

USB 스틱이 있는 Prime Infrastructure Gen1 어플라이언스 백업 및 복원

목차

[소개](#)

[문제](#)

[솔루션](#)

소개

이 문서에서는 Prime Infrastructure(PI) Gen 1 Appliance에서 USB 포트(Appliance)를 사용하여 외부적으로 또는 그 반대로 백업을 수행하는 방법에 대해 설명합니다.

문제

많은 고객 시나리오에서 데이터 센터에서 FTP/TFTP(File Transfer Protocol/Trivial File Transfer Protocol)의 사용이 제한되므로, 모든 FTP/TFTP 서버의 도움을 받아 백업을 PI에서 외부 지점으로 이동하는 것이 고객의 큰 과제입니다. 어플라이언스는 Linux 기반 서버이므로 부적절 한 전송 시 백업을 손상시킬 가능성이 있기 때문에 백업을 다른 방법에서 외부 지점으로 이동하는 것은 매우 어렵습니다.

솔루션

이러한 상황을 극복하기 위해 어플라이언스의 USB 포트를 사용하여 Prime Server에서 USB Stick으로 백업을 이동할 수 있는 대체 솔루션을 찾으십시오. 또 다른 장점은 훨씬 더 빠르고 FTP/TFTP/SFTP로 복사하는 데 걸리는 시간을 줄일 수 있으며 대용량 데이터를 복사하는 데도 많은 도움이 된다는 것입니다.

PI Gen1 장치에서 USB로 백업

1단계. USB 스틱을 삽입합니다.

2단계. 지원되는 새 파티션 ext4 파일 시스템을 만듭니다.

```
-bash-4.1# fdisk -l    n e l t 8e w
```

```
-bash-4.1# partprobe
```

```
-bash-4.1# mkfs.ext4 /dev/sdb1
```

```
-bash-4.1# mkdir /media/usb-drive/
```

```
-bash-4.1# mount -t ext4 /dev/sdb1 /media/usb-drive/
```

```
-bash-4.1# umount /media/usb-drive
```

3단계. defaultRepo에서 새로 마운트된 파일 시스템으로 백업을 복사합니다.

4단계. 두 위치에서 파일의 md5를 확인합니다.

USB에서 PI Gen1 어플라이언스로 백업

1단계. PI에 로그인합니다.

```
pi/admin#
```

2단계. 셸로 이동합니다.

```
pi/admin# shell
```

```
Enter shell access password :
```

```
Starting bash shell ...
```

```
ade #
```

```
ade #
```

```
ade # sudo su -
```

```
-bash-4.1#
```

3단계. PI에서 모든 디스크 파티션을 봅니다.

```
-bash-4.1# fdisk -l
```

```
Disk /dev/sda: 897.0 GB, 896998047744 bytes
```

```
255 heads, 63 sectors/track, 109053 cylinders
```

```
Units = cylinders of 16065 * 512 = 8225280 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disk identifier: 0x000591be
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	64	512000	83	Linux
Partition 1 does not end on cylinder boundary.						
/dev/sda2		64	77	102400	83	Linux
Partition 2 does not end on cylinder boundary.						
/dev/sda3		77	109054	875359232	8e	Linux LVM

Disk /dev/mapper/smosvg-rootvol: 4194 MB, 4194304000 bytes

255 heads, 63 sectors/track, 509 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x00000000

Disk /dev/mapper/smosvg-swapvol: 16.8 GB, 16777216000 bytes

255 heads, 63 sectors/track, 2039 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x00000000

Disk /dev/mapper/smosvg-tmpvol: 2113 MB, 2113929216 bytes

255 heads, 63 sectors/track, 257 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x00000000

Disk /dev/mapper/smosvg-usrvol: 7348 MB, 7348420608 bytes

255 heads, 63 sectors/track, 893 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x00000000

Disk /dev/mapper/smosvg-varvol: 4194 MB, 4194304000 bytes

255 heads, 63 sectors/track, 509 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x00000000

Disk /dev/mapper/smosvg-optvol: 716.3 GB, 716252905472 bytes

255 heads, 63 sectors/track, 87079 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x00000000

Disk /dev/mapper/smosvg-home: 134 MB, 134217728 bytes

255 heads, 63 sectors/track, 16 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x00000000

Disk /dev/mapper/smosvg-recvol: 134 MB, 134217728 bytes

255 heads, 63 sectors/track, 16 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x00000000

Disk /dev/mapper/smosvg-altrootvol: 134 MB, 134217728 bytes

255 heads, 63 sectors/track, 16 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x00000000

Disk /dev/mapper/smosvg-localdiskvol: 134.6 GB, 134553272320 bytes

255 heads, 63 sectors/track, 16358 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x00000000

Disk /dev/mapper/smosvg-storedatavol: 10.5 GB, 10502537216 bytes

255 heads, 63 sectors/track, 1276 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x00000000

Disk /dev/sdb: 62.1 GB, 62075699200 bytes

255 heads, 63 sectors/track, 7546 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0xa5fe72c5

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	7546	60613213+	8e	Linux LVM

4단계. 새 디렉토리를 생성하고 마운트합니다.

