

AppDirect 모드에 대해 VMware ESXi에서 DCPMM 구성

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[구성](#)

[서비스 프로파일 구성](#)

[ESXi 확인](#)

[가상 머신 NVDIMM 구성](#)

[가상 컴퓨터에서 네임스페이스 구성](#)

[문제 해결](#)

[관련 정보](#)

소개

이 문서에서는 호스트 관리 모드에서 인텔® Optane™ PMEM(Persistent Memory)을 사용하여 UCS(Unified Computing System) B 시리즈 서버에 ESXi를 구성하는 프로세스에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- UCS B 시리즈
- 인텔® Optane™ DCPMM(Data Center Persistent Memory Module) 개념
- VMware ESXi 및 vCenter 서버 관리

이 구성을 시도하기 전에 다음 요구 사항을 충족해야 합니다.

- B200/B480 M5 사양 [가이드](#)의 PMEM 지침을 [참조하십시오](#).
- CPU가 2세대 Intel® Xeon® 확장 가능한 프로세서인지 확인합니다.
- PMEM/DRAM(Dynamic Random Access Memory) 비율은 [KB 67645](#)의 요구 사항을 [충족합니다](#).
- ESXi는 6.7 U2 + Express 패치 10(ESXi670-201906002) 이상입니다.6.7 이전 릴리스는 지원되지 않습니다.
- UCS Manager 및 서버는 4.0(4) 이상 버전입니다.최신 권장 버전은 www.software.cisco.com/[를 참조하십시오](#).

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- UCS B480 M5
- UCS Manager 4.1(2b)

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 이해해야 합니다.

배경 정보

App Direct 모드로 구성된 UCS Server에서 VMware ESXi 가상 머신은 Optane DCPMM Persistent Memory Non-Volatile Dual In-Line Memory Module(NVDIMM)에 액세스합니다.

Intel Optane DCPMM은 UEFI(Unified Extensible Firmware Interface) 셸이나 OS 유틸리티를 통해 IPMCTL 관리 유틸리티를 통해 구성할 수 있습니다. 이 도구는 다음 작업 중 일부를 수행하도록 설계되었습니다.

- 모듈 검색 및 관리
- 모듈 펌웨어 업데이트 및 구성
- 상태 모니터링
- 목표, 지역 및 네임스페이스 프로비저닝 및 구성
- PMEM 디버그 및 문제 해결

UCS는 서비스 프로필에 연결된 영구 메모리 정책을 사용하여 손쉽게 구성할 수 있습니다.

오픈 소스 NDCTL(Non-Volatile Device Control) 유틸리티는 LIBNVDIMM Linux 커널 하위 시스템을 관리하는 데 사용됩니다. NDCTL 유틸리티를 사용하면 OS를 사용할 수 있는 영역 및 네임스페이스로 구성을 프로비저닝하고 수행할 수 있습니다.

ESXi 호스트에 추가된 영구 메모리는 호스트에서 감지되고 포맷되어 로컬 PMem 데이터 저장소로 마운트됩니다. PMEM을 사용하기 위해 ESXi는 VMFS(Virtual Machine Flying System)-L 파일 시스템 형식을 사용하며 호스트당 하나의 로컬 PMEM 데이터 저장소만 지원됩니다.

다른 데이터 저장소와 달리 PMEM 데이터 저장소는 기존 데이터 저장소로서 작업을 지원하지 않습니다. vmx 및 vmware.log 파일이 있는 VM 홈 디렉토리는 PMEM 데이터 저장소에 배치할 수 없습니다.

PMEM은 두 가지 다른 모드로 VM에 제공할 수 있습니다. 직접 액세스 모드 및 가상 디스크 모드.

