
    fi
  fi

  full=`echo ${bk} | grep .*_full.gz | wc -l`
  if [ ${full} -eq 1 ];then
    full_count=`expr $full_count + 1`
  fi

  index=`expr $index + 1`
done

echo "バックアップは正常終了しました。"
return 0
```

- ☛ バックアップを取得し、世代数を超えるバックアップを削除するスクリプトの例です。バックアップ先フォルダのファイル名の日付部分が古いファイルから順に削除されます。
- ☛ バックアップ時に作成したスナップショットは次回の差分バックアップに使用するため、削除せず残します。
- ☛ 処理に成功した場合は 0、失敗した場合は 1 の値を返します。

2) 日次差分バックアップスクリプトの作成

Point

- **日次差分バックアップスクリプトの処理フロー**



*1 :初回は前回バックアップ時のスナップショットが存在しないため、必ず先に週次フルバックアップスクリプトを実行します。

日次差分バックアップスクリプトを作成します。

```
# vi /shell/inc_backup.sh
```

日次差分バックアップを取得するスクリプトの作成例を以下に示します。

```
#!/bin/sh
#########################################################################
### 環境に合わせて変更してください
BkSource=data/share             # バックアップ元ファイルシステム
BkDest=/backup                  # バックアップ先ディレクトリ
#########################################################################

Snap=`date +"%Y%m%d"`_inc       # スナップショット名 yyyymmdd(年月日)_inc
BkFile=bk_${Snap}.gz            # バックアップファイル名

### 前回スナップショットの有無を確認
SnList+=(`zfs list -t snapshot | egrep "${BkSource}@.*_inc|${BkSource}@.*_full" | cut -d'
' -f1`)
if [ ${#SnList[@]} -eq 0 ];then
  echo "前回バックアップ時のスナップショットが存在しません。"
  return 1
else
  PrevSnap=${SnList[0]}
fi

### スナップショットの作成
zfs snapshot -r ${BkSource}@${Snap}

if [ $? -ne 0 ];then
```

```
  echo "スナップショットの作成に失敗しました。"
  return 1
else
  echo "スナップショット ${BkSource}@${Snap} を作成しました。"
fi

### 差分バックアップの取得
zfs send -R -i ${PrevSnap} ${BkSource}@${Snap} | gzip > ${BkDest}/${BkFile}

if [ $? -ne 0 ];then
  echo "バックアップの取得に失敗しました。"
  return 1
else
  echo "バックアップ ${BkDest}/${BkFile} を作成しました。"
fi

### 前回スナップショットの削除
zfs destroy -r ${PrevSnap}

if [ $? -ne 0 ];then
  echo "スナップショット ${PrevSnap} の削除に失敗しました。"
  return 1
else
  echo "スナップショット ${PrevSnap} を削除しました。"
fi

echo "バックアップは正常終了しました。"
return 0
```

☛ 処理に成功した場合は 0、失敗した場合は 1 の値を返します。

《注意》

前回バックアップ時のスナップショットからシェル実行時点までの差分を取得します。初回は前回バックアップ時のスナップショットが存在しないため、必ず週次フルバックアップスクリプトを実行してフルバックアップを取得してください。

3) バックアップスクリプトの権限変更

バックアップスクリプトに実行権限を付与します。

```
# chmod 755 /shell/inc_full_backup.sh
# chmod 755 /shell/inc_backup.sh
```

4) cron への登録

i) cron でバックアップスクリプトが自動的に実行されるよう設定します。バックアップは root ユーザーで実行するため、root ユーザーの crontab ファイルを編集します。

```
# crontab -e root
#ident  "%Z%%M% %I%     %E% SMI"
#
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# The root crontab should be used to perform accounting data collection.
#
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 [ -x /usr/lib/fs/nfs/nfsfind ] && /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
30 5 * * 0 /shell/inc_full_backup.sh > /tmp/inc_full_backup.log
30 5 * * 1-5 /shell/inc_backup.sh > /tmp/inc_backup.log
```

☛ 上記の例では、日曜の AM 5:30 にフルバックアップを実行し、平日（月～金曜日）の AM 5:30 に差分バックアップを実行するように設定しています。また、実行時のログを/tmp 配下に出力するようにしています。

ii) crontab ファイルの内容を表示し、設定を確認します。

```
# crontab -l root
     :
30 5 * * 0 /shell/inc_full_backup.sh > /tmp/inc_full_backup.log
30 5 * * 1-5 /shell/inc_backup.sh > /tmp/inc_backup.log
```

5. リストア手順

5.1.1. スナップショットからのロールバック

1) 最新のスナップショットからのロールバック

```
# zfs rollback data/share@20170801
```

☛ ロールバック対象のファイルシステムに下位のファイルシステムが存在する場合、上記コマンドのロールバック対象になりません。
下位のファイルシステムもロールバックする必要がある場合は、下位のファイルシステムごとにロールバックを実行してください。

2) 指定したスナップショットからのロールバック

最新のスナップショットより前のスナップショットにロールバックするには、中間にあるスナップショットをすべて削除する必要があります。

