The Zenoss Enablement Series:

How to Create and Size Resource Manager/Control Center File Systems

Document Version 500-P1

Zenoss, Inc.

www.zenoss.com

Table of Contents

Applies To

- Zenoss 5.0.x
- Control Center 1.0.x

Summary

This KB provides guidance on how to choose between using fixed disks and logical volumes. It describes the basic process to create and define the required partitions and filesystems for Zenoss Resource Manager and Control Center and provides two different example scenarios.

Background

Disk Choices - Fixed versus Logical Volumes

There are two options to choose between when configuring disks for use with Zenoss Resource Manager and Control Center – *fixed volume filesystems* (disk) or *logical volume filesystems*.

Fixed Disks

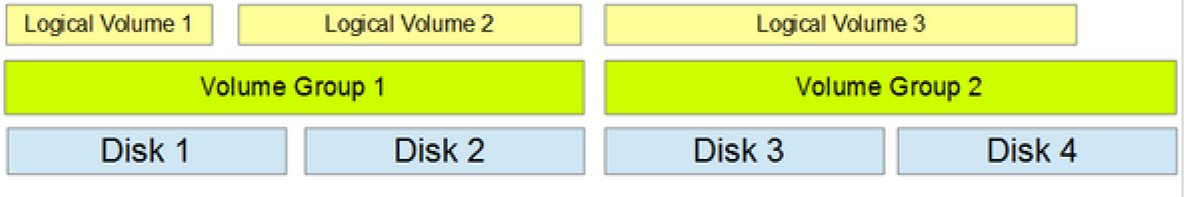
Using fixed disks/volumes means volumes can only reside on a single hard drive. In this case, the hard drive is divided into partition(s)/volume(s) with (pre)defined, fixed sizes. This means that if a partition becomes full, the size of the partition must be expanded. One solution is to move the partition to a different hard drive with more space. Moving the partition to another hard drive results in the added work required to reallocate the hard drive space on the original hard drive or lose the space if you do not use it.

Logical Volumes

Logical volumes are not limited to the available space on a single hard drive because they enable spanning data across multiple hard drives or block devices. This means that if a volume becomes full, the elastic quality of the logical volume enables addition of available storage space on another drive. This makes it more convenient to expand the available storage space when required.

Logical volumes enable leaving free (unpartitioned) space on hard drives that can be added to a specific volume when required. This additional space can be added without backing up the data and reformatting the hard drive. Additionally, the unpartitioned space can be (re)allocated dynamically through either the command line or a graphical UI, without a reboot.

Logical volume(s) are gathered into v*olume group(s).* A volume group defines the physical disk drives that provide space to the volume. The following example shows relationships between logical volumes, volume groups, and physical disk drives:



Note that logical volumes and volume groups can extend across multiple disks but a physical drive can exist only within a single volume group.

Caveats for Logical Volumes

Although it is easy and safe to add space to a logical volume, it is difficult and risky to attempt to shrink a logical volume. Shrinking a logical volume can result in data corruption or loss. If it becomes necessary to shrink a logical volume, it is recommended to back up the data and migrate it to another drive.

While spanning a logical volume across multiple drives enables elasticity, loss of a single drive can damage the logical volume. The data can become corrupt or the entire logical volume can become unusable. This makes it important to perform complete and regular backups of the entire volume.

Required Filesystems

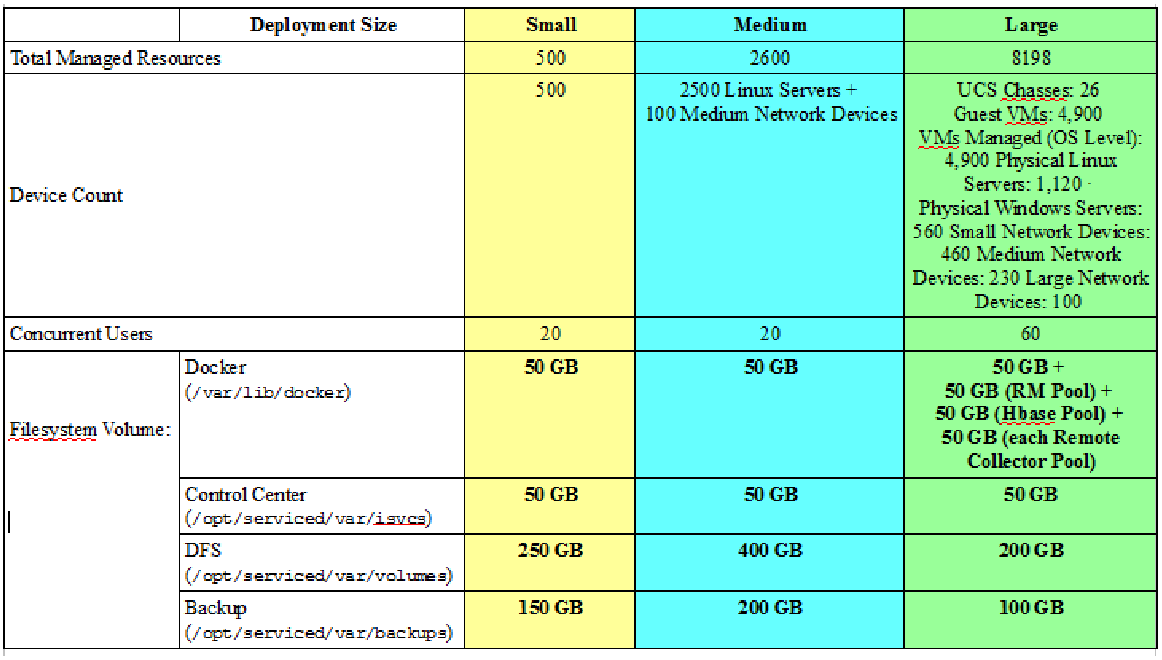
In addition to the Linux and swap partitions, there are four filesystems required for Zenoss Resource Manager and Control Center:

- `/var/lib/docker` - for Docker
- `/opt/serviced/var/isvcs` - For Control Center
- `/opt/serviced/var/volumes` - For DFS
- `/opt/serviced/var/backups` - For Backup

Filesystem Sizing

The relative sizes for the required filesystems are highly dependent on individual system deployments.

In general there are rough guidelines to help choose starting sizes for the filesystems, based on the number of *managed resources*, *number of concurrent users*, and the *device count.* The guidelines cover three sizes of deployments: small, medium and large.



	Deployment Size	Small	Medium	Large
Total Managed Resources		500	2600	8198
Device Count		500	2500 Linux Servers + 100 Medium Network Devices	UCS Chasses: 26 Guest VMs: 4,900 VMs Managed (OS Level): 4,900 Physical Linux Servers: 1,120 - Physical Windows Servers: 560 Small Network Devices: 460 Medium Network Devices: 230 Large Network Devices: 100
Concurrent Users		20	20	60
Filesystem Volume:	Docker (/var/lib/docker)	**50 GB**	**50 GB**	**50 GB + 50 GB (RM Pool) + 50 GB (Hbase Pool) + 50 GB (each Remote Collector Pool)**
	Control Center (/opt/serviced/var/isvcs)	**50 GB**	**50 GB**	**50 GB**
	DFS (/opt/serviced/var/volumes)	**250 GB**	**400 GB**	**200 GB**
	Backup (/opt/serviced/var/backups)	**150 GB**	**200 GB**	**100 GB**

Procedures

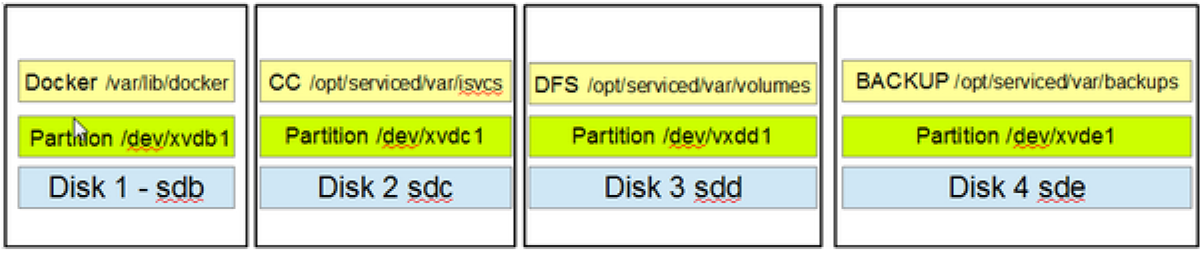
The following example procedures describe two different scenarios:

- Fixed Volume Filesystem Example-
  A system composed of fixed volumes only- Four disks configured as four fixed volume filesystems.
- Logical Volume Creation Example-
  A mixed system composed of both fixed and logical filesystems - Six disks configured as two logical volume filesystems (two partitioned disks each) and two fixed volume filesystems.

Fixed Volume Filesystem Example

Creating Fixed Volumes

The following example uses four available fixed disks. One disk is used for each of the required filesystems.



The following sections describe the basic process to create and mount the required partitions and filesystems using fixed disks on a RHEL host.

The steps for creating a fixed filesystem include:

- Creating the partition
- Creating the mount point
- Creating the filesystem
- Updating `fstab`
- Mounting the filesystem

Create the Fixed Filesystem for Docker

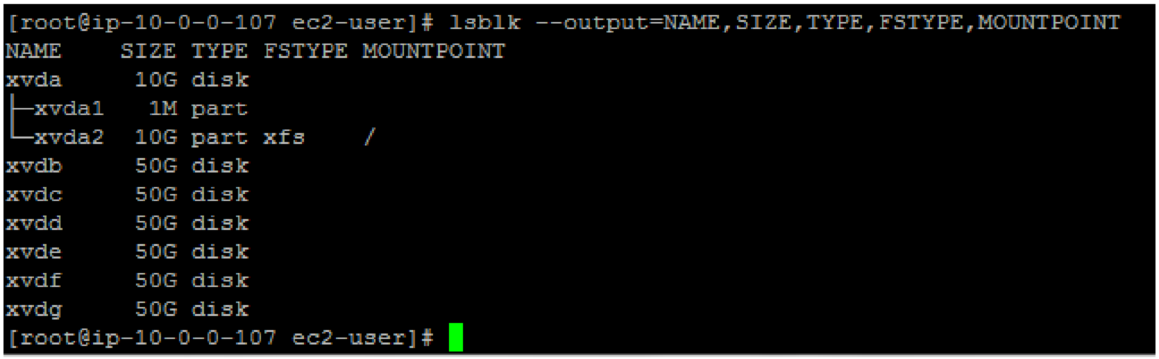
1. Determine the available disks and partitions. Use the lsblk command, for example:

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

Note: The options following the equal sign (=) define what output columns the command returns/prints.

The `lsblk` command example output shows:

- one formatted, partitioned and mounted disk (`xvda`)
- six unformatted, non-mounted disks (`xvdb` through `xvdg`)



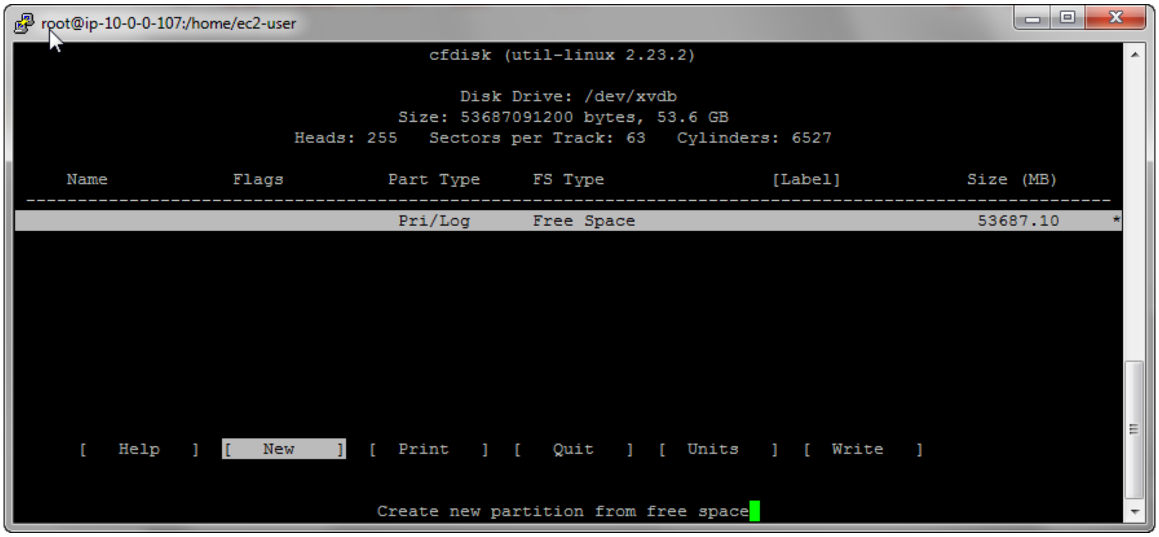
```
[root@ip-10-0-0-107 ec2-user]# lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
NAME     SIZE TYPE FSTYPE MOUNTPOINT
xvda      10G disk
├─xvda1    1M part
└─xvda2   10G part xfs    /
xvdb      50G disk
xvdc      50G disk
xvdd      50G disk
xvde      50G disk
xvdf      50G disk
xvdg      50G disk
[root@ip-10-0-0-107 ec2-user]#
```

2. Use the `cfdisk` utility to create the partition, define the size and set the type.
   Note: For basic instructions on using the `cfdisk` interface, see the section called *Using the cfdisk Tool*, below.
   a. Switch to a user with *root* privileges, for example:

```
sudo su
```

   b. Launch `cfdisk` and edit the drive, for example:

```
cfdisk /dev/xvdb
```



```
                              cfdisk (util-linux 2.23.2)

                                 Disk Drive: /dev/xvdb
                           Size: 53687091200 bytes, 53.6 GB
                     Heads: 255   Sectors per Track: 63   Cylinders: 6527

    Name          Flags          Part Type    FS Type            [Label]           Size (MB)
 ----------------------------------------------------------------------------------------------
                                  Pri/Log      Free Space                           53687.10  *









      [  Help  ]  [  New   ]  [  Print  ]  [  Quit  ]  [  Units  ]  [  Write  ]


                            Create new partition from free space
```

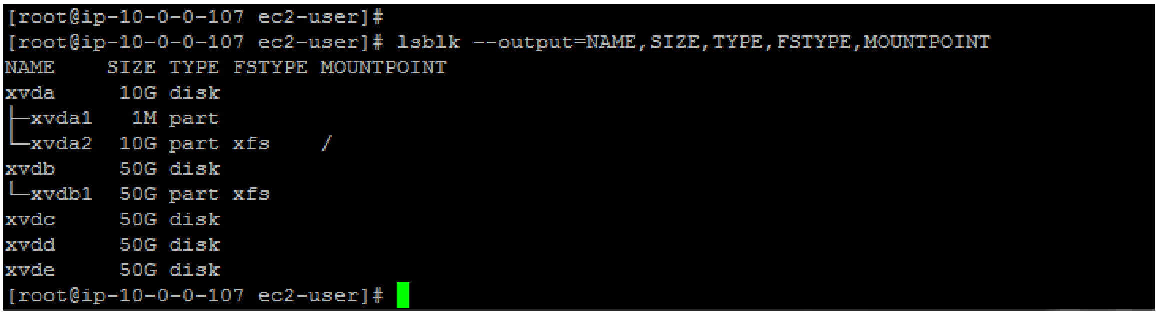
   c. Create the *partition.*
   Highlight **New**.
   Press **Enter**.
   d. Set the type (*Primary* (default) or *Logical*):
   Highlight **Primary.**
   Press **Enter.**

e. Set the *size*, for example, to use all available space:
   Press **Enter.**
f. Write the changes to the disk:
   Highlight **Write.**
   Press **Enter** to write to the disk.
g. Answer **yes** to proceed with format.
h. Quit the cfdisk utility:
   Select **Quit.**
   Press **Enter** to exit the tool.

3. Verify the new partition.

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

The new partition *vxdb1* displays in the output.



```
[root@ip-10-0-0-107 ec2-user]#
[root@ip-10-0-0-107 ec2-user]# lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
NAME    SIZE TYPE FSTYPE MOUNTPOINT
xvda     10G disk
├─xvda1   1M part
└─xvda2  10G part xfs    /
xvdb     50G disk
└─xvdb1  50G part xfs
xvdc     50G disk
xvdd     50G disk
xvde     50G disk
[root@ip-10-0-0-107 ec2-user]#
```

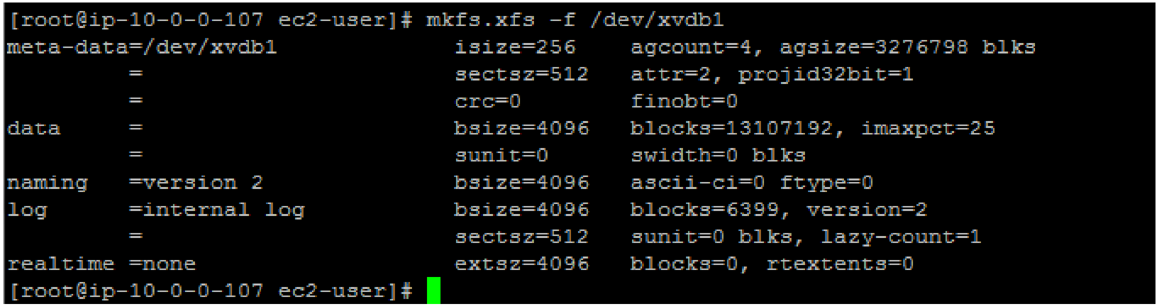
4. Create the mount point, for example:

```
mkdir -p /var/lib/docker
```

5. Create the file system. Use the `mkfs` command to format the partition. For example, to create an *xfs* file system, issue the following command:

```
mkfs.xfs -f /dev/xvdb1
```

The following example shows the output from the `mkfs` command:



```
[root@ip-10-0-0-107 ec2-user]# mkfs.xfs -f /dev/xvdb1
meta-data=/dev/xvdb1             isize=256    agcount=4, agsize=3276798 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=0        finobt=0
data     =                       bsize=4096   blocks=13107192, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=0
log      =internal log           bsize=4096   blocks=6399, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[root@ip-10-0-0-107 ec2-user]#
```

6. Add the new partition/filesystem information to the `/etc/fstab` file. Edit `the /etc/fstab` file to append the new line. For example:

   a. Open `/etc/fstab` in a text editor.

b. Enter the new information. For example, add the following line:

```
/dev/xvdb1 /var/lib/docker xfs defaults 0 0
```

Note: The fields in the fstab file are:

```
DeviceName  MountPoint  FileSystemType MountOptions  DumbOptions FileSystemCheckOptions(fsck)
```

c. Save and exit the file.