- 직접 액세스 모드
VM은 NVDIMM 형식으로 PMEMregion을 표시하여 이 모드에 대해 구성할 수 있습니다. 이 모드를 사용하려면 VM 운영 체제가 PMem 인식되어야 합니다. NVDIMM은 바이트 주소 지정 가능 메모리로 작동하므로 NVDIMM 모듈에 저장된 데이터는 전원 주기 동안 유지될 수 있습니다. NVDIMM은 PMEM을 형성할 때 ESXi에서 생성한 PMem 데이터 저장소에 자동으로 저장됩니다.
- 가상 디스크 모드

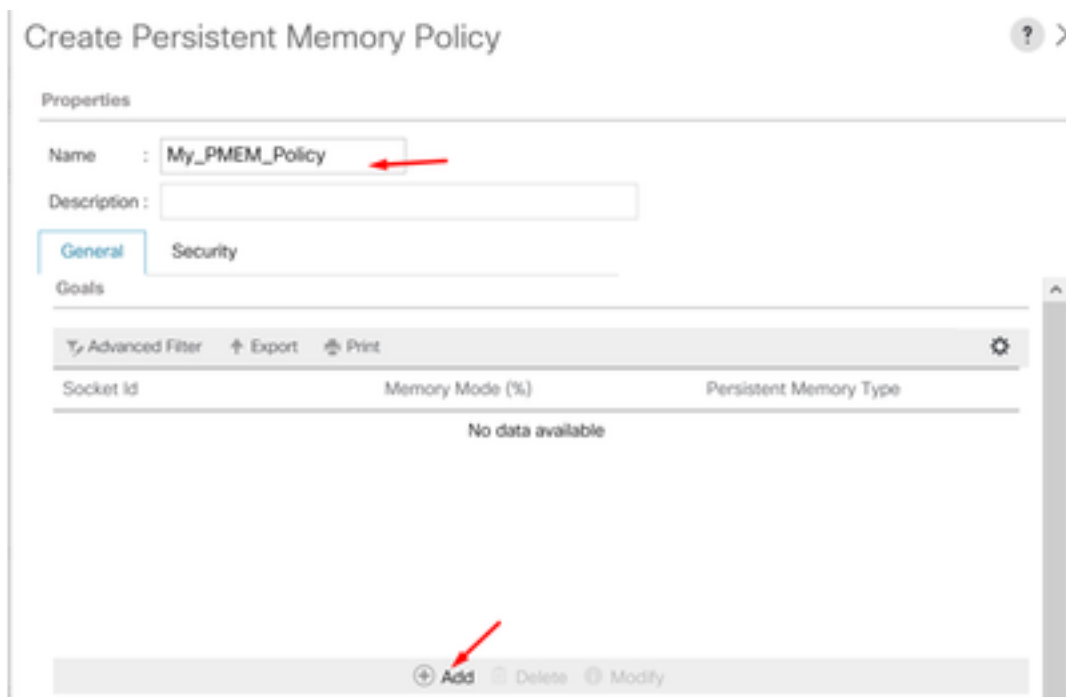
모든 하드웨어 버전을 지원하기 위해 VM에 상주하는 기존 및 레거시 OS를 대상으로 합니다.
.VM OS는 PMEM을 인식할 필요가 없습니다.이 모드에서는 VM OS에서 기존 SCSI(Small Computer System Interface) 가상 디스크를 생성하고 사용할 수 있습니다.
이 문서에서는 Direct-Access 모드에서 가상 머신을 사용하기 위한 구성에 대해 설명합니다.

구성

이 절차에서는 Intel Optane DCPMM을 사용하여 UCS Blade Series 서버에서 ESXi를 구성하는 방법에 대해 설명합니다.

서비스 프로파일 구성

1. UCS Manager GUI에서 **Servers(서버) > Persistent Memory Policy(영구 메모리 정책)**로 이동하고 이미지에 표시된 대로 **Add(추가)**를 클릭합니다.



2. 목표를 생성하고 메모리 모드가 이미지에 표시된 대로 0%인지 확인합니다.

Create Goal



Properties

Socket ID : All Sockets

Memory Mode (%) :

Persistent Memory Type : App Direct App Direct Non Interleaved

OK

Cancel

3. 원하는 서비스 프로필에 PMEM 정책을 추가합니다.

Service Profile(서비스 프로필) > Policies(정책) > Persistent Memory Policy(영구 메모리 정책)로 이동하고 생성된 정책을 연결합니다.

