

概要

ファイルシステムに xfs を利用し、UEFI 起動している場合のシステムバックアップとリストア方法

参考

http://chidipy.jp/server/other/xfs_efi_backuprestore.html

注意

https://bugzilla.redhat.com/show_bug.cgi?id=1405285

xfsdump にバグがあるので注意。

正常にバックアップが完了したように見えても、xfsresotre 時にエラーになることがある。

デバイス指定よりもマウントポイントを指定するほうが良い。

デバイスを指定した場合、bind でマウントされていたりすると bind 先をバックアップしようとする。

バックアップ

手順

1. バックアップ情報の保存
 - 1.efi 情報の保存
 - 2.fstab の保存
 - 3.label の収集
 4. パーティション情報の収集 (LVM 含む j)
2. バックアップ

バックアップ情報の保存

efi 情報の保存

```
efibootmgr -v > efibootmgr.txt
```

fstab の保存

```
cp /etc/fstab /mnt/backup/
```

ラベル情報

```
lsblk -f > lsblk.txt
```

パーティション情報の保存

```
fdisk -l /dev/sda > /mnt/backup/fdisk_sda.txt  
gdisk -l /dev/sda > /mnt/backup/gdisk_sda.txt
```

または

```
sfdisk -d /dev/hda > /mnt/backup/hda_info.txt  
リストア時に  
/sbin/sfdisk /dev/hda < hda.info
```

で復元可

LVM情報の保存

```
pvdisplay > /mnt/backup/pv_info.txt
vgdisplay > /mnt/backup/vg_info.txt
lvdisplay > /mnt/backup/lv_info.txt
```

または

```
vgdisplay -v > /mnt/backup/lvm_info.txt
```

バックアップ

UEFI で使用するファイルのバックアップ

```
sync # 念のため
cd /boot/efi
tar -C /boot/efi -cf バックアップ先 /boot_efi.tar .
```

```
xfsdump -l0 -e - /boot 2>>/mnt/backup/backup.log | gzip -c > /mnt/backup/boot.dump
xfsdump -l0 -e - / 2>>/mnt/backup/backup.log | gzip -c > /mnt/backup/root.dump
```

シングルモードの方がいいけど、そこまで神経質にならなくても良いと思います。
LVM のスナップショット機能を組み合わせてもいいと思う。

リストア

ディスクを交換せずにデータを戻すだけの場合は、データリストア以外の手順は省略して良い。
手順

1. パーティション復元
2. データのリストア
3. ラベル復元
4. efi 復元
5. SELinux 用の情報を付け直し

キーボードレイアウト設定

キーボードのレイアウトがおかしい場合

```
localectl set-keymap jp106
```

パーティションの復元

```
fdisk /dev/sda
```

または

```
gdisk /dev/sda
```

または

```
/sbin/sfdisk /dev/hda < hda.info
```

でパーティションを作る。

LVM 復元

VG、LV を同じ名前で再作成

LVM

基本的に

```
lvm pvcreate <デバイス>  
lvm vgcreate <vg名> <pvデバイスファイル>  
lvm lvcreate --size <lv容量> --name <lv名> <VG名>
```

を使う。

フォーマット

UEFI 用のファイルシステムを FAT32 で作成

```
mkfs.vfat -v -c -F32 -s2 -n ラベル -i <<UUID>> /dev/sda1
```

-i は UUID を指定する。ハイフンは不要。UUID を再作成する場合は -i は不要。

XFS のフォーマット

```
mkfs.xfs -f -m uuid=<<UUID>> -L ラベル /dev/sda2
```

ラベルの指定や UUID を再作成する場合

```
mkfs.xfs /dev/sda2  
mkfs.xfs /dev/mapper/centos-root
```

UUID 設定 (フォーマット時に指定しなかった場合)

基本的にフォーマット時に指定する。

以下のコマンドで再設定できるが、フォーマット後に UUID を変更すると以下の UEFI の再設定をしても起動できないことがある。

```
xfs_admin -U <<UUID>>
```

ラベル復元 (フォーマット時に指定しなかった場合)

ファイルシステムのラベル変更

XFS

```
xfs_admin -L <label> /dev/XXX
```

VFAT

```
fatlabel /dev/XXX <label>
```

ブートフラグを立てる

```
parted /dev/sda
p
set 1 boot on
```

この「1」はパーティション番号。pの結果からブートフラグを立てるパーティションを指定する。

データのリストア

リストア先をマウントしてから xfsrestore でリストア。

```
zcat /mnt/boot.dump | xfsrestore - /mnt/boot
zcat /mnt/root.dump | xfsrestore - /mnt/root
sync
```