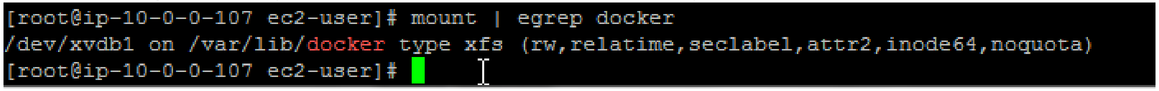
7. Mount the partition. For example, to mount all unmounted partitions:

```
mount -a
```

8. Verify the partition mounted correctly. For example, use following command and consult the output:

```
mount | egrep docker
```

The following shows a positive result in the command output:



```
[root@ip-10-0-0-107 ec2-user]# mount | egrep docker
/dev/xvdb1 on /var/lib/docker type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
[root@ip-10-0-0-107 ec2-user]#
```

Create the Fixed Filesystem for Control Center

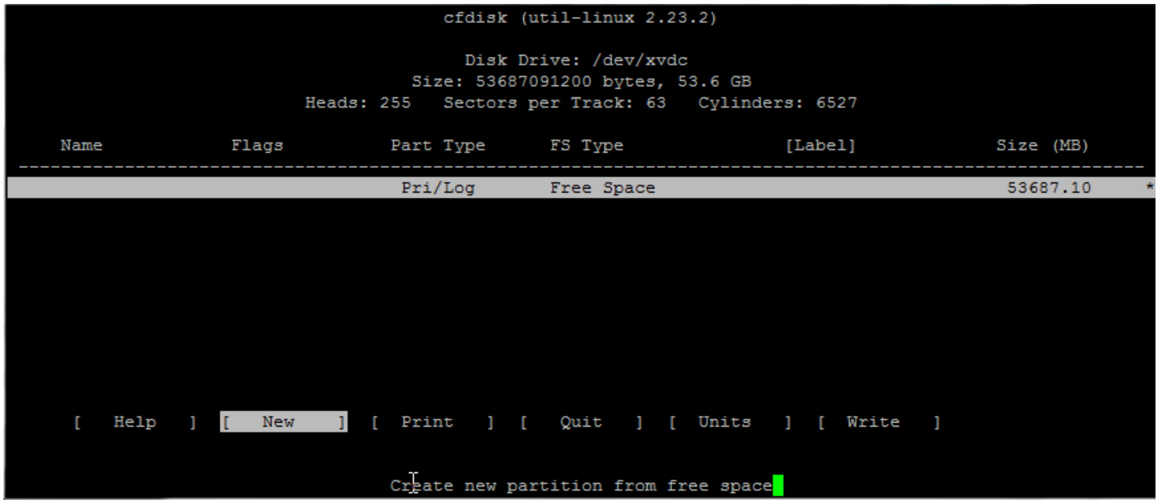
1. Use the `cfdisk` utility to create the partition, define the size and set the type.

a. Switch to a user with *root* privileges, for example:

```
sudo su
```

b. Launch `cfdisk` and edit the drive, for example:

```
cfdisk /dev/xvdc
```



```
                          cfdisk (util-linux 2.23.2)

                            Disk Drive: /dev/xvdc
                     Size: 53687091200 bytes, 53.6 GB
              Heads: 255   Sectors per Track: 63   Cylinders: 6527

   Name            Flags          Part Type     FS Type              [Label]          Size (MB)
 ------------------------------------------------------------------------------------------------
                                   Pri/Log       Free Space                            53687.10   *
```

```
                         Create new partition from free space
```

c. Create the *partition.*
Highlight **New**.
Press **Enter**.

d. Set the type (*Primary* (default) or *Logical*):
Highlight **Primary.**
Press **Enter.**

e. Set the *size*, for example, to use all available space:
Press **Enter.**

f. Write the changes to the disk:
Highlight **Write.**
Press **Enter** to write to the disk.

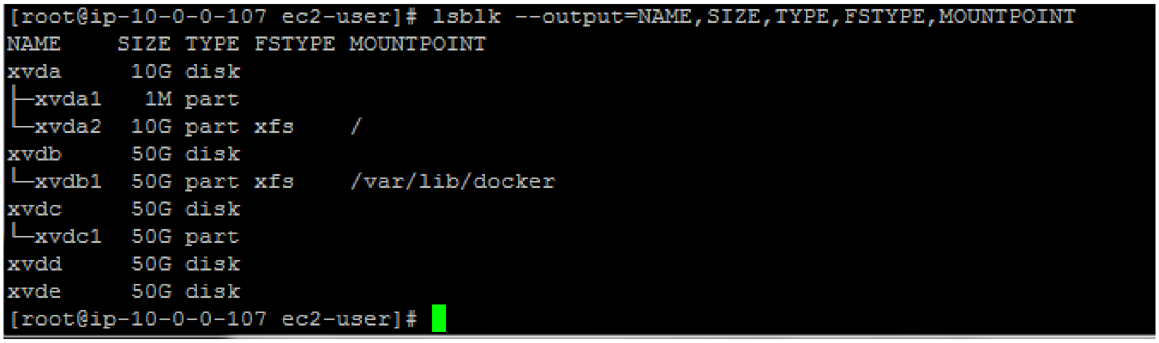
g. Answer **yes** to proceed with format.

h. Quit the cfdisk utility:
Select **Quit.**
Press **Enter** to exit the tool.

2. Verify the new partition:

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

The new partition *vxdc1* displays in the output.



```
[root@ip-10-0-0-107 ec2-user]# lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
NAME    SIZE TYPE FSTYPE MOUNTPOINT
xvda     10G disk
├─xvda1   1M part
└─xvda2  10G part xfs    /
xvdb     50G disk
└─xvdb1  50G part xfs    /var/lib/docker
xvdc     50G disk
└─xvdc1  50G part
xvdd     50G disk
xvde     50G disk
[root@ip-10-0-0-107 ec2-user]#
```

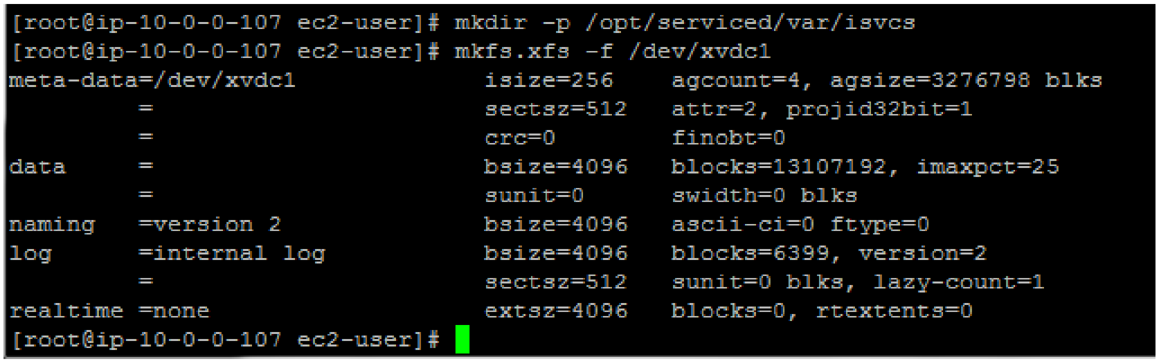
3. Create the mount point, for example:

```
mkdir -p /opt/serviced/var/isvcs
```

4. Create the file system. Use the `mkfs` command to format the partition. For example, to create an *xfs* file system, enter the following command:

```
mkfs.xfs -f /dev/xvdc1
```

The following example shows the `mkfs` command output:



```
[root@ip-10-0-0-107 ec2-user]# mkdir -p /opt/serviced/var/isvcs
[root@ip-10-0-0-107 ec2-user]# mkfs.xfs -f /dev/xvdc1
meta-data=/dev/xvdc1             isize=256    agcount=4, agsize=3276798 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=0        finobt=0
data     =                       bsize=4096   blocks=13107192, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=0
log      =internal log           bsize=4096   blocks=6399, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[root@ip-10-0-0-107 ec2-user]#
```

5. Add the new partition/filesystem information to the `/etc/fstab` file.
   Edit the `/etc/fstab` file to append the new line. For example:

   a. Open `/etc/fstab` in a text editor.
   b. Enter the new information For example, add the following line:

   ```
   /dev/xvdc1 /opt/serviced/var/isvcs xfs defaults 0 0
   ```

   c. Save and exit the file.

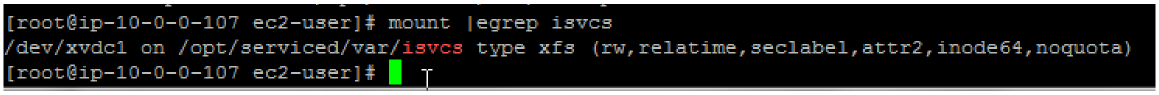
6. Mount the partition. For example, to mount all unmounted partitions:

```
mount -a
```

7. Verify the partition mounted correctly. For example:

```
mount | egrep isvcs
```

The following shows a positive result in the command output:



```
[root@ip-10-0-0-107 ec2-user]# mount |egrep isvcs
/dev/xvdc1 on /opt/serviced/var/isvcs type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
[root@ip-10-0-0-107 ec2-user]#
```

Create the Fixed Filesystem for DFS

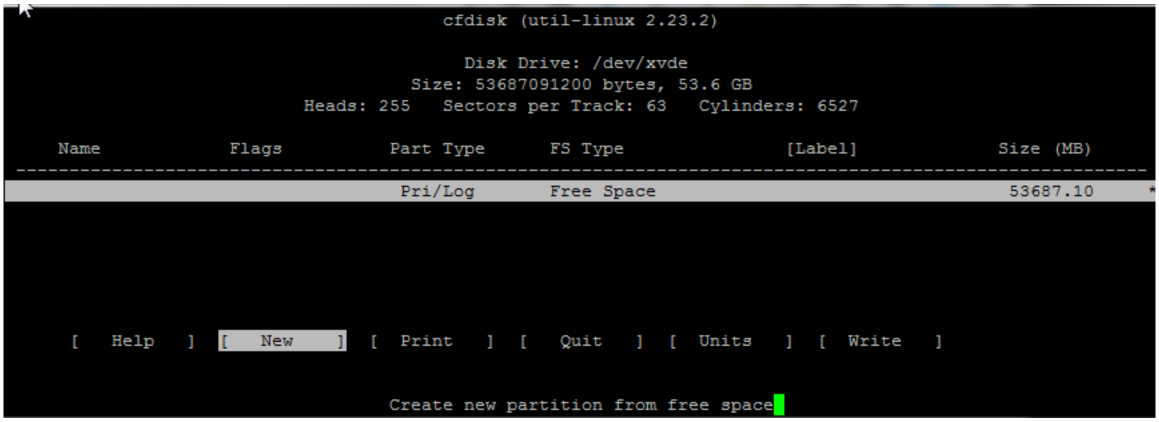
Use the `cfdisk` utility to create the partition, define the size and set the type.

1. Switch to a user with *root* privileges, for example:

   ```
   sudo su
   ```

2. Launch `cfdisk` and edit the drive, for example:

```
cfdisk /dev/xvdd
```



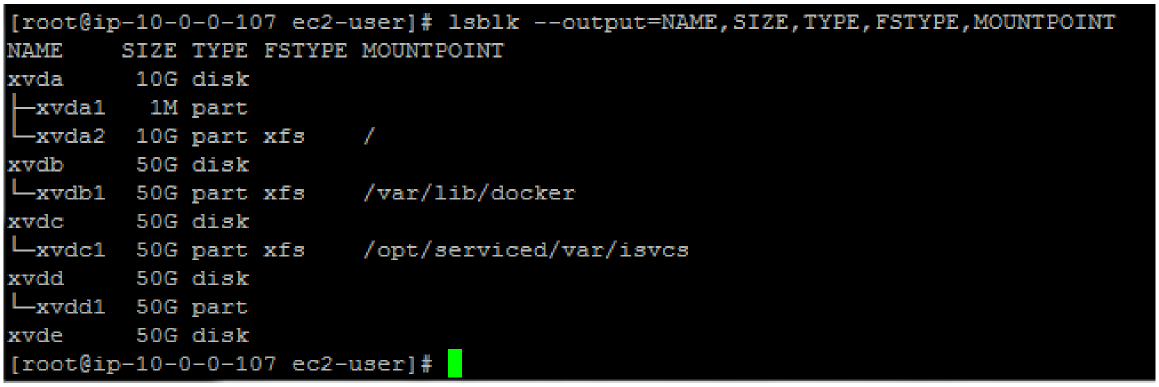
```
                              cfdisk (util-linux 2.23.2)

                                Disk Drive: /dev/xvde
                        Size: 53687091200 bytes, 53.6 GB
                 Heads: 255   Sectors per Track: 63   Cylinders: 6527

   Name            Flags          Part Type      FS Type            [Label]          Size (MB)
 -----------------------------------------------------------------------------------------------
                                   Pri/Log        Free Space                          53687.10    *




    [   Help   ]  [   New    ]  [  Print   ]  [   Quit   ]  [  Units   ]  [  Write   ]

                             Create new partition from free space
```

a. Create the *partition.*
   Highlight **New**.
   Press **Enter**.
b. Set the type (*Primary* (default) or *Logical*):
   Highlight **Primary.**
   Press **Enter.**
c. Set the *size*, for example, to use all available space:
   Press **Enter.**
d. Write the changes to the disk:
   Highlight **Write.**
   Press **Enter** to write to the disk.
e. Answer **yes** to proceed with format.
f. Quit the cfdisk utility:
   Select **Quit.**
   Press **Enter** to exit the tool.

2. Verify the new partition:

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

The new partition *vxdd1* displays in the output:



```
[root@ip-10-0-0-107 ec2-user]# lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
NAME    SIZE TYPE FSTYPE MOUNTPOINT
xvda     10G disk
├─xvda1   1M part
└─xvda2  10G part xfs    /
xvdb     50G disk
└─xvdb1  50G part xfs    /var/lib/docker
xvdc     50G disk
└─xvdc1  50G part xfs    /opt/serviced/var/isvcs
xvdd     50G disk
└─xvdd1  50G part
xvde     50G disk
[root@ip-10-0-0-107 ec2-user]#
```

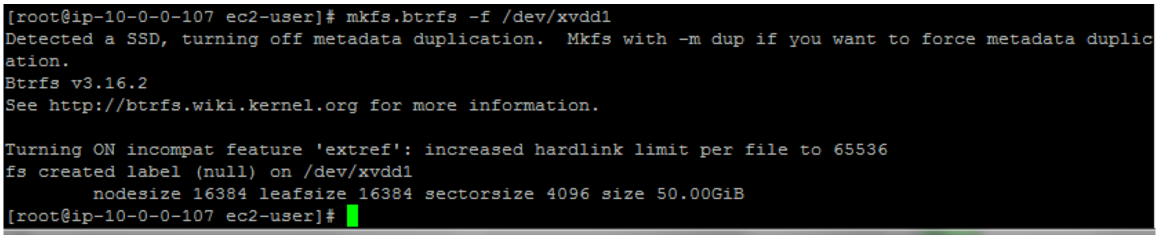
3. Create the mount point, for example:

```
mkdir -p /opt/serviced/var/volumes
```

4. Create the file system. Use the `mkfs` command to format the partition. For example, to create an *btrfs* file system:

```
mkfs.btrfs -f /dev/xvdd1
```

The following example shows the `mkfs` command output:



```
[root@ip-10-0-0-107 ec2-user]# mkfs.btrfs -f /dev/xvdd1
Detected a SSD, turning off metadata duplication.  Mkfs with -m dup if you want to force metadata duplic
ation.
Btrfs v3.16.2
See http://btrfs.wiki.kernel.org for more information.

Turning ON incompat feature 'extref': increased hardlink limit per file to 65536
fs created label (null) on /dev/xvdd1
        nodesize 16384 leafsize 16384 sectorsize 4096 size 50.00GiB
[root@ip-10-0-0-107 ec2-user]#
```

5. Add the new partition/filesystem information to the `/etc/fstab` file.
Edit the `/etc/fstab` file to append the new line. For example:

   a. Open `/etc/fstab` in a text editor.
   b. Enter the new information. For example, add the following line:

```
/dev/xvdd1 /opt/serviced/var/volumes btrfs rw,noatime,nodatacow,skip_balance 0 0
```

   c. Save and exit the file.