```
-bash-4.1# mkdir /media/usb-drive/
```

```
-bash-4.1# mount -t ext4 /dev/sdb1 /media/usb-drive/
```

```

-bash-4.1# ls -lv

total 60

-rw-----. 1 root root 8494 Aug 24 2018 anaconda-ks.cfg

drwxr-xr-x. 2 root root 4096 Aug 24 2018 bin

-rw-r--r--. 1 root root 18977 Aug 24 2018 install.log

-rw-r--r--. 1 root root 5646 Aug 24 2018 install.log.syslog

-rw-r--r--. 1 root root 5 Aug 24 2018 iso.ks.cfg

-rw-----. 1 root root 164 Aug 24 2018 ks-post.log

-rw-----. 1 root root 381 Aug 24 2018 ks-post-toinstall.log

-rw-rw-r--. 1 root root 120 Aug 23 17:47 test.log

-bash-4.1# cd /media/usb-drive/

-bash-4.1# pwd

/media/usb-drive

```

5단계. 백업을 복사하기 전에 USB에서 백업의 md5를 확인합니다.

```

-bash-4.1# ls -lv

total 21197320

-rw-r--r--. 1 root root 21706033973 Jun 28 14:57 pi-180419-1332__VER3.1.0.0.132_BKSZ204G_CPU16_MEM3G_RA

drwx-----. 2 root root 16384 Jun 28 14:29 lost+found

-bash-4.1#

-bash-4.1#

-bash-4.1# md5sum pi-180419-1332__VER3.1.0.0.132_BKSZ204G_CPU16_MEM3G_RAM15G_SWAP15G_APP_CK1589549125.t
44daa932e7ca10fafe480302f7a17b6a pi-180419-1332__VER3.1.0.0.132_BKSZ204G_CPU16_MEM3G_RAM15G_SWAP15G_AP

-bash-4.1#

-bash-4.1#

```

6단계. 백업을 /localdisk/defaultRepo 폴더에 복사합니다.

```

-bash-4.1# cp pi-180419-1332__VER3.1.0.0.132_BKSZ204G_CPU16_MEM3G_RAM15G_SWAP15G_APP_CK1589549125.tar.g

-bash-4.1#

```

```
-bash-4.1#
```

```
-bash-4.1# cd /localdisk/defaultRepo/
```

```
-bash-4.1# ls -lv
```

```
total 21218032
```

```
-rw-r--r--. 1 root root 21706033973 Aug 23 18:56 pi-180419-1332__VER3.1.0.0.132_BKSZ204G_CPU16_MEM3G_RA
```

```
-bash-4.1#
```

```
-bash-4.1#
```

7단계. 이전 md5를 사용하여 복사된 백업의 md5를 확인합니다.

```
-bash-4.1# md5sum pi-180419-1332__VER3.1.0.0.132_BKSZ204G_CPU16_MEM3G_RAM15G_SWAP15G_APP_CK1589549125.t
```

```
44daa932e7ca10fafe480302f7a17b6a pi-180419-1332__VER3.1.0.0.132_BKSZ204G_CPU16_MEM3G_RAM15G_SWAP15G_AP
```

```
-bash-4.1#
```

```
-bash-4.1#
```

8단계. 디렉토리를 마운트 해제합니다.

```
-bash-4.1# umount /media/usb-drive
```

```
-bash-4.1#
```

```
-bash-4.1#
```

```
-bash-4.1#
```

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.