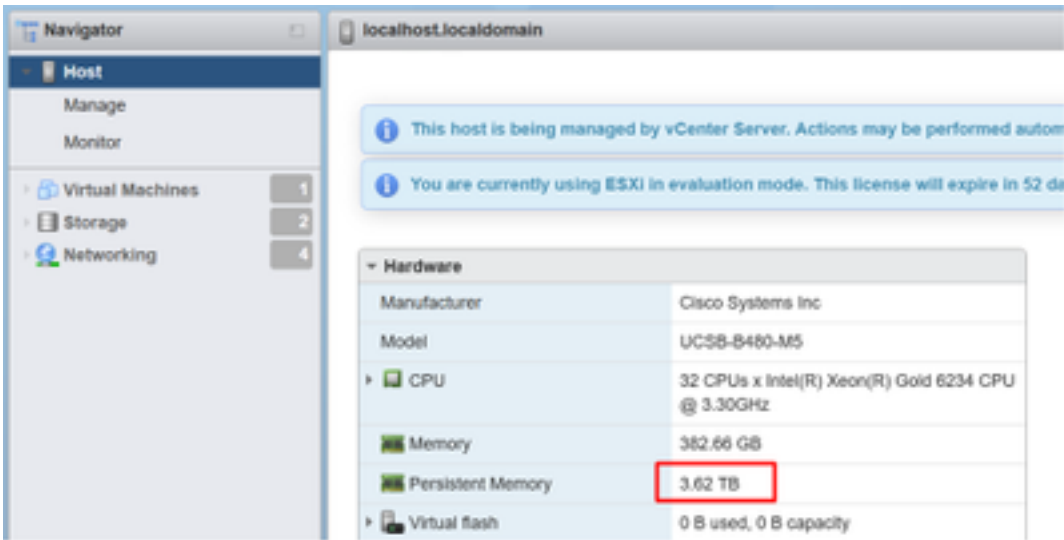
4. 지역의 상태를 확인합니다.

선택한 Server(서버) > Inventory(인벤토리) > Persistent Memory(영구 메모리) > Regions(영역)로 이동합니다.AppDirect 유형이 표시됩니다.이 방법은 CPU 소켓당 하나의 영역을 만듭니다.

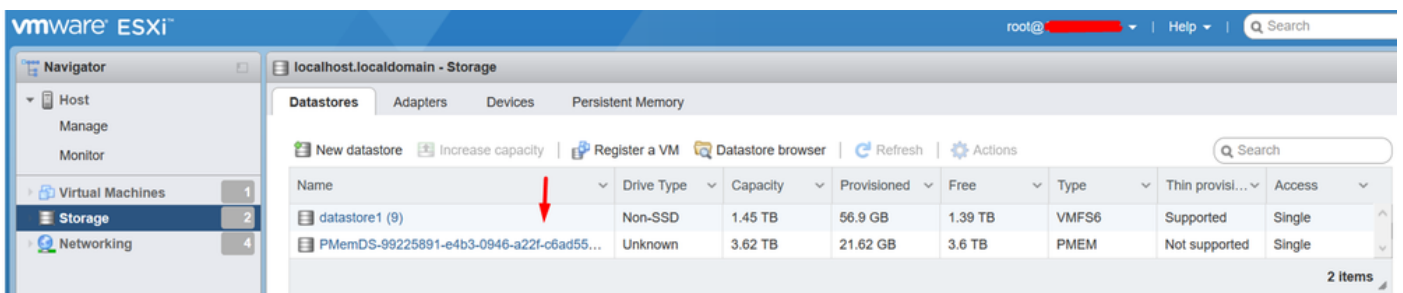
Id	Socket Id	Local DIMM Slot	DIMM Locator Id	Type	Total Capacity (L)	Free Capacity (L)	Health Status
1	Socket 1	Not Applicable	DIMM_A2.DIMM...	AppDirect	928	928	Healthy
2	Socket 2	Not Applicable	DIMM_G2.DIMM...	AppDirect	928	928	Healthy
3	Socket 3	Not Applicable	DIMM_N2.DIMM...	AppDirect	928	928	Healthy
4	Socket 4	Not Applicable	DIMM_U2.DIMM...	AppDirect	928	928	Healthy

ESXi 확인

1. 웹 콘솔에서 호스트는 사용 가능한 총 PMEM을 표시합니다.