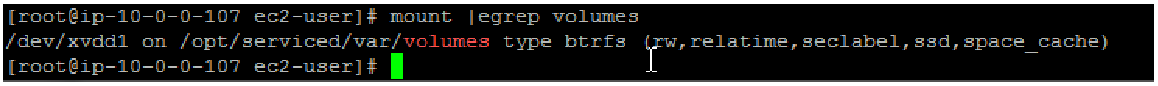
6. Mount the partition. For example, to mount all unmounted partitions:

```
mount -a
```

7. Verify the partition mounted correctly. For example:

```
mount | egrep volumes
```

The following shows a positive result in the command output:



```
[root@ip-10-0-0-107 ec2-user]# mount |egrep volumes
/dev/xvdd1 on /opt/serviced/var/volumes type btrfs (rw,relatime,seclabel,ssd,space_cache)
[root@ip-10-0-0-107 ec2-user]#
```

Note: If you need to determine the free space available on a btrfs filesystem, use the `btrfs` command, for example:

```
btrfs filesystem df /opt/serviced/var/volumes
```

Create the Fixed Filesystem for Backup

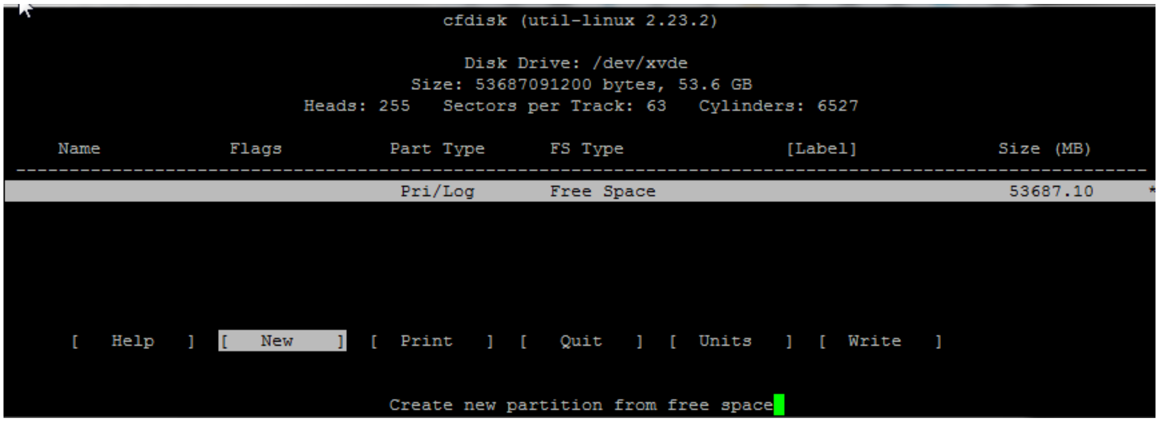
Use the `cfdisk` utility to create the partition, define the size and set the type.

1. Switch to a user with *root* privileges, for example:

```
sudo su
```

2. Launch `cfdisk` and edit the drive, for example:

```
cfdisk /dev/xvde
```



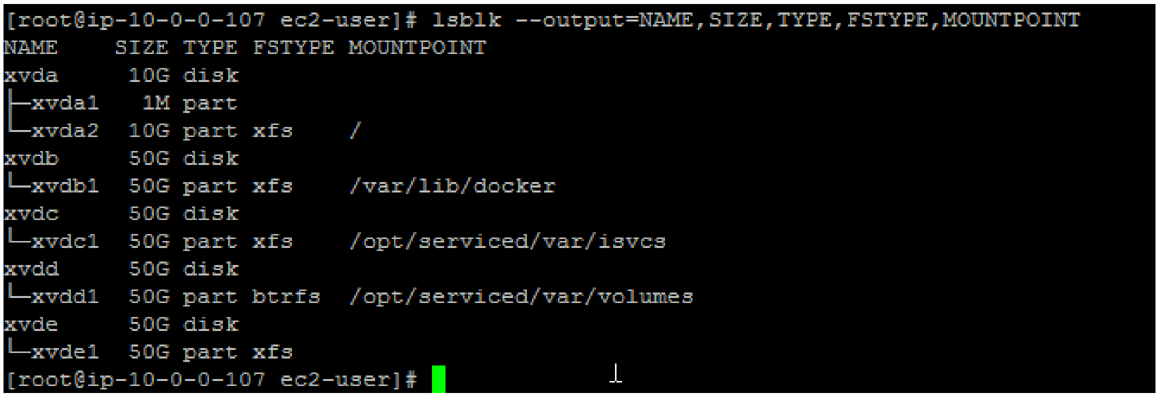
```
                         cfdisk (util-linux 2.23.2)

                          Disk Drive: /dev/xvde
                     Size: 53687091200 bytes, 53.6 GB
              Heads: 255   Sectors per Track: 63   Cylinders: 6527

   Name          Flags          Part Type     FS Type           [Label]           Size (MB)
 ----------------------------------------------------------------------------------------------
                                 Pri/Log      Free Space                           53687.10   *





   [  Help  ]  [  New   ]  [  Print  ]  [  Quit  ]  [  Units  ]  [  Write  ]

                     Create new partition from free space
```

   a. Create the *partition.*
      Highlight **New**.
      Press **Enter**.
   b. Set the type (*Primary* (default) or *Logical*):
      Highlight **Primary.**
      Press **Enter.**
   c. Set the *size*, for example, to use all available space:
      Press **Enter.**
   d. Write the changes to the disk:
      Highlight **Write.**
      Press **Enter** to write to the disk.
   e. Answer **yes** to proceed with format.
   f. Quit the cfdisk utility:
      Select **Quit.**
      Press **Enter** to exit the tool.

3. Verify the new partition. Issue the following command:

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

The new partition *vxde1* displays in the output.



```
[root@ip-10-0-0-107 ec2-user]# lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
NAME     SIZE TYPE FSTYPE MOUNTPOINT
xvda      10G disk
├─xvda1    1M part
└─xvda2  10G part xfs    /
xvdb      50G disk
└─xvdb1  50G part xfs    /var/lib/docker
xvdc      50G disk
└─xvdc1  50G part xfs    /opt/serviced/var/isvcs
xvdd      50G disk
└─xvdd1  50G part btrfs  /opt/serviced/var/volumes
xvde      50G disk
└─xvde1  50G part xfs
[root@ip-10-0-0-107 ec2-user]#
```

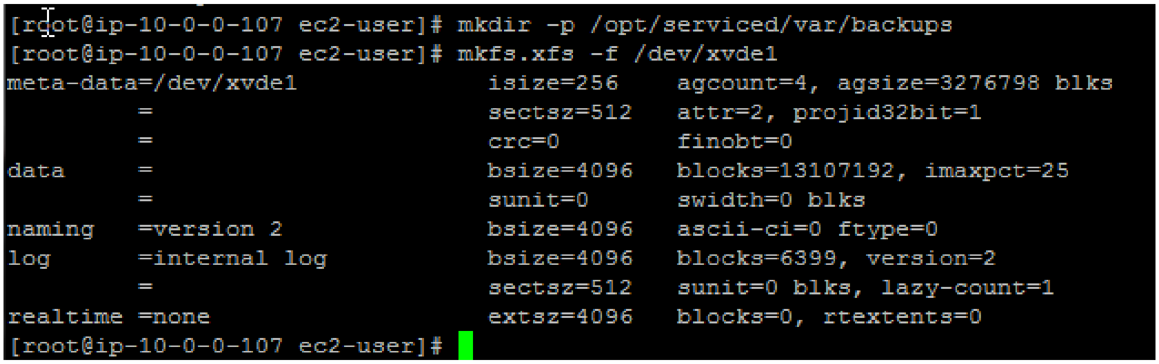
4. Create the mount point, for example:

```
mkdir -p /opt/serviced/var/backups
```

5. Create the file system. Use the `mkfs` command to format the partition. For example, to create an *xfs* file system issue the following command:

```
mkfs.xfs -f /dev/xvde1
```

The following example shows the `mkfs` command output:



```
[root@ip-10-0-0-107 ec2-user]# mkdir -p /opt/serviced/var/backups
[root@ip-10-0-0-107 ec2-user]# mkfs.xfs -f /dev/xvde1
meta-data=/dev/xvde1             isize=256    agcount=4, agsize=3276798 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=0        finobt=0
data     =                       bsize=4096   blocks=13107192, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=0
log      =internal log           bsize=4096   blocks=6399, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[root@ip-10-0-0-107 ec2-user]#
```

6. Add the new partition/filesystem information to the `/etc/fstab` file.
Edit the `/etc/fstab` file to append the new line. For example:

   a. Open `/etc/fstab` in a text editor.
   b. Enter the new information. For example, add the following line:

   ```
   /dev/xvde1 /opt/serviced/var/backups xfs defaults 0 0
   ```

   c. Save and exit the file.

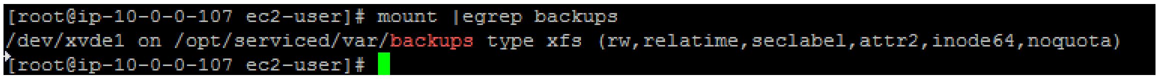
7. Mount the partition. For example to mount all unmounted partitions:

```
mount -a
```

8. Verify the partition mounted correctly. For example:

```
mount | egrep backups
```

The following shows a positive result in the command output:

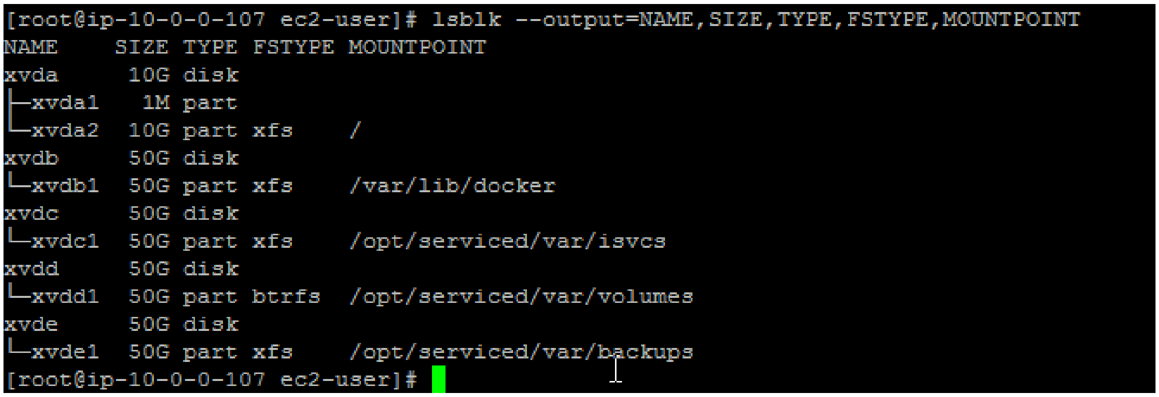


```
[root@ip-10-0-0-107 ec2-user]# mount |egrep backups
/dev/xvde1 on /opt/serviced/var/backups type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
[root@ip-10-0-0-107 ec2-user]#
```

Completion - Fixed Volume Creation

The following output from the `lsblk` command shows the successful creation, partitioning and mounting of the four required filesystems, each on a fixed volume filesystem:

- *Docker* on single disk (*xvdb*), mounted on `/var/lib/docker`.
- *Control Center*, on single disk (*xvdc*), mounted on `/opt/serviced/var/isvcs`.
- *DFS* on single disk (*xvdd*), mounted on `/opt/serviced/var/volumes`.
- *Backups*, on single disk (*xvde*), mounted on `/opt/serviced/var/backups`.

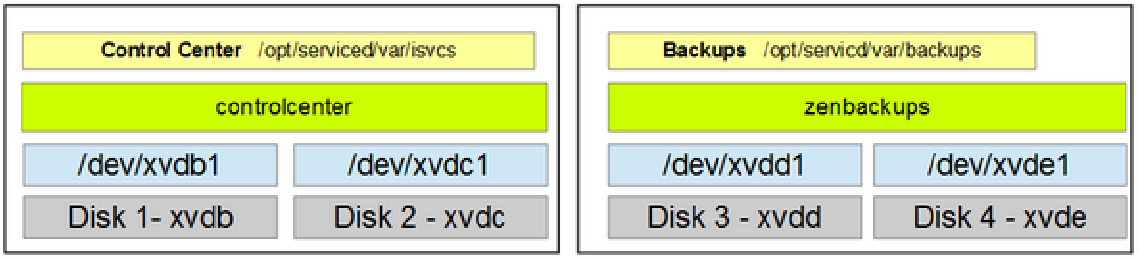


```
[root@ip-10-0-0-107 ec2-user]# lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
NAME     SIZE TYPE FSTYPE MOUNTPOINT
xvda      10G disk
├─xvda1    1M part
└─xvda2   10G part xfs    /
xvdb      50G disk
└─xvdb1   50G part xfs    /var/lib/docker
xvdc      50G disk
└─xvdc1   50G part xfs    /opt/serviced/var/isvcs
xvdd      50G disk
└─xvdd1   50G part btrfs  /opt/serviced/var/volumes
xvde      50G disk
└─xvde1   50G part xfs    /opt/serviced/var/backups
[root@ip-10-0-0-107 ec2-user]#
```

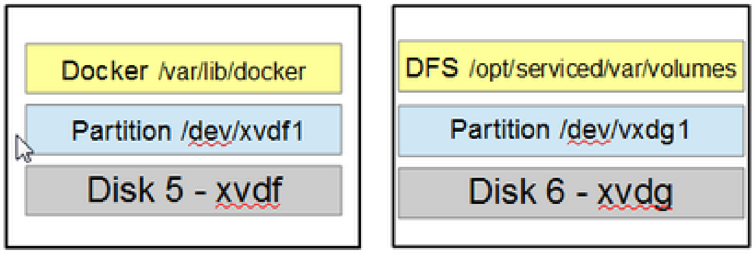
Logical Volume Creation Example

The following example requires 6 disks:

- Four disks, two each for the logical volumes *Control Center* and *Backups*:



- Two fixed disks, one each for *Docker* and *DFS:*



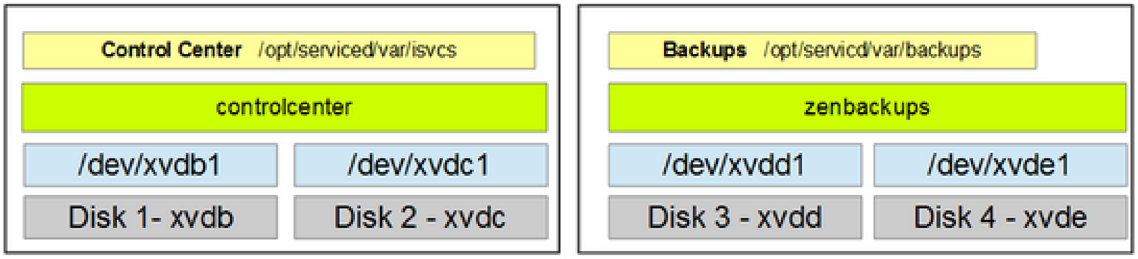
Creating Logical and Fixed Volumes

The steps to creating logical volumes include:

- Planning your storage layout to determine the number of disks required.
- Verifying that the LVM tools are installed on the operating system. The tools might not be installed by default.
- Preparing the disks for use with LVM.
- Creating *v*olume groups that define which physical volumes are used for each volume group.
- Creating logical volume(s) within each volume group.
- Creating the filesystem for each logical volume.

Note: A volume group is a collection of physical volumes and a physical volume can only live in a single volume group.

The following figure shows an example set of logical volumes that uses four physical discs to create two logical volumes (*isvcs* and *backups*):



Filesystem Scenario for Installation

A common filesystem scenario for installing Resource Manager and Control Center is:

- Create logical volumes so space requirements can be more easily managed for:
    - *Control Center* (`/opt/serviced/var/isvcs`)
    - *Backups* (`/opt/serviced/var/backups`).
- Create the remaining filesystems as fixed filesystems that do not require extensive elastic storage:
    - *Docker* (`/var/lib/docker`)
    - *DFS* (`/opt/serviced/var/volumes`)

Verify that the LVM Tools are Installed

- Verify that the LVM tools are installed on the operating system. Issue the following command:

```
rpm -qa |grep -i lvm
```

If the result includes the following, the basic LVM tools are installed:

```
lvm2-libs-2.02.115-3.el7_1.1.x86_64
```

and

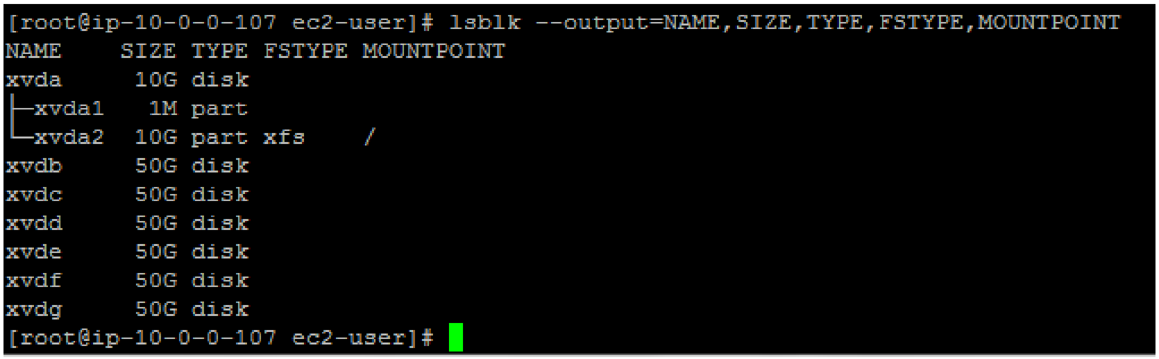
```
lvm2-2.02.115-3.el7_1.1.x86_64
```

If the tools are not installed, see the section titled Appendix: Installing the LVM Tools.