```
# zfs rollback -r data/share@20170731
```

☛ -r オプションを指定すると、指定したスナップショットよりあとに作成されたスナップショットを削除してロールバックを実行します。

《参考》特定のファイルのロールバック

スナップショットから特定のファイルを選択してロールバックすることができます。
以下の場合にロールバックを実行する例を示します。

項目	内容
ロールバック対象ファイル	/data/share/file
ロールバック元スナップショット	data/share@20170801

スナップショットの作成後に変更されたファイルの元データは、ファイルシステム配下の隠しフォルダに退避されています。退避された元データをコピーすることで復元することができます。

```
# ls -l /data/share/.zfs/snapshot/20170801
total 204833
-rw-------   1 root     root     104857600  8 月  1 日  18:58 file
# cp -p /data/share/.zfs/snapshot/20170801/file /data/share
```

5.1.2. フルバックアップからのリストア

フルバックアップからファイルシステムのデータをリストアします。本書では、以下の場合を例にリストアします。

項目	内容
リストア対象ファイルシステム	data/share
バックアップファイル名	/backup/bk_20170801.gz

1) スナップショットの削除

リストア先のファイルシステムにスナップショットが存在している場合、リストア前にすべてのスナップショットを削除してください。

```
# zfs list -t snapshot | grep data/share
data/share@20170802                              1K      -   305M  -
data/share@20170803                              1K      -   305M  -
# zfs destroy -r data/share@20170802
# zfs destroy -r data/share@20170803
```

☛ 複数のスナップショットが存在する場合は、1 つずつ削除を実行します。

2) リストアの実行

```
# gzip -dc /backup/bk_20170801.gz | pv | zfs receive -v -F data/share
receiving full stream of data/share@20170801 into data/share@20170801
100MB 0:00:01 [74.3MB/s] [  <=>                                                 ]
 ↑受信途中のストリームファイルのサイズ、経過時間、データの転送速度
received 100MB stream in 7 seconds (14.3MB/sec)
 ↑リストアの所要時間の詳細
```

☛ zfs receive コマンドに-F オプションを指定すると、リストア先にファイルシステムが存在する場合でも上書きしてリストアします。

☛ zfs receive コマンドに–v オプションを指定すると、リストアの所要時間の詳細を表示します。

☛ pv コマンドを使用することで、リストア実行中の進捗状況（ファイルサイズ、経過時間）とデータの転送速度を表示できます。

☛ ストリームの送信（zfs send コマンド）時に-R オプションを指定していた場合、下位のファイルシステムも同時に復元されます。

5.1.3. 差分バックアップからのリストア

フルバックアップおよび差分バックアップからファイルシステムのデータをリストアします。本書では、以下の場合を例にリストアします。

項目	内容
リストア対象ファイルシステム	data/share
バックアップファイル名	/backup/bk_20170801_full.gz（直近のフルバックアップ） /backup/bk_20170802_inc.gz（差分バックアップ） /backup/bk_20170803_inc.gz（差分バックアップ）

1) スナップショットの削除

リストア先のファイルシステムにスナップショットが存在している場合、リストア前にすべてのスナップショットを削除してください。

```
# zfs list -t snapshot | grep data/share
data/share@20170803_inc                          1K      -    305M  -
# zfs destroy -r data/share@20170803_inc
```

☛ 複数のスナップショットが存在する場合は、1 つずつ削除を実行します。

2) フルバックアップからのリストアの実行

```
# gzip -dc /backup/bk_20170801_full.gz | pv | zfs receive -v -F data/share
receiving full stream of data/share@20170801_full into data/share@20170801_full
201MB 0:00:01 [ 102MB/s] [  <=>                                                 ]
 ↑受信途中のストリームファイルのサイズ、経過時間、データの転送速度
received 201MB stream in 10 seconds (20.1MB/sec)
↑リストアの所要時間の詳細
```

☛ zfs receive コマンドに-F オプションを指定すると、リストア先にファイルシステムが存在する場合でも上書きしてリストアします。

☛ zfs receive コマンドに–v オプションを指定すると、リストアの所要時間の詳細を表示します。

☛ pv コマンドを使用することで、リストア実行中の進捗状況（ファイルサイズ、経過時間）とデータの転送速度を表示することができます。

3) 差分バックアップからのリストアの実行

フルバックアップ後に取得した差分バックアップを、古いバックアップから順に適用します。

```
# gzip -dc /backup/bk_20170802_inc.gz | pv | zfs receive -v -F data/share
receiving incremental stream of data/share@20170802_inc into data/share@20170802_inc
10MB 0:00:00 [19.7MB/s] [  <=>                                                  ]
received 10.0MB stream in 3 seconds (3.34MB/sec)
# gzip -dc /backup/bk_20170803_inc.gz | pv | zfs receive -v -F data/share
receiving incremental stream of data/share@20170803_inc into data/share@20170803_inc
12.2kB 0:00:00 [ 105MB/s] [  <=>                                                ]
received 10.0KB stream in 3 seconds (3.34KB/sec)
```

☛ 誤った順序で差分バックアップを適用しようとしても、リストアは開始されません。

改版履歴

改版日	版数	改版内容
2018 年 1 月	1.0	新規作成