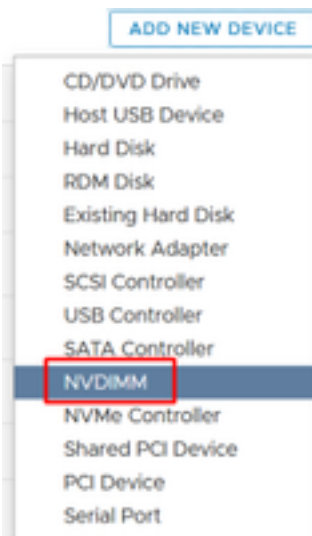


2. ESXi는 이미지에 표시된 대로 총 PMEM으로 구성된 특수 데이터 저장소를 표시합니다.



가상 머신 NVDIMM 구성

1. ESXi에서 가상 시스템은 Optane DCPMM PMEM에 NVDIMM으로 액세스합니다. 가상 머신에 NVDIMM을 할당하려면 vCenter를 통해 가상 머신에 액세스한 다음 Actions(작업) > **Edit Settings(설정 수정)**로 이동하고 **ADD NEW DEVICE(새 디바이스 추가)**를 클릭하고 이미지에 표시된 대로 NVDIMM을 선택합니다.



참고: 가상 머신을 만들 때 OS 호환성이 인텔® Optane™ 영구 메모리를 지원하는 최소 필수 버전을 충족하는지 확인합니다. 그렇지 않으면 **NVDIMM** 옵션이 선택 가능한 항목에 나타나지 않습니다.

2. 이미지에 표시된 대로 NVDIMM의 크기를 설정합니다.



가상 컴퓨터에서 네임스페이스 구성

1. NDCTL 유틸리티는 PMEM 또는 NVDIMM을 관리하고 구성하는 데 사용됩니다.

이 예에서는 Red Hat 8이 구성에 사용됩니다. Microsoft에는 영구 메모리 네임스페이스 관리를 위한 PowerShell cmdlet이 있습니다.

Linux 배포판에 따라 사용 가능한 툴을 사용하여 NDCTL 유틸리티를 다운로드합니다.

예:

```
# yum install ndctl # zypper install ndctl # apt-get install ndctl
```

2. NVDIMM이 가상 머신에 할당되면 ESXi에서 기본적으로 생성한 NVDIMM 영역 및 네임스페이스를 확인하고 구성과 일치하는 공간이 있는지 확인합니다. 네임스페이스의 모드가 **raw**로 설정되어 있는지 확인합니다. 이는 ESXi가 네임스페이스를 생성했음을 의미합니다. 확인하려면 다음 명령을 사용합니다.

```
# ndctl list -RuN
```

```
admin@localhost:/etc
File Edit View Search Terminal Help
[admin@localhost etc]$ ndctl list -RuN
{
  "regions": [
    {
      "dev": "region0",
      "size": "20.00 GiB (21.47 GB)",
      "available_size": 0,
      "max_available_extent": 0,
      "type": "pmem",
      "persistence_domain": "unknown",
      "namespaces": [
        {
          "dev": "namespace0.0",
          "mode": "raw",
          "size": "20.00 GiB (21.47 GB)",
          "blockdev": "pmem0"
        }
      ]
    }
  ]
}
```

3. (선택 사항) 네임스페이스를 아직 만들지 않은 경우 다음 명령을 사용하여 네임스페이스를 만들 수 있습니다.

```
# ndctl create-namespace
```

`ndctl create-namespace` 명령은 기본적으로 `fsdax` 모드에서 새 네임스페이스를 만들고 새 `/dev/pmem([x].[y])` 디바이스를 생성합니다. 네임스페이스가 이미 생성된 경우 이 단계를 건너뛸 수 있습니다.