View Disks and Partitions

Use the `lsblk` command to determine available disks and partitions. For example:

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```



```
[root@ip-10-0-0-107 ec2-user]# lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
NAME     SIZE TYPE FSTYPE MOUNTPOINT
xvda      10G disk
├─xvda1    1M part
└─xvda2   10G part xfs    /
xvdb      50G disk
xvdc      50G disk
xvdd      50G disk
xvde      50G disk
xvdf      50G disk
xvdg      50G disk
[root@ip-10-0-0-107 ec2-user]#
```

The output from the command shows six disks (*xvdb* through *xvdg*) available.
Note: The options following the equal sign (=) define what output columns the `lsblk` command returns/prints.

Prepare the Disks for Logical Volumes

Preparing the disks for the logical Volume Manager (LVM) includes defining partitions, if necessary, and using the `pvcreate` command to initialize the partitions/disks. This example uses partitions instead of whole disk physical volumes. This is because although LVM works well with whole disk physical volumes, other operating systems might see the disk as empty and data might be overwritten. Because of this, using whole disk physical volumes is not recommended in some cases. Using partitions to contain the physical volumes enables other operating systems to recognize the partitions

Create Partitions for Logical Volumes

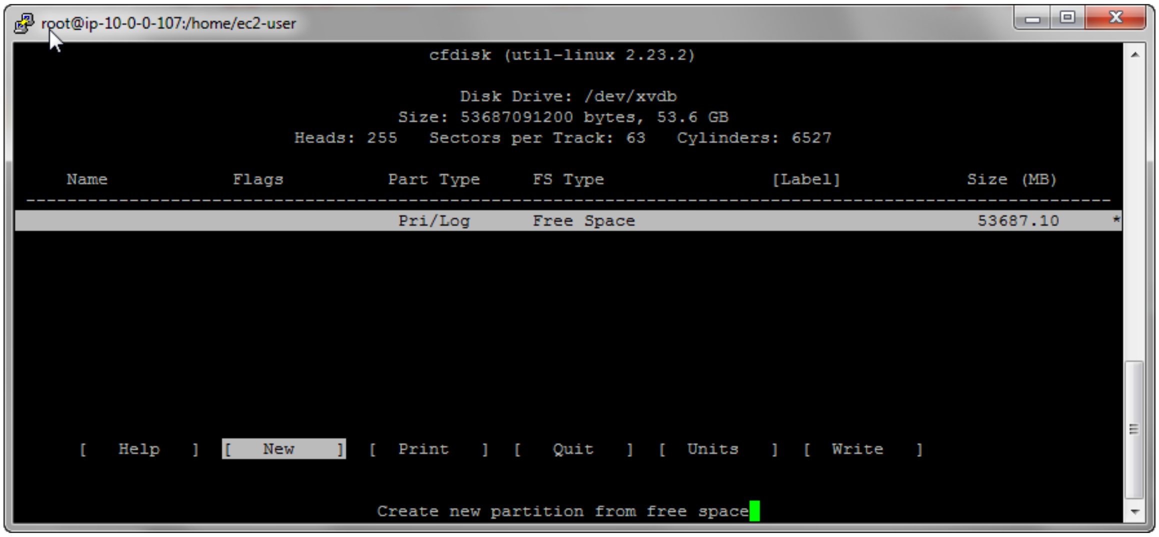
Create partitions on disks *xvdb* through *xvde:*

1. Use the `cfdisk` utility to create the partition, define the size and set the type.
   Note: For basic instructions on using the `cfdisk` interface, see the section called *Using the cfdisk Tool*, below.
   a. Switch to a user with *root* privileges, for example:

   ```
   sudo su
   ```

   b. Launch `cfdisk` and edit the drive, for example:

   ```
   cfdisk /dev/xvdb
   ```



```
                              cfdisk (util-linux 2.23.2)

                                 Disk Drive: /dev/xvdb
                           Size: 53687091200 bytes, 53.6 GB
                 Heads: 255   Sectors per Track: 63   Cylinders: 6527

    Name            Flags          Part Type    FS Type             [Label]          Size (MB)
 ---------------------------------------------------------------------------------------------
                                    Pri/Log      Free Space                           53687.10






      [   Help   ]  [   New    ]  [  Print   ]  [   Quit   ]  [  Units   ]  [  Write   ]

                            Create new partition from free space
```

c. Create the *partition.*
   Highlight **New**.
   Press **Enter**.

d. Set the type (*Primary* (default) or *Logical*):
   Highlight **Primary.**
   Press **Enter.**

e. Set the *size*, for example, to use all available space:
   Press **Enter.**

f. Write the changes to the disk:
   Highlight **Write.**
   Press **Enter** to write to the disk.

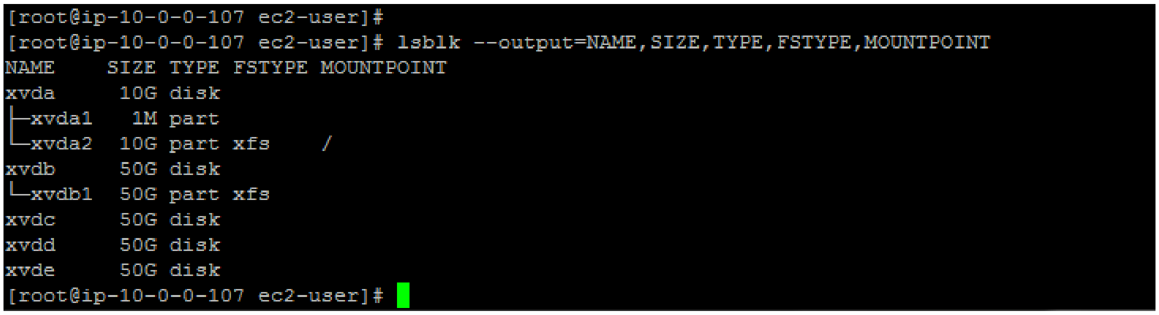
g. Answer **yes** to proceed with format.

h. Quit the cfdisk utility:
   Select **Quit.**
   Press **Enter** to exit the tool.

2. Verify the new partition.

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

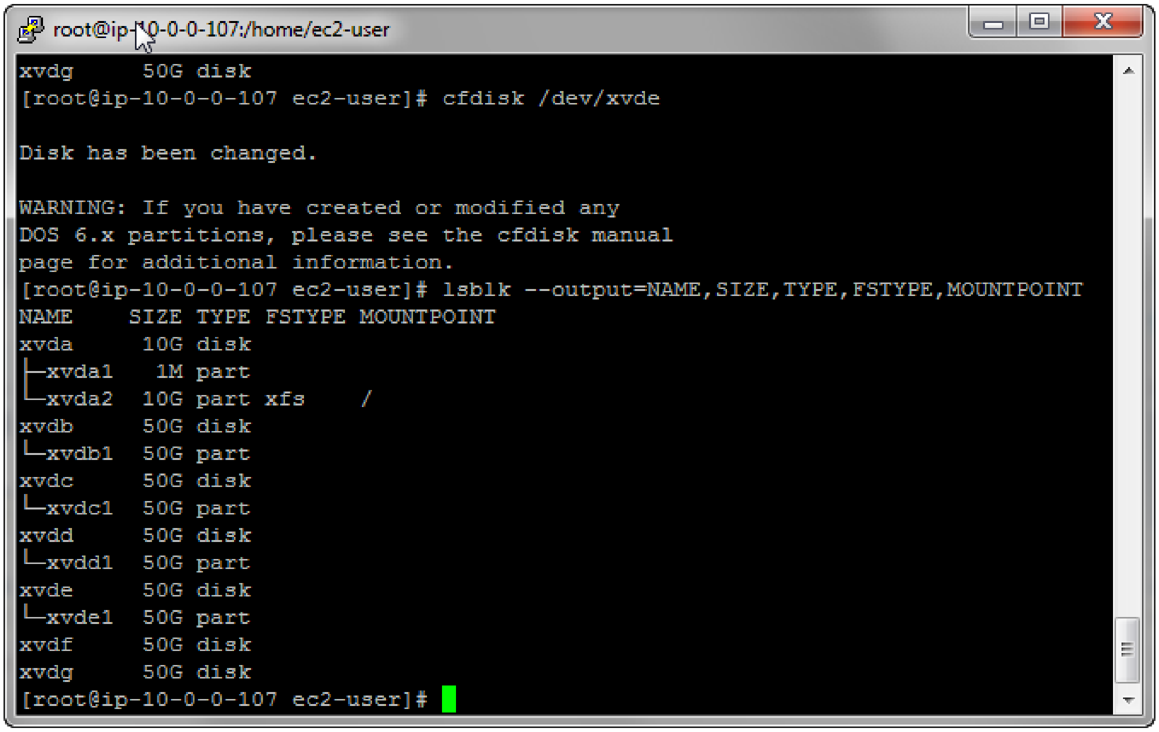
The new partition *vxdb1* displays in the output.



```
[root@ip-10-0-0-107 ec2-user]#
[root@ip-10-0-0-107 ec2-user]# lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
NAME    SIZE TYPE FSTYPE MOUNTPOINT
xvda     10G disk
├─xvda1   1M part
└─xvda2  10G part xfs    /
xvdb     50G disk
└─xvdb1  50G part xfs
xvdc     50G disk
xvdd     50G disk
xvde     50G disk
[root@ip-10-0-0-107 ec2-user]#
```

3. Repeat this Partition Creation process for the additional disks for the logical volumes. In this example, disks *xvdc, xvdd* and *xvde*.

4. Verify the four partitioned disks. In this example the new partitions are *xvdb1, xvdc1, xvdd1, xvde1*:



root@ip-10-0-0-107:/home/ec2-user

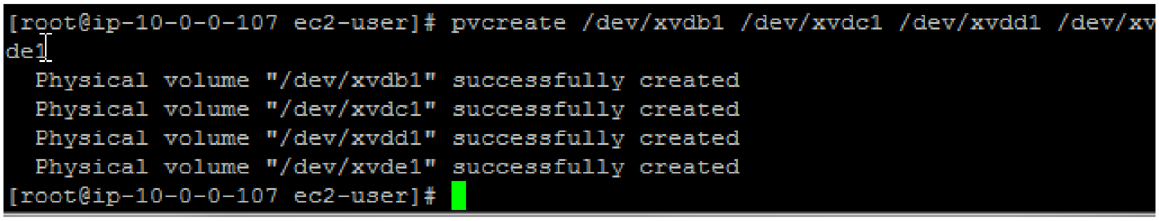
```
xvdg     50G disk
[root@ip-10-0-0-107 ec2-user]# cfdisk /dev/xvde

Disk has been changed.

WARNING: If you have created or modified any
DOS 6.x partitions, please see the cfdisk manual
page for additional information.
[root@ip-10-0-0-107 ec2-user]# lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
NAME    SIZE TYPE FSTYPE MOUNTPOINT
xvda     10G disk
├─xvda1   1M part
└─xvda2  10G part xfs    /
xvdb     50G disk
└─xvdb1  50G part
xvdc     50G disk
└─xvdc1  50G part
xvdd     50G disk
└─xvdd1  50G part
xvde     50G disk
└─xvde1  50G part
xvdf     50G disk
xvdg     50G disk
[root@ip-10-0-0-107 ec2-user]#
```

Initialize the Partitions

1. Initialize the partition/disk.
Before a disk or partition can be used as a physical volume, it must be initialized with the `pvcreate` command. This initialization creates a *volume group descriptor* at the start of the disk/partition.
For this example, create disk partitions *xvdb1* through *xvde1*:

```
pvcreate /dev/xvdb1 /dev/xvdc1 /dev/xvdd1 /dev/xvde1
```

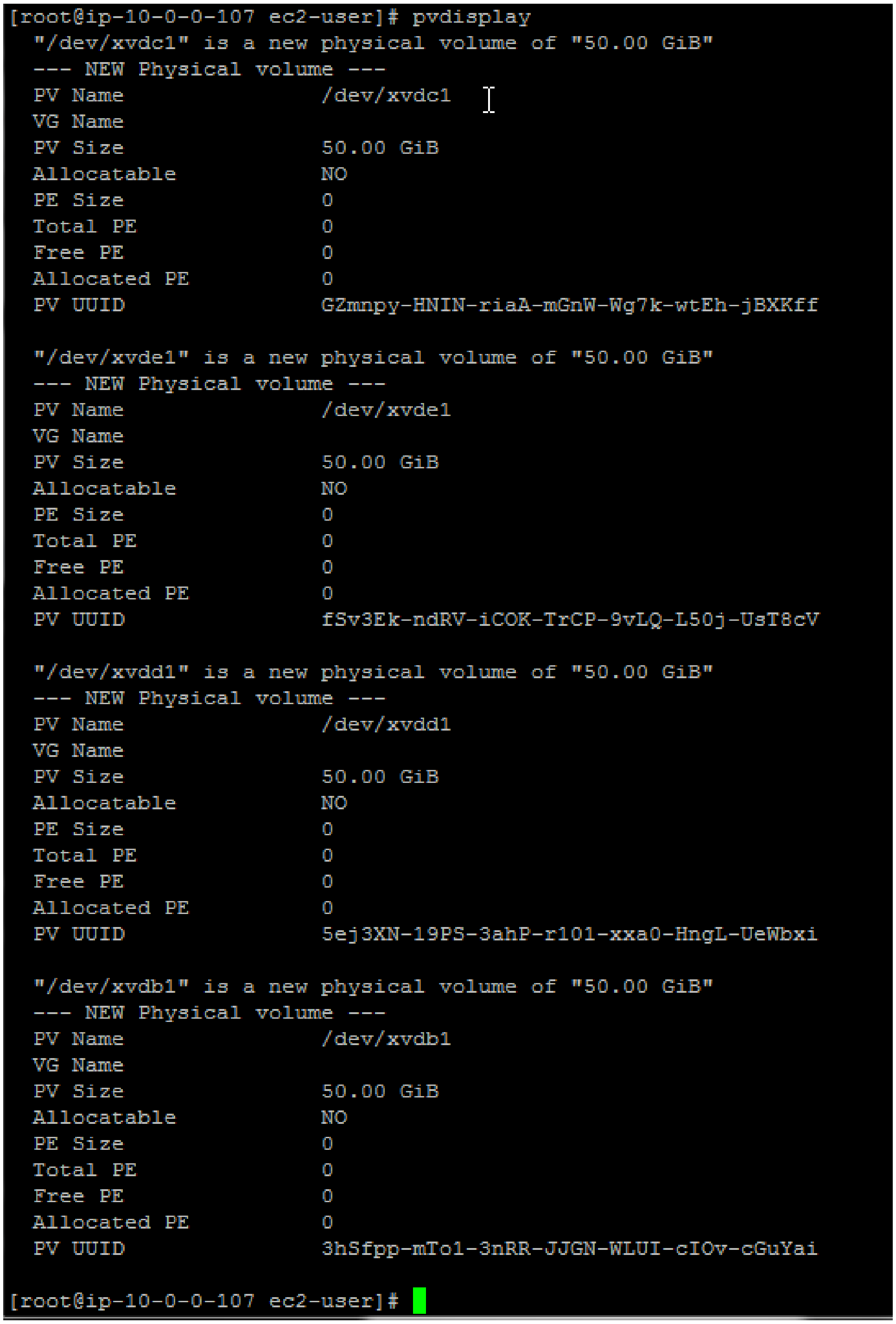
The following output shows the successful physical volume creation:



```
[root@ip-10-0-0-107 ec2-user]# pvcreate /dev/xvdb1 /dev/xvdc1 /dev/xvdd1 /dev/xv
de1
  Physical volume "/dev/xvdb1" successfully created
  Physical volume "/dev/xvdc1" successfully created
  Physical volume "/dev/xvdd1" successfully created
  Physical volume "/dev/xvde1" successfully created
[root@ip-10-0-0-107 ec2-user]#
```