UEFI のデータリストア

```
mount /dev/sda1 /mnt/efi
cd /mnt/efi
tar xf /mnt/backup/boot_efi.tar
```

スワップ領域

```
mkswap /dev/mapper/centos-swap
```

ラベルを付ける場合

```
mkswap -L ラベル /dev/mapper/centos-swap
```

fstab 編集

ルートのデバイスをマウントして /etc/fstab を編集する

```
mount /dev/mapper/centos-root /mnt/root
vi /mnt/root/etc/fstab
```

ディスクを変更したり、パーティションを作り直した場合、UUID が変わっているので UUID を

```
lsblk -f
```

の結果に合わせて書き換える。

面倒な場合は

```
/dev/mapper/centos-root /          xfs      defaults    0 0
/dev/sda2 /boot          xfs      defaults    0 0
/dev/sda1 /boot/efi          vfat     umask=0077,shortname=winnt 0 0
/dev/mapper/centos-swap swap      swap     defaults    0 0
```

のように UUID ではなく、デバイス名に書き換えても良い。

OS 起動後に UUID に戻してもいいし、LABEL の運用にしてもよい。

ラベルを使用する場合は ファイルシステムのラベル変更 を参考にデバイスにラベルを設定する。

UEFI ブート設定 (UUID を再作成した場合)

UUID が変更した場合は、UEFI ブートの設定を変更する。

```
efibootmgr -v # 起動順の参照
BootCurrent: 0000
Timeout: 1 seconds
BootOrder: 0000,0001
  Boot0000 * CentOS HD(1,800,64000,7 e44 aa01 -f593 -4 ce4 -8 ec8 -b3 afba558
cfc)File(¥EFI¥CENTOS¥SHIM.EFI)
  Boot0001* UEFI OS HD(1,800,64000,7e44aa01-f593-4ce4-8ec8-b3afba558cfc)File(¥EFI¥BOOT¥BOOTX64
.EFI)
# efibootmgr -b 1 -B # 削除
# efibootmgr -b 0 -B # 削除
```

-b は bootnum を指定。-B は --delete-bootnum。

起動順登録

```
efibootmgr -c -d /dev/sda -p 1 -l '¥EFI¥centos¥shimx64.efi' -L 'CentOS'
```

- -d はディスクのデバイスファイルパス
- -p は /boot/efi のパーティション番号
- -l はファイルシステム内のブートローダファイルへのパス

```
  Boot0000 * CentOS HD(1,GPT,185 b5 b92 -638 c-410 a-9718 -f5 d29 b8 c22 d6,0 x800,0 x64000
)/File(¥EFI¥centos¥shimx64.efi)
```

こんな感じに登録される。

```
efibootmgr -o 0000
```

のように起動順を指定する。

grup2 ブート設定ファイル再作成 (UUID を再作成した場合)

chroot 後に設定する。

```
chroot /mnt/sysimage
```

/boot/efi/EFI/centos/grub.cfg にも古い UUID が指定されているので再作成する

```
grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg
```

初期 RAM ディスク再作成 (UUID を再作成した場合)

古いディスクを探しに行くので再作成。これも chroot 後に実行。

```
cd /boot
mv initramfs-3.10.0-693.el7.x86_64.img initramfs-3.10.0-693.el7.x86_64.img.old
# dracut -f initramfs-3.10.0-693.el7.x86_64.img 3.10.0-693.el7.x86_64
```

3.10.0-693.el7.x86_64 は環境やカーネルのバージョンにより変わる。

ディスク同期

```
sync;sync;sync
```

好きなように・・・。

dump の除外について

<http://www.walbrix.com/jp/blog/2008-12-dumpxfsdump.html>

d 属性がついているファイルやディレクトリは dump や xfsdump によるダンプ時に除外することが出来る。

ファイルに d 属性を付けるには、

```
chattr +d ファイル名
```

とする。

xfsdump の場合は -e オプション指定時、dump の場合はインクリメンタルバックアップ時又は -h 0 オプションの指定時に d 属性のファイルがスキップされる。

下記は、ホームディレクトリ以下で 100MB 以上の大きさのファイルはバックアップ対象外にする例。

```
find /home -size +100M -exec chattr +d {} \;
```