4. PMEM 액세스 모드를 선택합니다. 구성에 사용할 수 있는 모드는 다음과 같습니다.

- 섹터 모드:

스토리지를 빠른 블록 디바이스로 표시합니다. 영구 메모리를 사용할 수 없는 기존 애플리케이션에 유용합니다.

- Fsdax 모드:

영구 메모리 장치가 NVDIMM에 직접 액세스할 수 있도록 합니다. 파일 시스템 직접 액세스를 사용하려면 직접 액세스 프로그래밍 모델을 사용하려면 `fsdax` 모드를 사용해야 합니다. 이 모드에서는 NVDIMM 위에 파일 시스템을 생성할 수 있습니다.

- Devdax 모드:

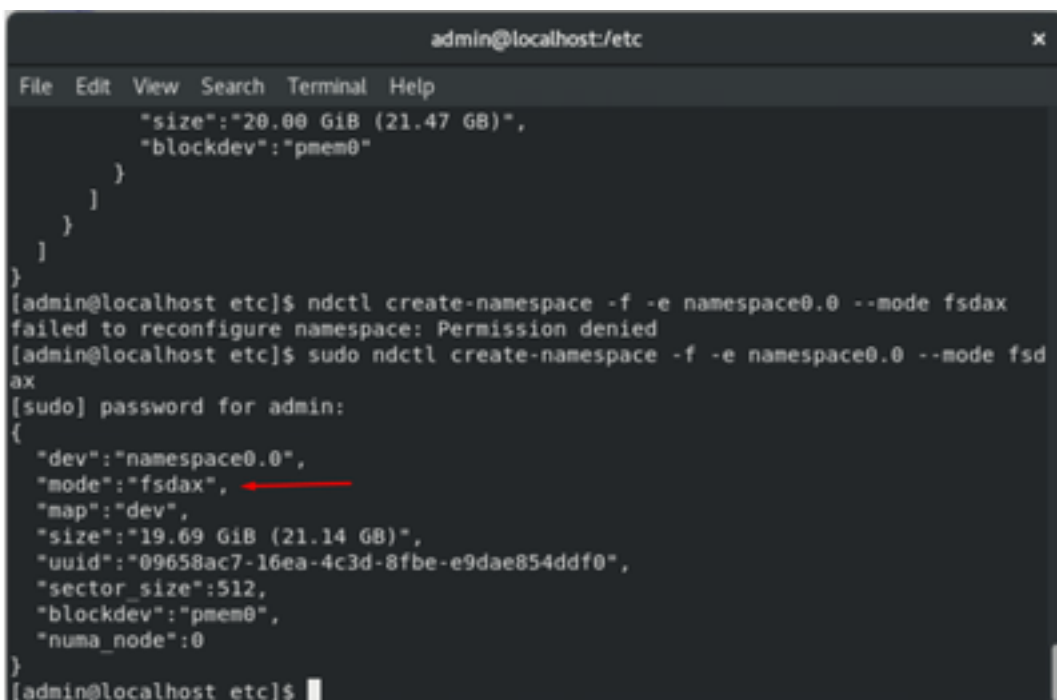
DAX 문자 장치를 사용하여 영구 메모리에 대한 원시 액세스를 제공합니다. `devdax` 모드를 사용하는 장치에서는 파일 시스템을 만들 수 없습니다.

- 원시 모드:

이 모드는 몇 가지 제한 사항이 있으며 영구 메모리를 사용하는 경우에는 권장되지 않습니다. 모드를 `fsdax` 모드로 변경하려면 다음 명령을 사용합니다.

```
ndctl create-namespace -f -e
```