2. View the results:

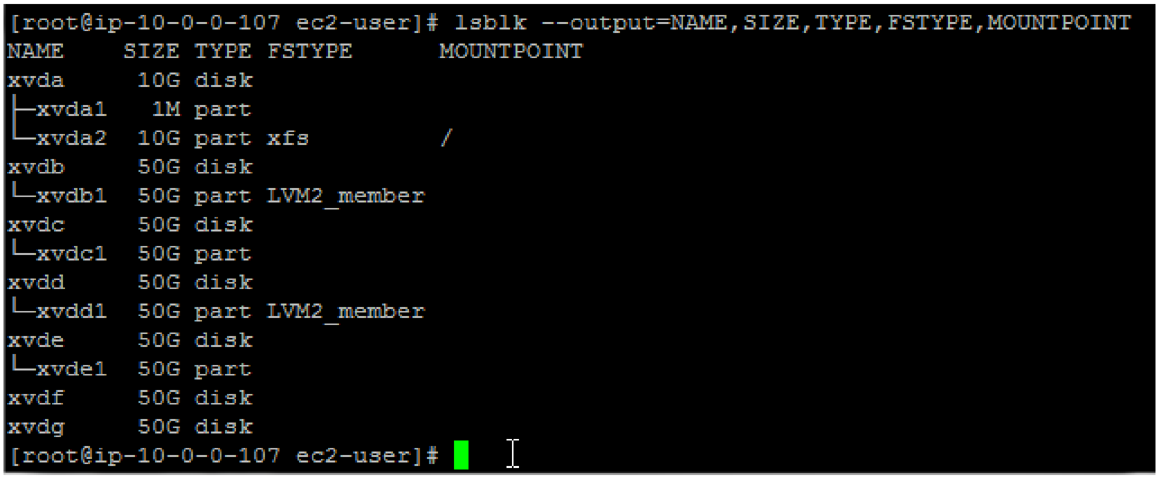
```
pvdisplay
```



```
[root@ip-10-0-0-107 ec2-user]# pvdisplay
  "/dev/xvdc1" is a new physical volume of "50.00 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/xvdc1
  VG Name
  PV Size               50.00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               GZmnpy-HNIN-riaA-mGnW-Wg7k-wtEh-jBXKff

  "/dev/xvde1" is a new physical volume of "50.00 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/xvde1
  VG Name
  PV Size               50.00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               fSv3Ek-ndRV-iCOK-TrCP-9vLQ-L50j-UsT8cV

  "/dev/xvdd1" is a new physical volume of "50.00 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/xvdd1
  VG Name
  PV Size               50.00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               5ej3XN-19PS-3ahP-r101-xxa0-HngL-UeWbxi

  "/dev/xvdb1" is a new physical volume of "50.00 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/xvdb1
  VG Name
  PV Size               50.00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               3hSfpp-mTo1-3nRR-JJGN-WLUI-cIOv-cGuYai

[root@ip-10-0-0-107 ec2-user]#
```

Use the `lsblk` command to view the disk partitions and the logical volume membership(s):

```
Lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```



```
[root@ip-10-0-0-107 ec2-user]# lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
NAME     SIZE TYPE FSTYPE      MOUNTPOINT
xvda      10G disk
├─xvda1    1M part
└─xvda2   10G part xfs         /
xvdb      50G disk
└─xvdb1   50G part LVM2_member
xvdc      50G disk
└─xvdc1   50G part
xvdd      50G disk
└─xvdd1   50G part LVM2_member
xvde      50G disk
└─xvde1   50G part
xvdf      50G disk
xvdg      50G disk
[root@ip-10-0-0-107 ec2-user]#
```

The results show that for this example, there are two LVM2_member groups - *xvdb1* and *xvdc1* belong to one, and *xvdd1* and *xvde1* belong to the second group.

Create the Logical Volume for Control Center

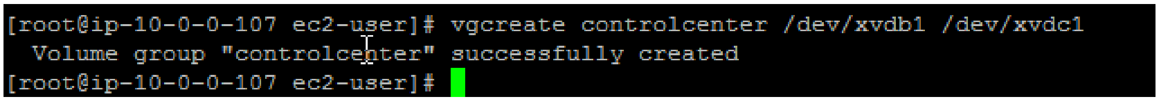
The steps for creating the logical volume include:

- Creating the volume group that defines which physical volumes are used
- Creating the *logical volume* within the volume group
- Creating the mount point
- Creating the filesystem for the logical volume
- Mounting the filesystem
- Updating the `fstab` file

Create the controlcenter Volume Group

1. Create the *controlcenter* volume group and add the first two disk partitions (*vxdb1* & *vxdc1*) to it, for example:

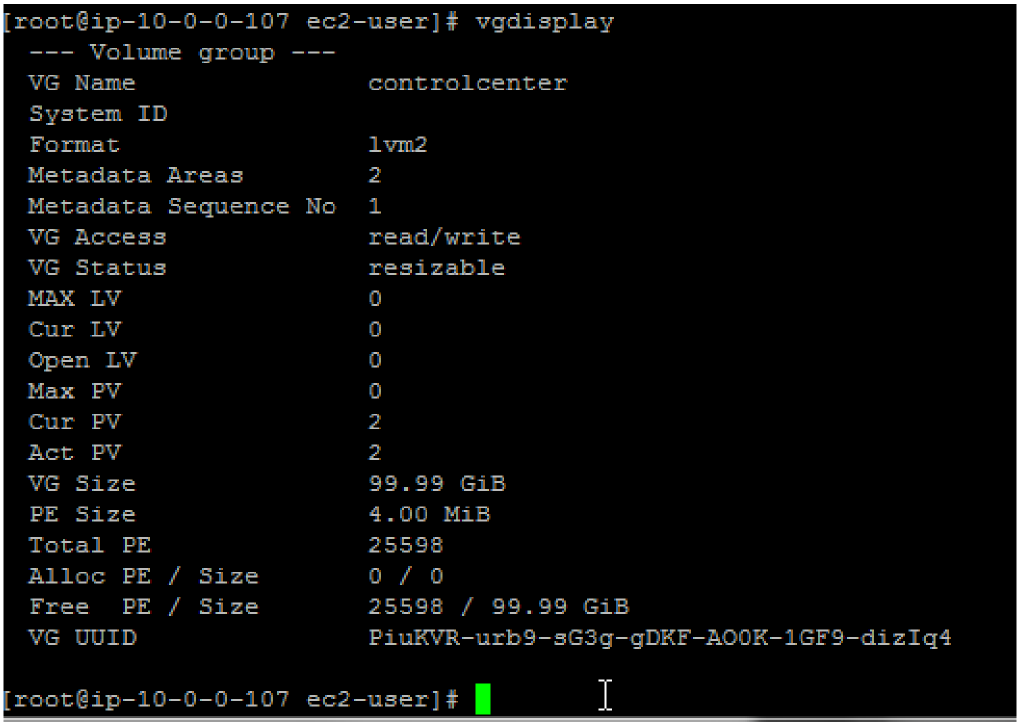
```
vgcreate controlcenter /dev/xvdb1 /dev/xvdc1
```



```
[root@ip-10-0-0-107 ec2-user]# vgcreate controlcenter /dev/xvdb1 /dev/xvdc1
  Volume group "controlcenter" successfully created
[root@ip-10-0-0-107 ec2-user]#
```

2. Display the volume group information:

```
vgdisplay
```

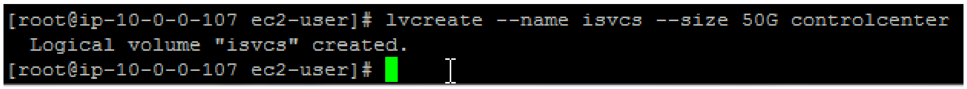


```
[root@ip-10-0-0-107 ec2-user]# vgdisplay
  --- Volume group ---
  VG Name               controlcenter
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               99.99 GiB
  PE Size               4.00 MiB
  Total PE              25598
  Alloc PE / Size       0 / 0
  Free  PE / Size       25598 / 99.99 GiB
  VG UUID               PiuKVR-urb9-sG3g-gDKF-AO0K-1GF9-dizIq4

[root@ip-10-0-0-107 ec2-user]#
```

Create the isvcs Logical Volume

Create the logical volume. For example to create a logical volume with the name *isvcs*, a size of *50GB*, that resides in the volume group *controlcenter*:

```
lvcreate --name isvcs --size 50G controlcenter
```



```
[root@ip-10-0-0-107 ec2-user]# lvcreate --name isvcs --size 50G controlcenter
  Logical volume "isvcs" created.
[root@ip-10-0-0-107 ec2-user]#
```

Create the Mount Point

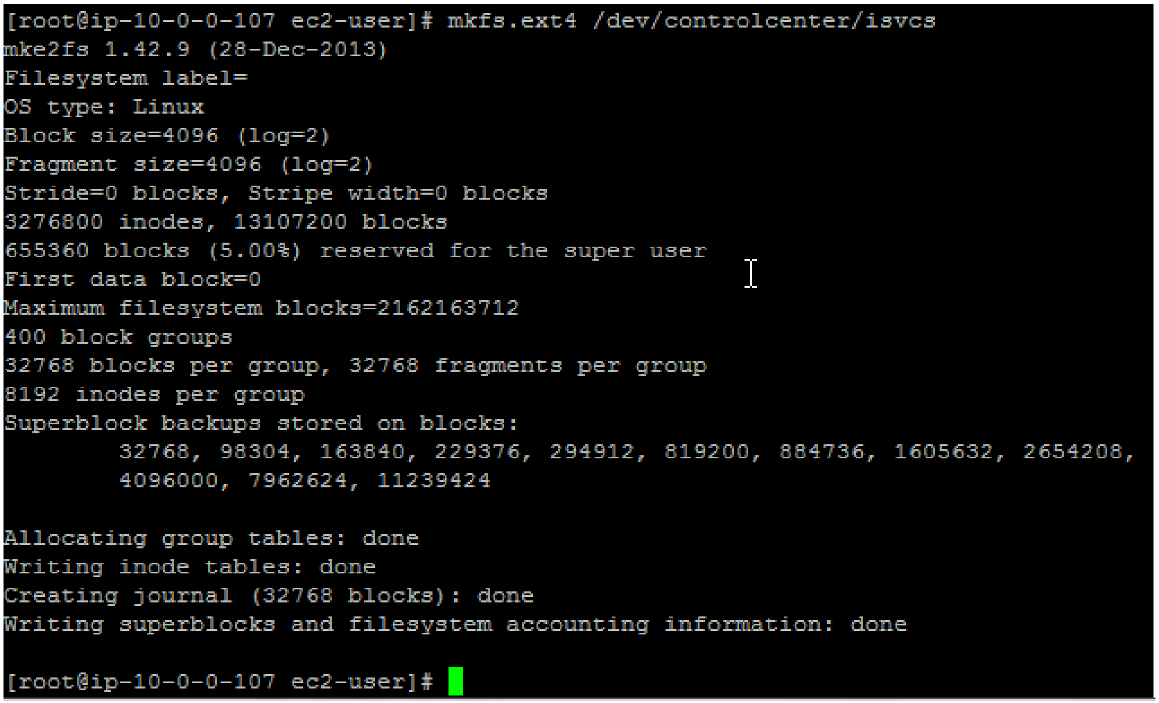
Create the mount point for `isvcs`, for example:

```
mkdir -p /opt/serviced/var/isvcs
```

Create the Filesystem

Create the *ext4* filesystem, for example:

```
mkfs.ext4 /dev/controlcenter/isvcs
```



```
[root@ip-10-0-0-107 ec2-user]# mkfs.ext4 /dev/controlcenter/isvcs
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
3276800 inodes, 13107200 blocks
655360 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2162163712
400 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624, 11239424

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

[root@ip-10-0-0-107 ec2-user]#
```

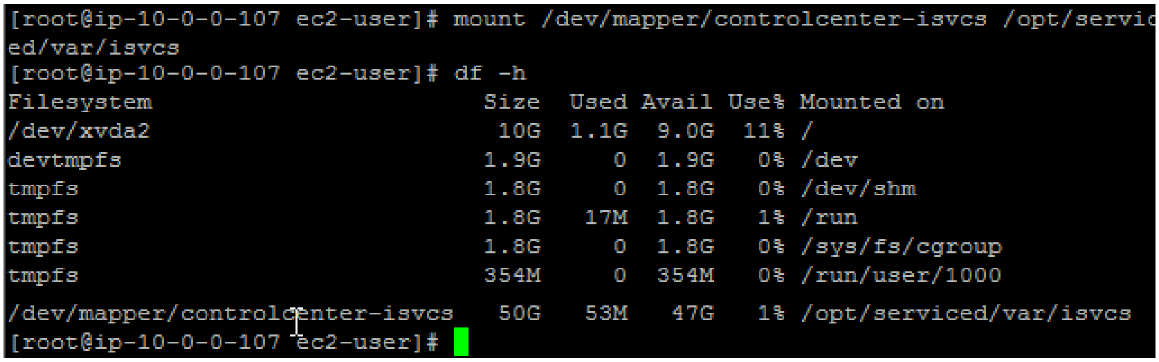
Mount the Filesystem

Mount the filesystem, for example:

```
mount /dev/mapper/controlcenter-isvcs /opt/serviced/var/isvcs
```

View the logical volume information:

```
df -h
```



```
[root@ip-10-0-0-107 ec2-user]# mount /dev/mapper/controlcenter-isvcs /opt/servic
ed/var/isvcs
[root@ip-10-0-0-107 ec2-user]# df -h
Filesystem                          Size  Used Avail Use% Mounted on
/dev/xvda2                           10G  1.1G  9.0G  11% /
devtmpfs                            1.9G     0  1.9G   0% /dev
tmpfs                               1.8G     0  1.8G   0% /dev/shm
tmpfs                               1.8G   17M  1.8G   1% /run
tmpfs                               1.8G     0  1.8G   0% /sys/fs/cgroup
tmpfs                               354M     0  354M   0% /run/user/1000
/dev/mapper/controlcenter-isvcs      50G   53M   47G   1% /opt/serviced/var/isvcs
[root@ip-10-0-0-107 ec2-user]#
```

Update /etc/fstab

Add the new partition/filesystem information to the `/etc/fstab` file.
Edit the `/etc/fstab` file to append the new line. For example:

1. Open `/etc/fstab` in a text editor.

2. Enter the new information For example, add the following line:

```
/dev/mapper/controlcenter-isvcs /opt/serviced/var/isvcs ext4 defaults 0 0
```

3. Save and exit the file.

Create the Logical Volume for Backup

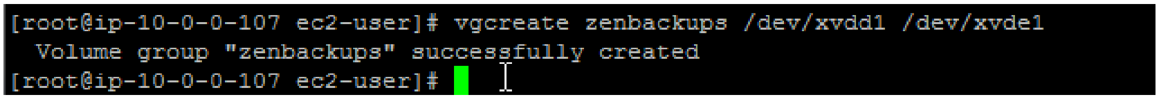
The steps for creating the logical volume include:

- Creating the volume group that defines which physical volumes are used
- Creating the logical volume within the volume group
- Creating the mount point
- Creating the filesystem for the logical volume
- Mounting the filesystem
- Updating the `fstab` file

Create the backups Volume Group

1. Create the *volume group*. For example, create the *zenbackups* volume group and add the remaining two disk partitions (*vxdd1* & *vxde1*) to the volume group:

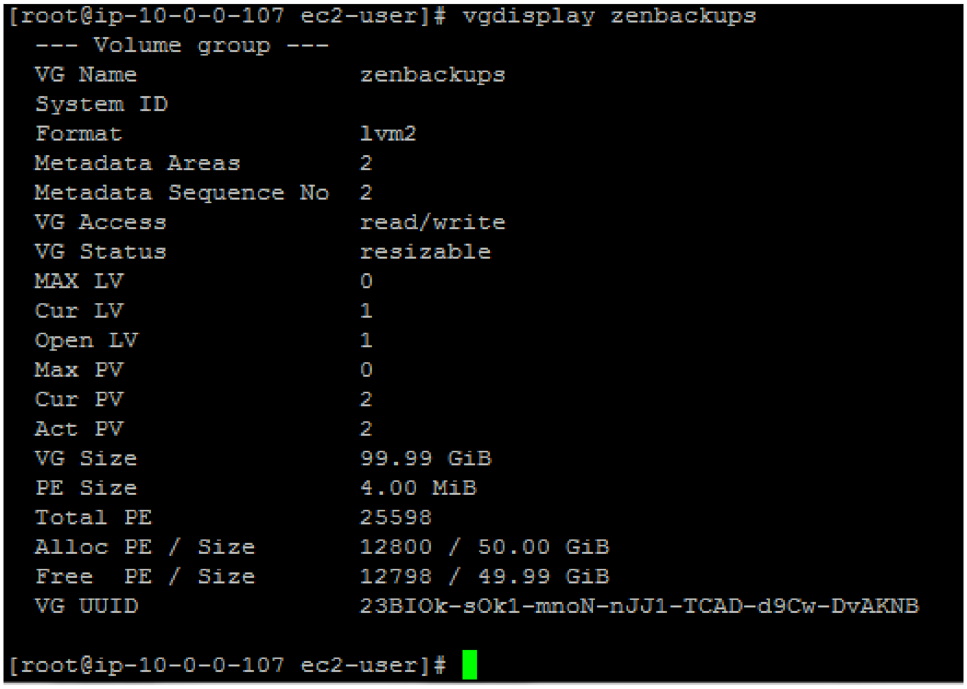
```
vgcreate zenbackups /dev/xvdd1 /dev/xvde1
```



```
[root@ip-10-0-0-107 ec2-user]# vgcreate zenbackups /dev/xvdd1 /dev/xvde1
  Volume group "zenbackups" successfully created
[root@ip-10-0-0-107 ec2-user]#
```

2. Display the volume group information:

```
vgdisplay
```

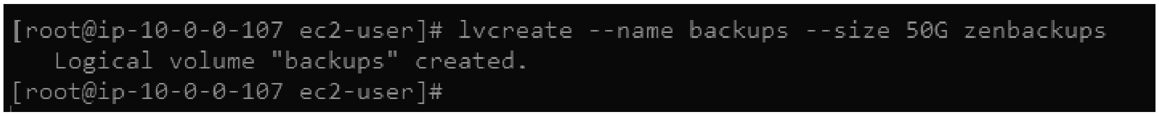


```
[root@ip-10-0-0-107 ec2-user]# vgdisplay zenbackups
  --- Volume group ---
  VG Name               zenbackups
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  2
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               1
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               99.99 GiB
  PE Size               4.00 MiB
  Total PE              25598
  Alloc PE / Size       12800 / 50.00 GiB
  Free  PE / Size       12798 / 49.99 GiB
  VG UUID               23BIOk-sOk1-mnoN-nJJ1-TCAD-d9Cw-DvAKNB

[root@ip-10-0-0-107 ec2-user]#
```

Create the zenbackups Logical Volume

Create the logical volume. For example, to create a logical volume with the name *backups*, a size of *50GB*, that resides in the volume group *zenbackups*:

```
lvcreate --name backups --size 50G zenbackups
```



```
[root@ip-10-0-0-107 ec2-user]# lvcreate --name backups --size 50G zenbackups
  Logical volume "backups" created.
[root@ip-10-0-0-107 ec2-user]#
```

Create the Mount Point

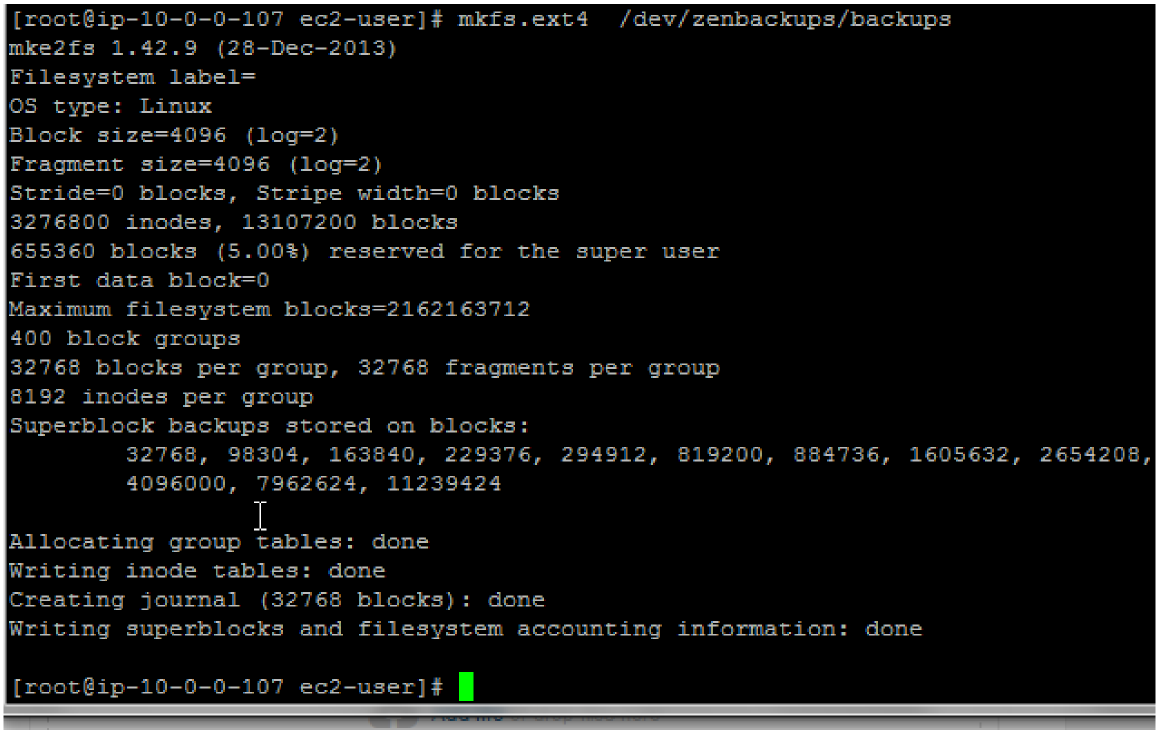
Create the mount point, for example:

```
mkdir -p /opt/serviced/var/backups
```

Create the Filesystem

1. Create the filesystem, for example:

```
mkfs.ext4 /dev/zenbackups/backups
```



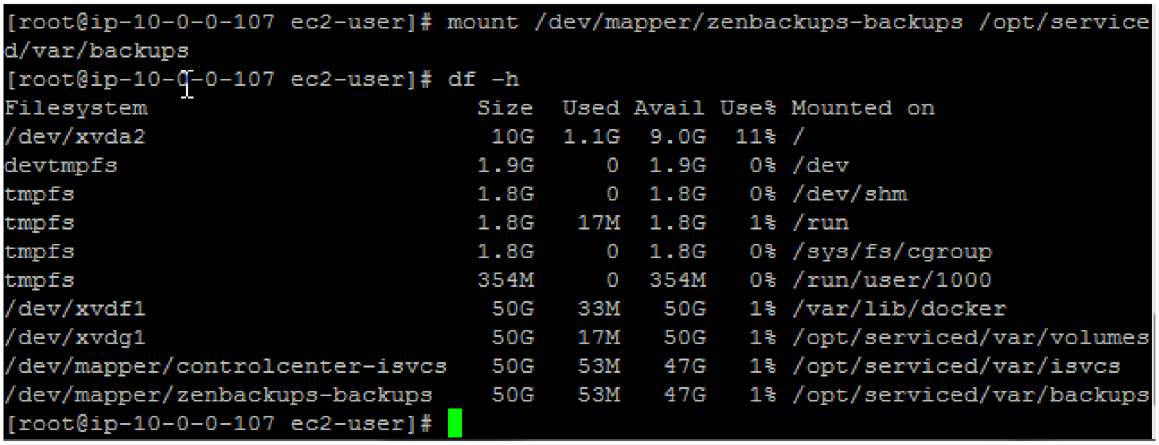
```
[root@ip-10-0-0-107 ec2-user]# mkfs.ext4  /dev/zenbackups/backups
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
3276800 inodes, 13107200 blocks
655360 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2162163712
400 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624, 11239424

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

[root@ip-10-0-0-107 ec2-user]#
```

2. Mount the file system:

```
mount /dev/mapper/zenbackups-backups /opt/serviced/var/backups
```

3. View the logical volume information:

```
df -h
```



```
[root@ip-10-0-0-107 ec2-user]# mount /dev/mapper/zenbackups-backups /opt/service
d/var/backups
[root@ip-10-0-0-107 ec2-user]# df -h
Filesystem                         Size  Used Avail Use% Mounted on
/dev/xvda2                          10G  1.1G  9.0G  11% /
devtmpfs                           1.9G     0  1.9G   0% /dev
tmpfs                              1.8G     0  1.8G   0% /dev/shm
tmpfs                              1.8G   17M  1.8G   1% /run
tmpfs                              1.8G     0  1.8G   0% /sys/fs/cgroup
tmpfs                              354M     0  354M   0% /run/user/1000
/dev/xvdf1                          50G   33M   50G   1% /var/lib/docker
/dev/xvdg1                          50G   17M   50G   1% /opt/serviced/var/volumes
/dev/mapper/controlcenter-isvcs     50G   53M   47G   1% /opt/serviced/var/isvcs
/dev/mapper/zenbackups-backups      50G   53M   47G   1% /opt/serviced/var/backups
[root@ip-10-0-0-107 ec2-user]#
```

Update /etc/fstab

Add the new partition/filesystem information to the `/etc/fstab` file.
Edit the `/etc/fstab` file to append the new line. For example:

a. Open /etc/fstab in a text editor.
b. Enter the new information For example, add the following line:

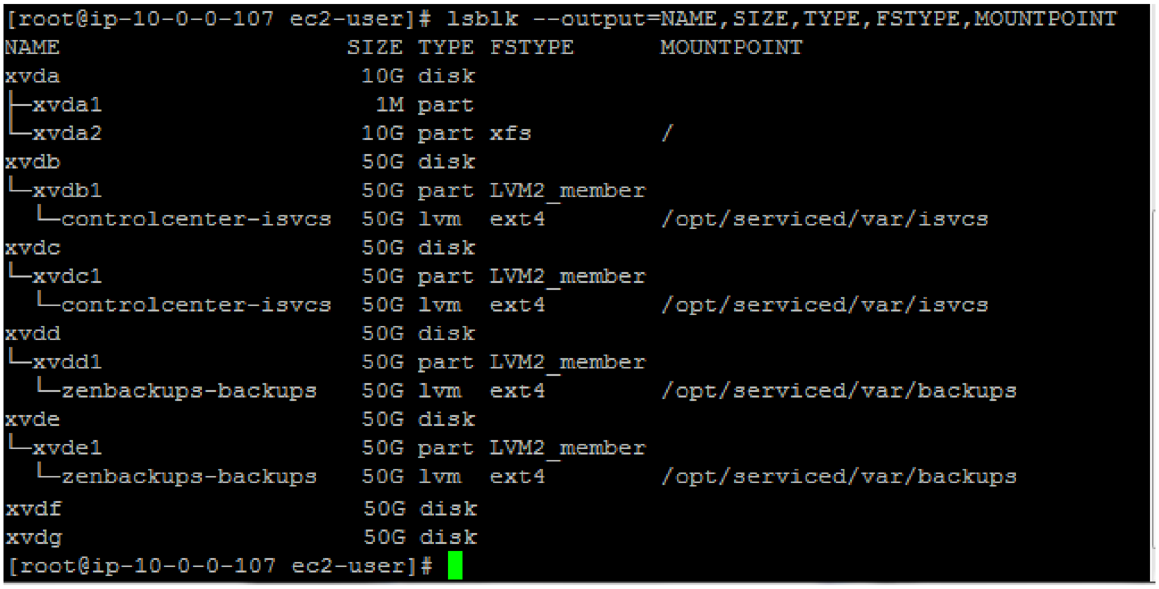
```
/dev/mapper/zenbackups-backups /opt/serviced/var/backups ext4 defaults 0 0
```

c. Save and exit the file

Completion – Logical Volumes

The following output from the `lsblk` command shows the successful creation, partitioning and mounting of the two logical filesystems for *controlcenter-isvcs* and *zenbackups-backups*, configured with two disk partitions for each logical filesystem:

- Disk partitions *xvdb1* & *xvdc1* for *controlcenter-isvcs,* mounted on: `/opt/serviced/var/isvcs`
- Disk partitions *xvdd1* & *xvde1* for *zenbackups-backups*, mounted on: `/opt/serviced/var/backups`



```
[root@ip-10-0-0-107 ec2-user]# lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
NAME                         SIZE TYPE FSTYPE      MOUNTPOINT
xvda                          10G disk
├─xvda1                        1M part
└─xvda2                       10G part xfs         /
xvdb                          50G disk
└─xvdb1                       50G part LVM2_member
  └─controlcenter-isvcs       50G lvm  ext4        /opt/serviced/var/isvcs
xvdc                          50G disk
└─xvdc1                       50G part LVM2_member
  └─controlcenter-isvcs       50G lvm  ext4        /opt/serviced/var/isvcs
xvdd                          50G disk
└─xvdd1                       50G part LVM2_member
  └─zenbackups-backups        50G lvm  ext4        /opt/serviced/var/backups
xvde                          50G disk
└─xvde1                       50G part LVM2_member
  └─zenbackups-backups        50G lvm  ext4        /opt/serviced/var/backups
xvdf                          50G disk
xvdg                          50G disk
[root@ip-10-0-0-107 ec2-user]# 
```

Create the Fixed Filesystem for Docker

The steps for creating the fixed filesystem include:

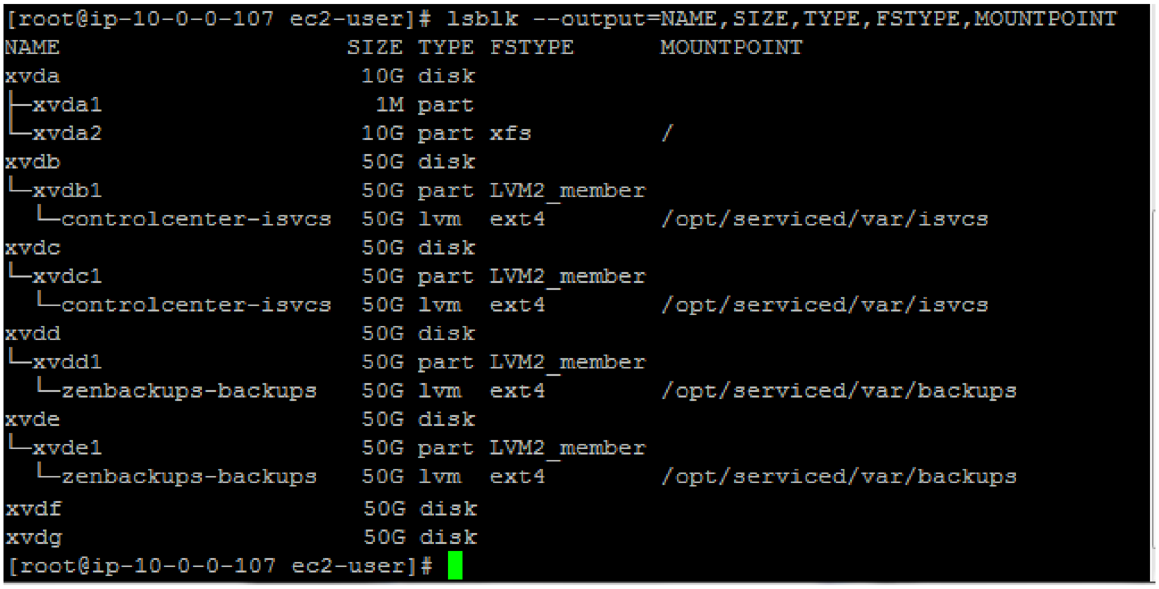
- Creating the partition
- Creating the mount point
- Creating the filesystem
- Updating `fstab`
- Mounting the filesystem

1. Determine the available disks and partitions. Use the `lsblk` command, for example:

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

Note: The options following the equal sign (=) define what output columns the command returns/prints.

The output from the `lsblk` command shows 2 available disks, *xvdf* and *xvdg*:



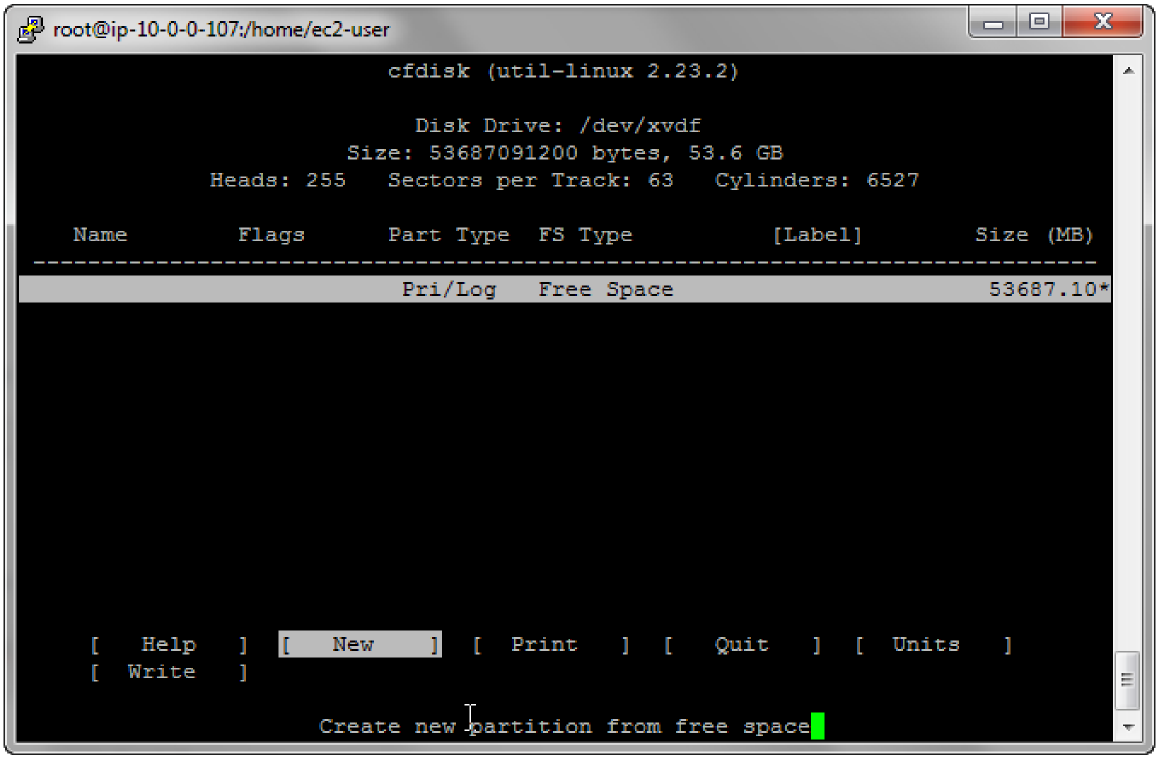
```
[root@ip-10-0-0-107 ec2-user]# lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
NAME                        SIZE TYPE FSTYPE      MOUNTPOINT
xvda                         10G disk
├─xvda1                       1M part
└─xvda2                      10G part xfs         /
xvdb                         50G disk
└─xvdb1                      50G part LVM2_member
  └─controlcenter-isvcs      50G lvm  ext4        /opt/serviced/var/isvcs
xvdc                         50G disk
└─xvdc1                      50G part LVM2_member
  └─controlcenter-isvcs      50G lvm  ext4        /opt/serviced/var/isvcs
xvdd                         50G disk
└─xvdd1                      50G part LVM2_member
  └─zenbackups-backups       50G lvm  ext4        /opt/serviced/var/backups
xvde                         50G disk
└─xvde1                      50G part LVM2_member
  └─zenbackups-backups       50G lvm  ext4        /opt/serviced/var/backups
xvdf                         50G disk
xvdg                         50G disk
[root@ip-10-0-0-107 ec2-user]# 
```

2. Use the `cfdisk` utility to create the partition, define the size and set the type.
   Note: For basic instructions on using the cfdisk interface, see the section called *Using the cfdisk Tool*, below.
   a. Switch to a user with *root* privileges, for example:

   ```
   sudo su
   ```

   b. Launch `cfdisk` and edit the drive, for example:

   ```
   cfdisk /dev/xvdf
   ```



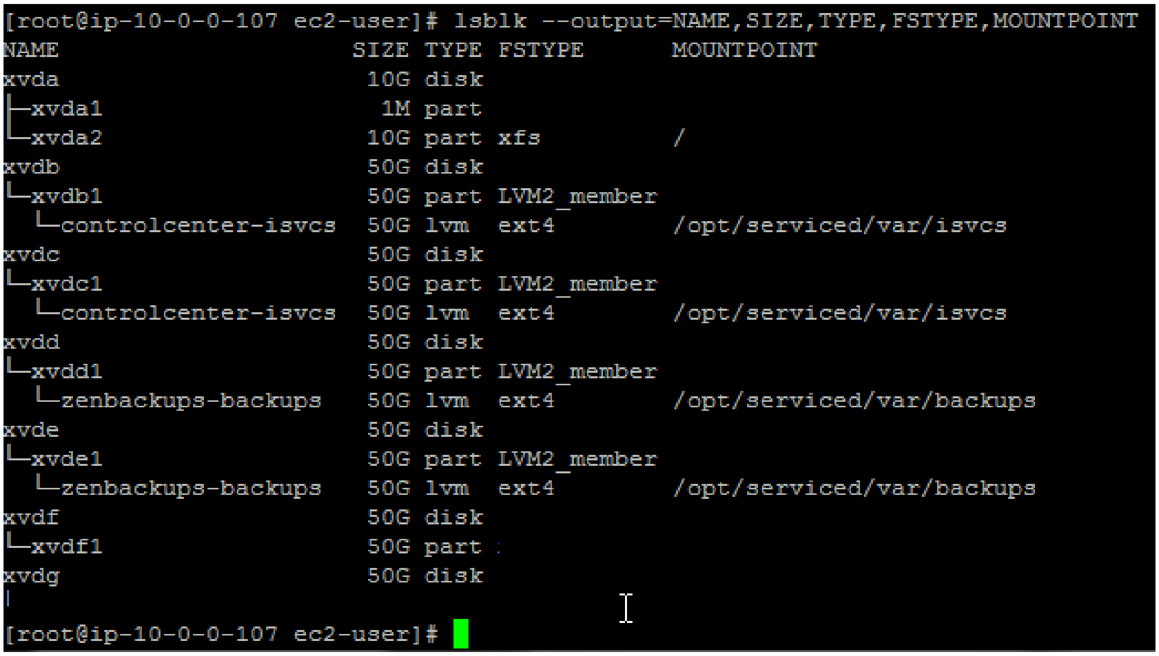
```
                         cfdisk (util-linux 2.23.2)