`dev`가 이미 생성된 경우, `dev` 네임스페이스는 모드를 `fsdax`로 포맷하고 수정하는 데 **사용됩니다**.



```
admin@localhost:/etc
File Edit View Search Terminal Help
    "size": "20.00 GiB (21.47 GB)",
    "blockdev": "pmem0"
  }
}
}
}
}
[admin@localhost etc]$ ndctl create-namespace -f -e namespace0.0 --mode fsdax
failed to reconfigure namespace: Permission denied
[admin@localhost etc]$ sudo ndctl create-namespace -f -e namespace0.0 --mode fsdax
[sudo] password for admin:
{
  "dev": "namespace0.0",
  "mode": "fsdax",
  "map": "dev",
  "size": "19.69 GiB (21.14 GB)",
  "uuid": "09658ac7-16ea-4c3d-8fbe-e9dae854ddf0",
  "sector_size": 512,
  "blockdev": "pmem0",
  "numa_node": 0
}
[admin@localhost etc]$
```

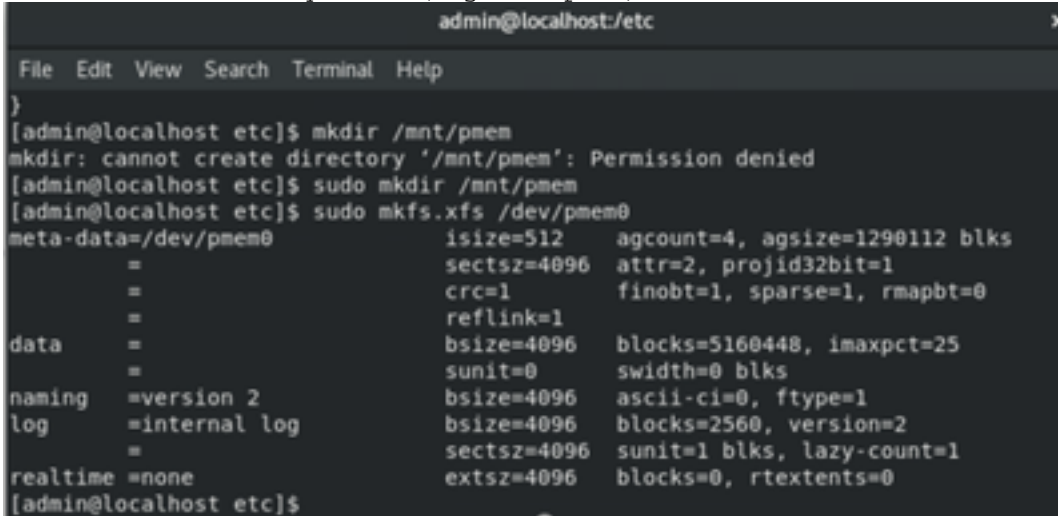
참고: 이 명령을 사용하려면 어카운트에 루트 권한이 있어야 합니다. sudo 명령이 필요할 수 있습니다.

5. 디렉토리 및 파일 시스템을 생성합니다.

Direct Access(직접 액세스) 또는 DAX는 애플리케이션이 기존 I/O 스택을 우회하여 CPU에서 지속적인 미디어에 직접 액세스할 수 있도록 하는 메커니즘입니다. DAX 지원 영구 메모리 파일 시스템에는 ext4, XFS 및 Windows NTFS가 포함됩니다.

생성 및 마운트된 XFS 파일 시스템의 예:

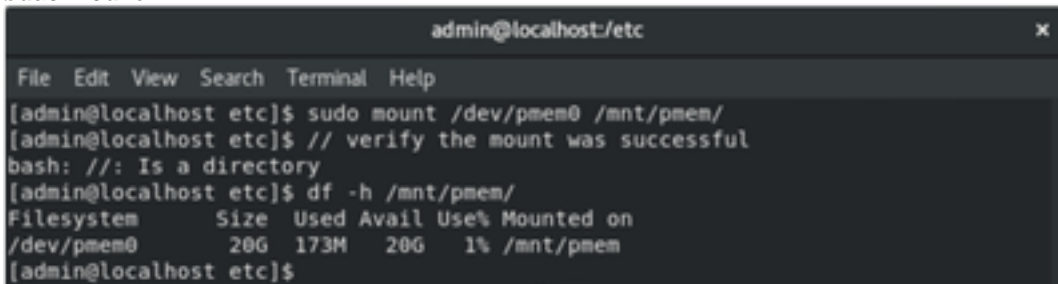
```
sudo mkdir < directory route (e.g./mnt/pmem) > sudo mkfs.xfs < /dev/devicename (e.g. pmem0) >
```



```
admin@localhost:/etc
File Edit View Search Terminal Help
}
[admin@localhost etc]$ mkdir /mnt/pmem
mkdir: cannot create directory '/mnt/pmem': Permission denied
[admin@localhost etc]$ sudo mkdir /mnt/pmem
[admin@localhost etc]$ sudo mkfs.xfs /dev/pmem0
meta-data=/dev/pmem0          isize=512    agcount=4, agsize=1290112 blks
=                               sectsz=4096  attr=2, projid32bit=1
=                               crc=1      finobt=1, sparse=1, rmapbt=0
=                               reflink=1
data            =             bsize=4096  blocks=5160448, imaxpct=25
=                               sunit=0    swidth=0 blks
naming         =version 2   bsize=4096  ascii-ci=0, ftype=1
log            =internal log bsize=4096  blocks=2560, version=2
=                               sectsz=4096  sunit=1 blks, lazy-count=1
realtime      =none        extsz=4096  blocks=0, rtextents=0
[admin@localhost etc]$
```

6. 파일 시스템을 마운트하고 성공했는지 확인합니다.

```
sudo mount
```

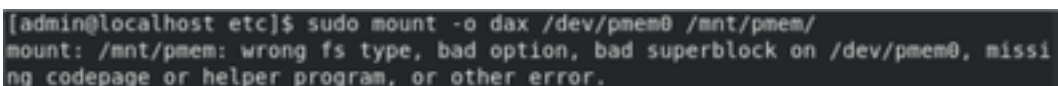


```
admin@localhost:/etc
File Edit View Search Terminal Help
[admin@localhost etc]$ sudo mount /dev/pmem0 /mnt/pmem/
[admin@localhost etc]$ // verify the mount was successful
bash: //: Is a directory
[admin@localhost etc]$ df -h /mnt/pmem/
Filesystem      Size  Used Avail Use% Mounted on
/dev/pmem0      20G  173M   20G   1% /mnt/pmem
[admin@localhost etc]$
```

VM에서 PMEM을 사용할 준비가 되었습니다.

문제 해결

오류가 발견되면 `-o dax` 마운트 옵션을 사용하여 이 DAX 지원 파일 시스템을 마운트하는 것이 좋습니다.



```
[admin@localhost etc]$ sudo mount -o dax /dev/pmem0 /mnt/pmem/
mount: /mnt/pmem: wrong fs type, bad option, bad superblock on /dev/pmem0, missing codepage or helper program, or other error.
```

파일 시스템 복구를 실행하여 무결성을 보장합니다.


```
[admin@localhost etc]$ sudo xfs_repair /dev/pmem0
[sudo] password for admin:
Phase 1 - find and verify superblock...
Phase 2 - using internal log
- zero log...
- scan filesystem freespace and inode maps...
- found root inode chunk
Phase 3 - for each AG...
- scan and clear agi unlinked lists...
- process known inodes and perform inode discovery...
- agno = 0
- agno = 1
- agno = 2
- agno = 3
- process newly discovered inodes...
Phase 4 - check for duplicate blocks...
- setting up duplicate extent list...
- check for inodes claiming duplicate blocks...
- agno = 0
- agno = 1
- agno = 2
- agno = 3
Phase 5 - rebuild AG headers and trees...
- reset superblock..
Phase 6 - check inode connectivity...
- resetting contents of realtime bitmap and summary inodes
- traversing filesystem ...
- traversal finished ...
- moving disconnected inodes to lost+found ...
Phase 7 - verify and correct link counts...
done
[admin@localhost etc]$
```

이를 해결하려면 `-o dax` 옵션 없이 마운트를 마운트할 수 있습니다.

참고: xfsprogs 버전 5.1에서 기본값은 `reflink` 옵션이 활성화된 XFS 파일 시스템을 생성하는 것입니다. 이전에는 기본적으로 비활성화되었습니다. **역광** 및 `dax` 옵션은 마운트가 실패하게 하는 상호 배타적입니다.

"DAX와 리플렉스는 함께 사용할 수 없습니다!" mount 명령이 실패할 경우 dmesg에서 오류를 확인할 수 있습니다.

```

admin@localhost:/etc
File Edit View Search Terminal Help
log      =internal log          bsize=4096   blocks=2560, version=2
         =                      sectsz=4096  sunit=1 blks, lazy-count=1
realtime =none                extsz=4096   blocks=0, rtextents=0
[admin@localhost etc]$ mount -o dax /dev/pmem0 /mnt/pmem
mount: only root can use "--options" option
[admin@localhost etc]$ sudo mount -o dax /dev/pmem0 /mnt/pmem/
mount: /mnt/pmem: wrong fs type, bad option, bad superblock on /dev/pmem0, missing
codepage or helper program, or other error.
[admin@localhost etc]$ dmesg -T | tail
[mar nov 10 00:12:18 2020] VFS: busy inodes on changed media or resized disk sr0
[mar nov 10 00:12:22 2020] ISO 9660 Extensions: Microsoft Joliet Level 3
[mar nov 10 00:12:22 2020] ISO 9660 Extensions: RRIP_1991A
[mar nov 10 01:47:35 2020] pmem0: detected capacity change from 0 to 21137195008
[mar nov 10 01:51:19 2020] XFS (pmem0): DAX enabled. Warning: EXPERIMENTAL, use
at your own risk
[mar nov 10 01:51:19 2020] XFS (pmem0): DAX and reflink cannot be used together!
[mar nov 10 01:53:06 2020] XFS (pmem0): DAX enabled. Warning: EXPERIMENTAL, use
at your own risk
[mar nov 10 01:53:06 2020] XFS (pmem0): DAX and reflink cannot be used together!
[mar nov 10 01:59:29 2020] XFS (pmem0): DAX enabled. Warning: EXPERIMENTAL, use
at your own risk
[mar nov 10 01:59:29 2020] XFS (pmem0): DAX and reflink cannot be used together!
[admin@localhost etc]$

```

해결 방법으로 `-o dax` 옵션을 제거합니다.

```

admin@localhost:/etc
File Edit View Search Terminal Help
[admin@localhost etc]$ sudo mount /dev/pmem0 /mnt/pmem/
[admin@localhost etc]$ // verify the mount was successful
bash: //: Is a directory
[admin@localhost etc]$ df -h /mnt/pmem/
Filesystem      Size  Used Avail Use% Mounted on
/dev/pmem0      20G  173M   20G   1% /mnt/pmem
[admin@localhost etc]$

```

ext4 FS를 사용하여 마운트합니다.

EXT4 파일 시스템은 Reflink 기능을 구현하지 않고 DAX를 지원하기 때문에 대안으로 사용할 수 있습니다.

```

[admin@localhost etc]$ sudo mkfs.ext4 /dev/pmem0
mke2fs 1.44.3 (10-July-2018)
/dev/pmem0 contains a xfs file system
Proceed anyway? (y,N) y
Creating filesystem with 5160448 4k blocks and 1291808 inodes
Filesystem UUID: 164c6d57-0462-45a0-9b94-703719272816
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

[admin@localhost etc]$ sudo mount /dev/pmem0 /mnt/pmem/
[admin@localhost etc]$ df -h /mnt/pmem/
Filesystem      Size  Used Avail Use% Mounted on
/dev/pmem0      20G   45M   19G   1% /mnt/pmem
[admin@localhost etc]$

```

관련 정보

- [빠른 시작 가이드:인텔® Optane™ DC 영구 메모리 프로비저닝](#)

- [영구 메모리 구성](#)
- [인텔® Optane™ 영구 메모리용 관리 유틸리티 ipmctl 및 ndctl](#)
- [기술 지원 및 문서 - Cisco Systems](#)