                            Disk Drive: /dev/xvdf
                      Size: 53687091200 bytes, 53.6 GB
            Heads: 255   Sectors per Track: 63   Cylinders: 6527

    Name        Flags      Part Type  FS Type          [Label]        Size (MB)
 ------------------------------------------------------------------------------
                            Pri/Log   Free Space                      53687.10*











     [  Help  ]  [  New  ]  [  Print  ]  [  Quit  ]  [  Units  ]
     [  Write  ]

                    Create new partition from free space
```

c. Create the *partition.*
   Highlight **New**.
   Press **Enter**.
d. Set the type (*Primary* (default) or *Logical*):
   Highlight **Primary.**
   Press **Enter.**
e. Set the *size*, for example, to use all available space:
   Press **Enter.**
f. Write the changes to the disk:
   Highlight **Write.**
   Press **Enter** to write to the disk.
g. Answer **yes** to proceed with format.
h. Quit the cfdisk utility:
   Select **Quit.**
   Press **Enter** to exit the tool.

3. Verify the new partition.

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

The new partition *vxdf1* displays in the output:



```
[root@ip-10-0-0-107 ec2-user]# lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
NAME                        SIZE TYPE FSTYPE      MOUNTPOINT
xvda                         10G disk
├─xvda1                       1M part
└─xvda2                      10G part xfs         /
xvdb                         50G disk
└─xvdb1                      50G part LVM2_member
  └─controlcenter-isvcs      50G lvm  ext4        /opt/serviced/var/isvcs
xvdc                         50G disk
└─xvdc1                      50G part LVM2_member
  └─controlcenter-isvcs      50G lvm  ext4        /opt/serviced/var/isvcs
xvdd                         50G disk
└─xvdd1                      50G part LVM2_member
  └─zenbackups-backups       50G lvm  ext4        /opt/serviced/var/backups
xvde                         50G disk
└─xvde1                      50G part LVM2_member
  └─zenbackups-backups       50G lvm  ext4        /opt/serviced/var/backups
xvdf                         50G disk
└─xvdf1                      50G part
xvdg                         50G disk

[root@ip-10-0-0-107 ec2-user]#
```

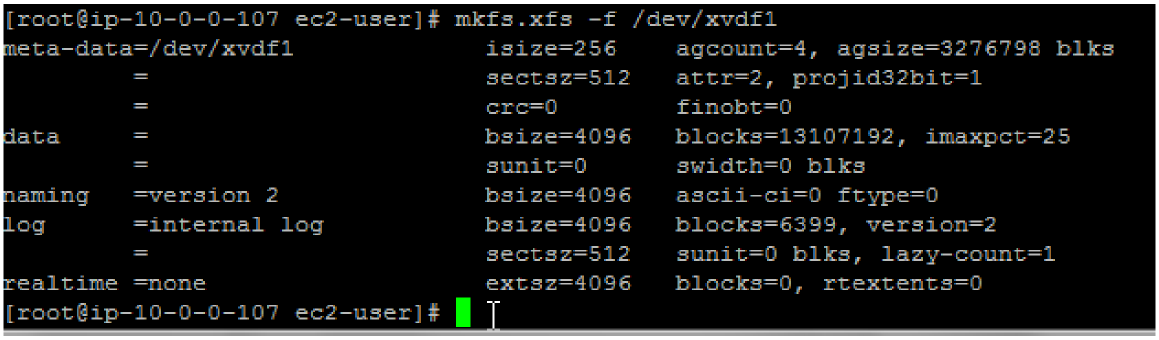
4. Create the mount point, for example:

```
mkdir -p /var/lib/docker
```

5. Create the file system. Use the mkfs command to format the partition. For example, to create an *xfs* file system, issue the following command:

```
mkfs.xfs -f /dev/xvdf1
```

The following example shows the output from the `mkfs` command:



```
[root@ip-10-0-0-107 ec2-user]# mkfs.xfs -f /dev/xvdf1
meta-data=/dev/xvdf1             isize=256    agcount=4, agsize=3276798 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=0        finobt=0
data     =                       bsize=4096   blocks=13107192, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=0
log      =internal log           bsize=4096   blocks=6399, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[root@ip-10-0-0-107 ec2-user]#
```

6. Add the new partition/filesystem information to the /etc/fstab file. Edit the /etc/fstab file to append the new line. For example:

   a. Open `/etc/fstab` in a text editor
   b. Enter the new information. For example, add the following line:

   ```
    /dev/xvdf1 /var/lib/docker xfs defaults 0 0
   ```

   c. Save and exit the file.

Note: The fields in the `fstab` file are:

```
DeviceName  MountPoint  FileSystemType MountOptions  DumbOptions  FileSystemCheckOptions(fsck)
```

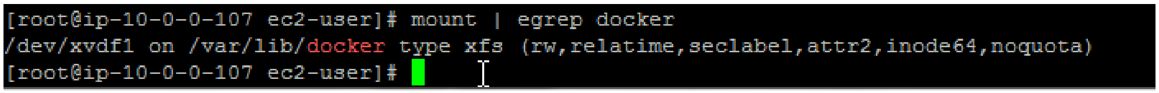
7. Mount the partition. For example to mount all unmounted partitions:

```
mount -a
```

8. Verify the partition mounted correctly. For example, use following command and consult the output:

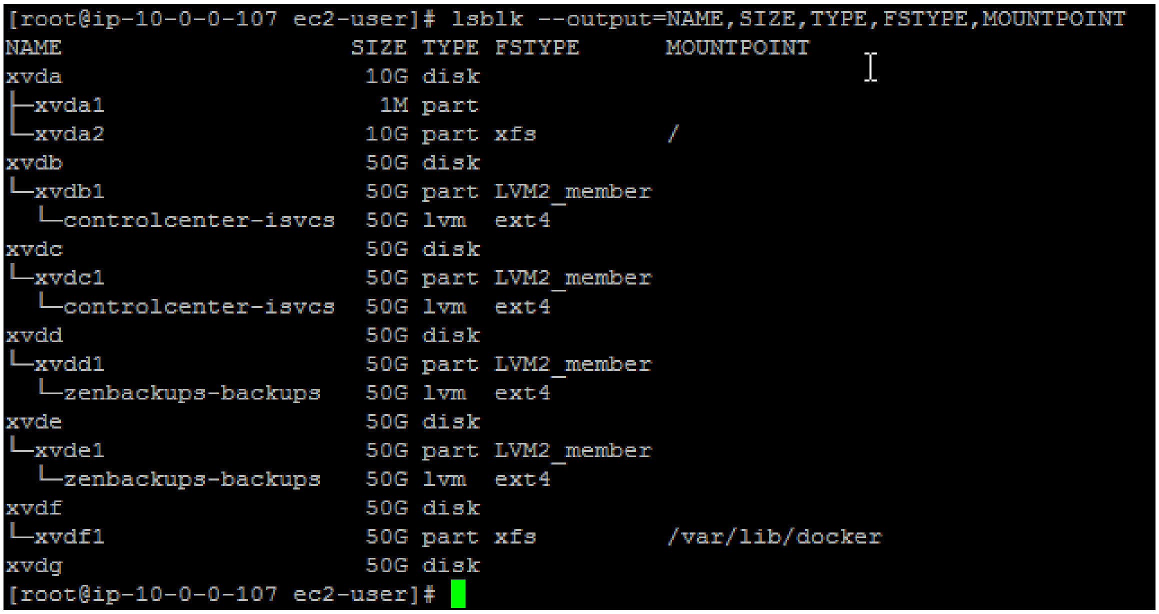
```
mount | egrep docker
```

The following shows a positive result in the command output:



```
[root@ip-10-0-0-107 ec2-user]# mount | egrep docker
/dev/xvdf1 on /var/lib/docker type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
[root@ip-10-0-0-107 ec2-user]#
```

The `lsblk` command shows the new mounted *xvdf1* partition:



```
[root@ip-10-0-0-107 ec2-user]# lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
NAME                        SIZE TYPE FSTYPE      MOUNTPOINT
xvda                         10G disk
├─xvda1                       1M part
└─xvda2                      10G part xfs         /
xvdb                         50G disk
└─xvdb1                      50G part LVM2_member
  └─controlcenter-isvcs      50G lvm  ext4
xvdc                         50G disk
└─xvdc1                      50G part LVM2_member
  └─controlcenter-isvcs      50G lvm  ext4
xvdd                         50G disk
└─xvdd1                      50G part LVM2_member
  └─zenbackups-backups       50G lvm  ext4
xvde                         50G disk
└─xvde1                      50G part LVM2_member
  └─zenbackups-backups       50G lvm  ext4
xvdf                         50G disk
└─xvdf1                      50G part xfs         /var/lib/docker
xvdg                         50G disk
[root@ip-10-0-0-107 ec2-user]#
```

Create the Fixed Filesystem for DFS

The steps for creating the fixed filesystem include:

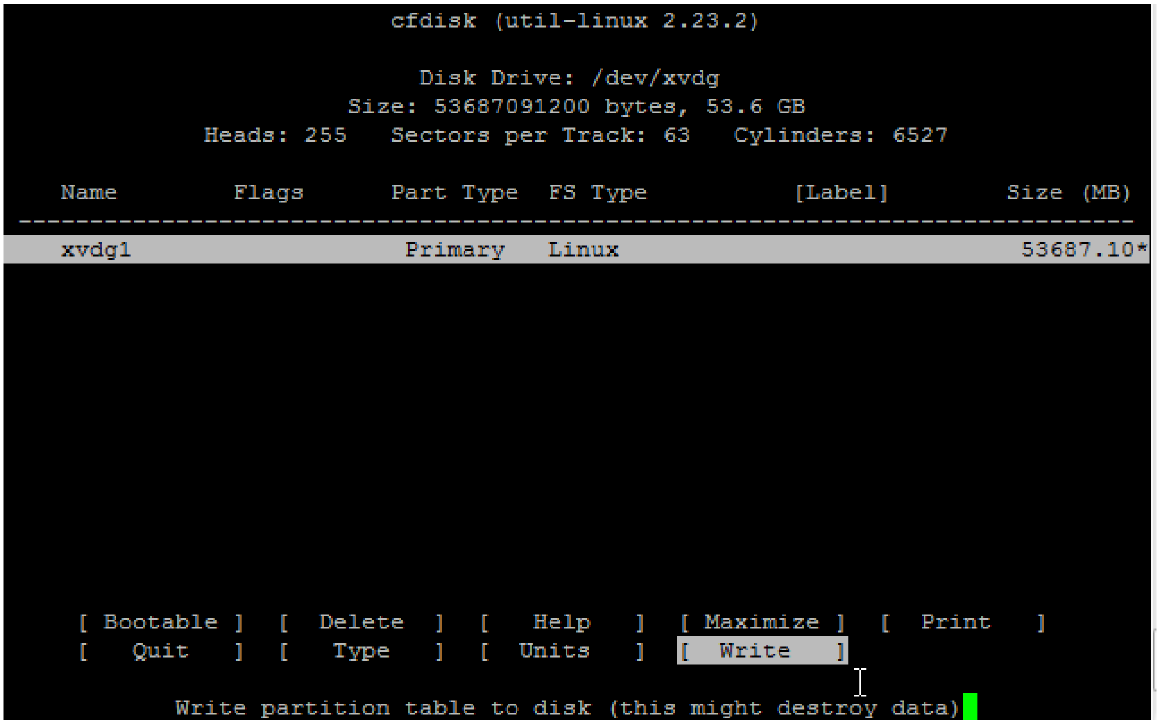
- Creating the partition
- Creating the mount point
- Creating the filesystem
- Updating `fstab`
- Mounting the filesystem

1. Use the `cfdisk` utility to create the partition, define the size and set the type
   a. Switch to a user with *root* privileges, for example:

```
sudo su
```

b. Launch `cfdisk` and edit the drive, for example:

```
cfdisk /dev/xvdg
```



```
                          cfdisk (util-linux 2.23.2)

                            Disk Drive: /dev/xvdg
                       Size: 53687091200 bytes, 53.6 GB
             Heads: 255   Sectors per Track: 63   Cylinders: 6527

    Name        Flags      Part Type  FS Type          [Label]        Size (MB)
 ------------------------------------------------------------------------------
    xvdg1                   Primary   Linux                            53687.10*






     [ Bootable ]  [  Delete  ]  [   Help   ]  [ Maximize ]  [  Print   ]
     [   Quit   ]  [   Type   ]  [  Units   ]  [  Write   ]

            Write partition table to disk (this might destroy data)
```

c. Create the *partition.*
Highlight **New**.
Press **Enter***.*

d. Set the type (*Primary* (default) or *Logical*):
Highlight **Primary.**
Press **Enter**

e. Set the *size*, for example, to use all available space:
Press **Enter**.

f. Write the changes to the disk:
Highlight **Write.**
Press **Enter** to write to the disk.

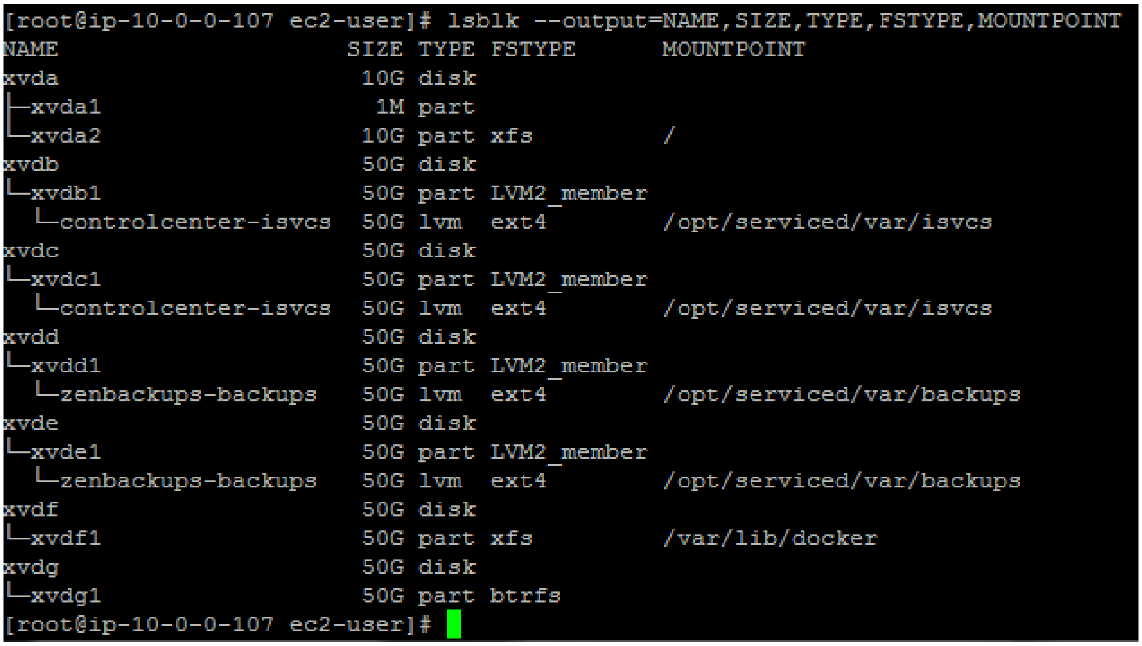
g. Answer **yes** to proceed with format.

h. Quit the cfdisk utility:
Select **Quit.**
Press **Enter** to exit the tool.

2. Verify the new partition:

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

The new partition *vxdg1* displays in the output:



```
[root@ip-10-0-0-107 ec2-user]# lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
NAME                       SIZE TYPE FSTYPE      MOUNTPOINT
xvda                        10G disk
├─xvda1                      1M part
└─xvda2                     10G part xfs         /
xvdb                        50G disk
└─xvdb1                     50G part LVM2_member
  └─controlcenter-isvcs     50G lvm  ext4        /opt/serviced/var/isvcs
xvdc                        50G disk
└─xvdc1                     50G part LVM2_member
  └─controlcenter-isvcs     50G lvm  ext4        /opt/serviced/var/isvcs
xvdd                        50G disk
└─xvdd1                     50G part LVM2_member
  └─zenbackups-backups      50G lvm  ext4        /opt/serviced/var/backups
xvde                        50G disk
└─xvde1                     50G part LVM2_member
  └─zenbackups-backups      50G lvm  ext4        /opt/serviced/var/backups
xvdf                        50G disk
└─xvdf1                     50G part xfs         /var/lib/docker
xvdg                        50G disk
└─xvdg1                     50G part btrfs
[root@ip-10-0-0-107 ec2-user]#
```

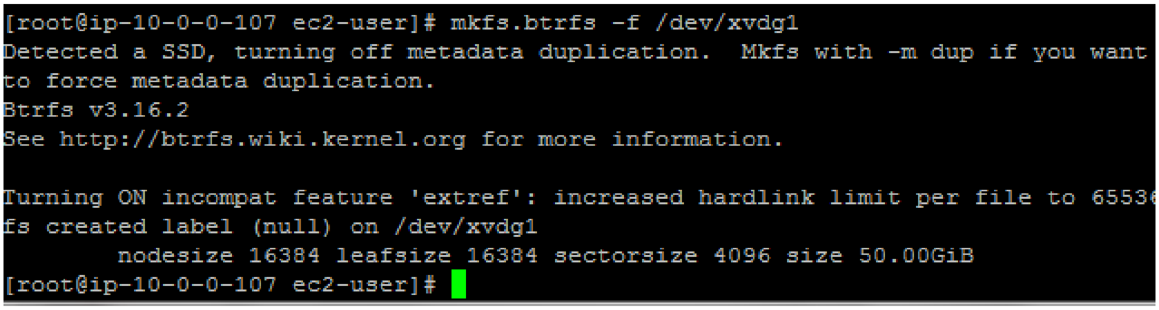
3. Create the mount point, for example:

```
mkdir -p /opt/serviced/var/volumes
```

4. Create the file system. Use the `mkfs` command to format the partition. For example, to create a *btrfs* file system:

```
mkfs.btrfs -f /dev/xvdg1
```

The following example shows the `mkfs` command output:



```
[root@ip-10-0-0-107 ec2-user]# mkfs.btrfs -f /dev/xvdg1
Detected a SSD, turning off metadata duplication.  Mkfs with -m dup if you want
to force metadata duplication.
Btrfs v3.16.2
See http://btrfs.wiki.kernel.org for more information.

Turning ON incompat feature 'extref': increased hardlink limit per file to 65536
fs created label (null) on /dev/xvdg1
        nodesize 16384 leafsize 16384 sectorsize 4096 size 50.00GiB
[root@ip-10-0-0-107 ec2-user]#
```

5. Add the new partition/filesystem information to the `/etc/fstab` file.
   Edit the `/etc/fstab` file to append the new line. For example:

   a. Open `/etc/fstab` in a text editor.
   b. Enter the new information. For example, add the following line:

```
/dev/xvdg1 /opt/serviced/var/volumes btrfs rw,noatime,nodatacow,skip_balance 0 0
```

   c. Save and exit the file.

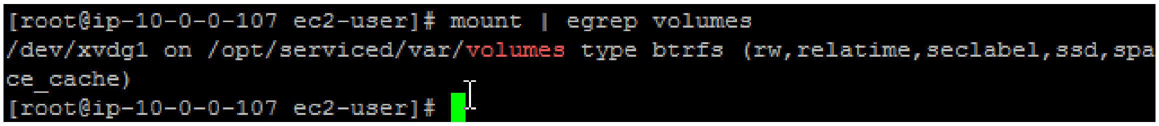
6. Mount the partition. For example to mount all unmounted partitions:

```
mount -a
```

7. Verify the partition mounted correctly. For example:

```
mount | egrep volumes
```

The following shows a positive result in the command output:

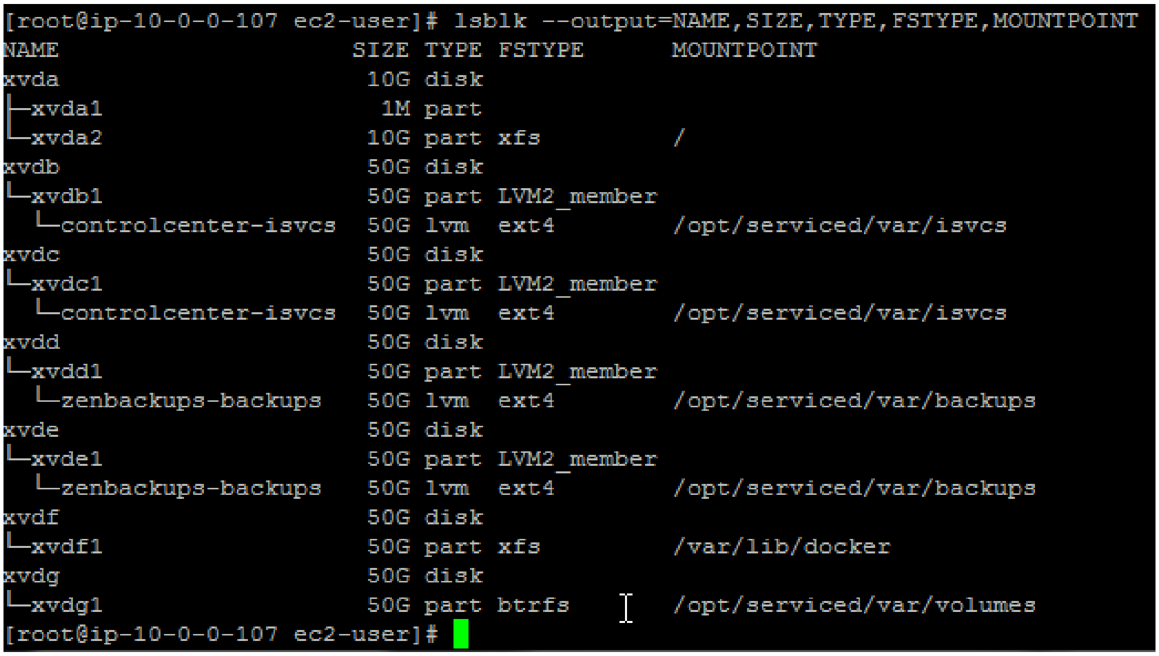


```
[root@ip-10-0-0-107 ec2-user]# mount | egrep volumes
/dev/xvdg1 on /opt/serviced/var/volumes type btrfs (rw,relatime,seclabel,ssd,spa
ce_cache)
[root@ip-10-0-0-107 ec2-user]#
```

Note: If you need to determine the free space available on a btrfs filesystem, use the `btrfs` command, for example:

```
btrfs filesystem df /opt/serviced/var/volumes
```

The `lsblk` command shows the new mounted xvdg1 partition:

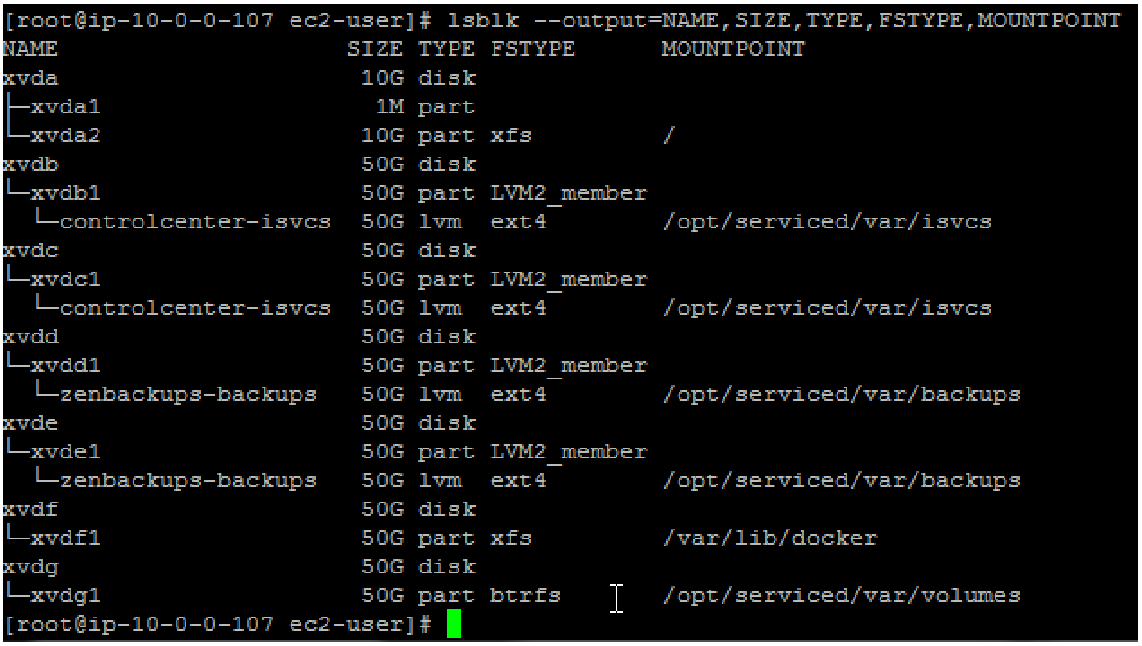


```
[root@ip-10-0-0-107 ec2-user]# lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
NAME                       SIZE TYPE FSTYPE      MOUNTPOINT
xvda                        10G disk
├─xvda1                      1M part
└─xvda2                     10G part xfs         /
xvdb                        50G disk
└─xvdb1                     50G part LVM2_member
  └─controlcenter-isvcs     50G lvm  ext4        /opt/serviced/var/isvcs
xvdc                        50G disk
└─xvdc1                     50G part LVM2_member
  └─controlcenter-isvcs     50G lvm  ext4        /opt/serviced/var/isvcs
xvdd                        50G disk
└─xvdd1                     50G part LVM2_member
  └─zenbackups-backups      50G lvm  ext4        /opt/serviced/var/backups
xvde                        50G disk
└─xvde1                     50G part LVM2_member
  └─zenbackups-backups      50G lvm  ext4        /opt/serviced/var/backups
xvdf                        50G disk
└─xvdf1                     50G part xfs         /var/lib/docker
xvdg                        50G disk
└─xvdg1                     50G part btrfs       /opt/serviced/var/volumes
[root@ip-10-0-0-107 ec2-user]#
```

Completion - Logical & Fixed Volume Creation

The following output from the `lsblk` command shows the successful partitioning, creation, and mounting of the four required filesystems:

- Logical volume for *Control Center*, using two disk partitions (*xvdb1* & *xvdc1*), mounted on: `/opt/serviced/var/isvcs`
- Logical volume for *Backups*, using two disk partitions (*xvdd1* & *xvde1*), mounted on: `/opt/serviced/var/backups`
- Fixed volume for *Docker* on single disk partition (*xvdf1*), mounted on: `/var/lib/docker`
- Fixed volume for *DFS* on single disk partition (*xvdg1*), mounted on: `/opt/serviced/var/volumes`



```
[root@ip-10-0-0-107 ec2-user]# lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
NAME                     SIZE TYPE FSTYPE      MOUNTPOINT
xvda                      10G disk
├─xvda1                    1M part
└─xvda2                   10G part xfs         /
xvdb                      50G disk
└─xvdb1                   50G part LVM2_member
  └─controlcenter-isvcs   50G lvm  ext4        /opt/serviced/var/isvcs
xvdc                      50G disk
└─xvdc1                   50G part LVM2_member
  └─controlcenter-isvcs   50G lvm  ext4        /opt/serviced/var/isvcs
xvdd                      50G disk
└─xvdd1                   50G part LVM2_member
  └─zenbackups-backups    50G lvm  ext4        /opt/serviced/var/backups
xvde                      50G disk
└─xvde1                   50G part LVM2_member
  └─zenbackups-backups    50G lvm  ext4        /opt/serviced/var/backups
xvdf                      50G disk
└─xvdf1                   50G part xfs         /var/lib/docker
xvdg                      50G disk
└─xvdg1                   50G part btrfs       /opt/serviced/var/volumes
[root@ip-10-0-0-107 ec2-user]#
```

Appendix: Installing the LVM Tools

If the LVM tools are not installed, perform the following:

1. Download and install the LVM tools:

```
yum install -y lvm2
```

2. Issue the following commands to start the `lvm2` services:

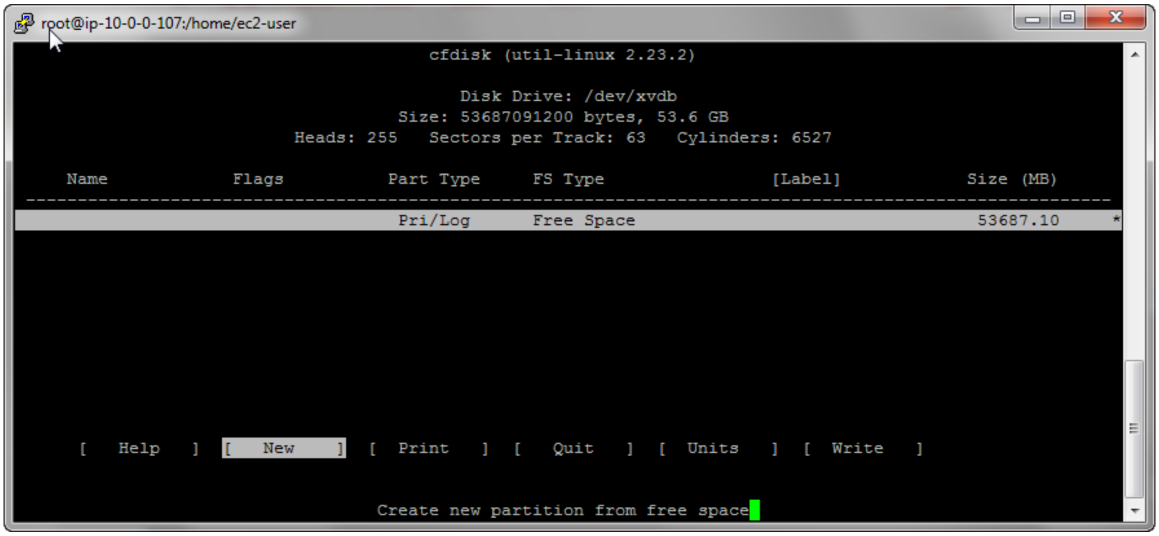
```
systemctl enable lvm2-lvmetad.service
systemctl enable lvm2-lvmetad.socket
systemctl start lvm2-lvmetad.service
systemctl start lvm2-lvmetad.socket
```

Appendix: Using the cfdisk Tool

The `cfdisk` command provides a text user interface (TUI) for editing a disk partition table.

The following figure shows an example initial `cfdisk` screen that displays with the command:

```
cfdisk /dev/xvdb
```



```
                          cfdisk (util-linux 2.23.2)

                             Disk Drive: /dev/xvdb
                       Size: 53687091200 bytes, 53.6 GB
              Heads: 255   Sectors per Track: 63   Cylinders: 6527

    Name         Flags        Part Type    FS Type          [Label]        Size (MB)
 ------------------------------------------------------------------------------------
                              Pri/Log      Free Space                       53687.10 *

     [  Help  ]  [  New  ]  [  Print  ]  [  Quit  ]  [  Units  ]  [  Write  ]

                      Create new partition from free space
```

Navigating the cfdisk Interface

- To select an entry in the table, use the **up** and **down** *arrow* keys to highlight the current entry.
- To select a command from the menu at the bottom of the interface, use the **left** and **right** *arrow* keys, or **Tab** and **Shift-Tab**. The current command is highlighted.
- To execute a selected command, press the **Enter** key.
- To return to the previous level of the menu, press the **Esc** key.
- To exit the interface, select **Quit** from the menu, and press the **Enter key**.

For additional information about the `cfdisk` command, read the man page:

```
man cfdisk
```