Elastic Volume Service

Best Practices

Issue	01
Date	2020-03-12

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process*. For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

Contents

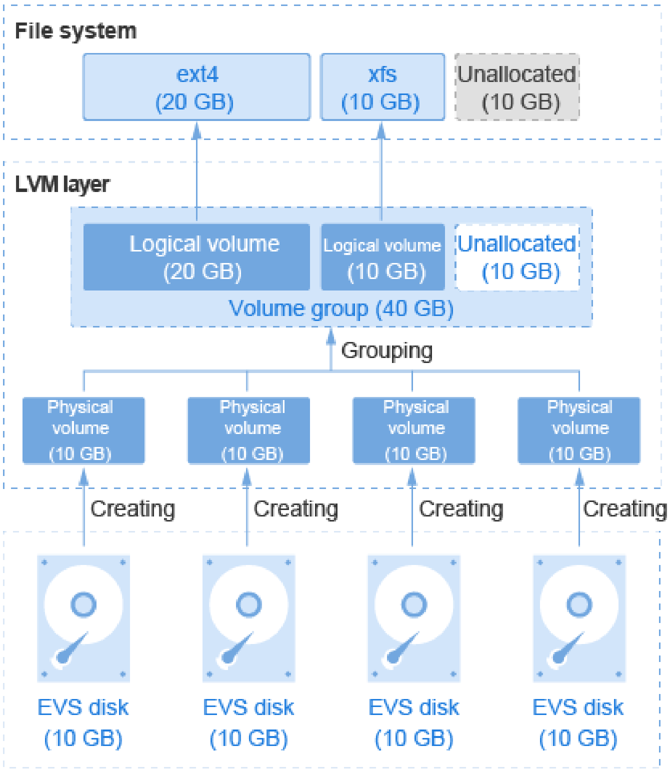
1 Using LVM to Manage EVS Disks

1.1 Overview of EVS Disk Management Using LVM

LVM is short for Logical Volume Manager, which is a mechanism used for managing disk partitions in Linux.

By adding a logical layer between EVS disks and file systems, LVM abstracts EVS disk partitions into logical volumes that can then be flexibly partitioned as needed for upper layer file systems. **Figure 1-1** shows the LVM architecture.

Figure 1-1 LVM architecture



The process of managing EVS disks using LVM is as follows:

1. **Create physical volumes using EVS disks.**
2. **Create a volume group for the physical volumes.**
3. **Create logical volumes in the volume group.**
4. **Create file systems on logical volumes.**

With LVM, a file system can be created on top of multiple EVS disks and can be easily resized as needed. This way, the file system size is no longer limited by the underlying disk capacity.

For example, you can expand the size of the ext4 file system in **Figure 1-1** in either of the following ways:

- Extend the logical volume directly if the unallocated space in the volume group is sufficient.
- Extend the volume group and then logical volumes if the unallocated space in a volume group is insufficient.

Glossary

- Physical Volume

  Physical volumes are basic storage devices in LVM and are created based on EVS disks and LVM management parameters.
- Volume Group

  A volume group concatenates physical volumes into a large storage pool that can be consecutively addressed.
- Logical Volume

  Logical volumes are obtained by partitioning the volume group according to the logic.

1.2 Resource Planning

Table 1-1 Resource planning and costs

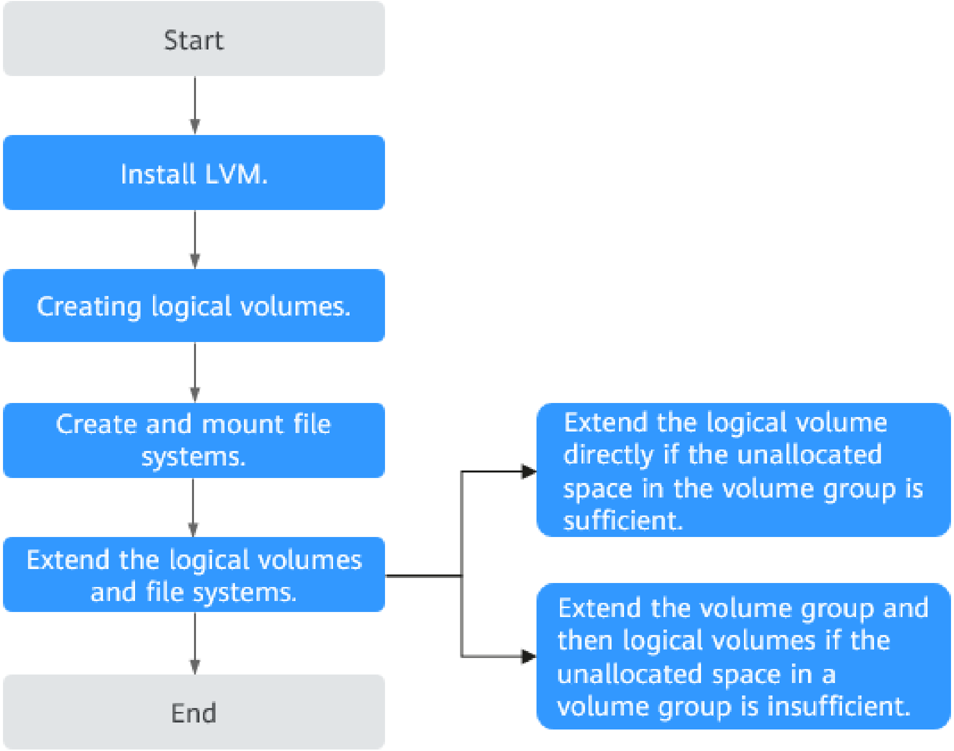
Resource	Description	Quantity	Monthly Fee
Elastic IP (EIP)	The ECS needs to have an EIP bound.	1	For details about the billing modes and billing standards, see **Billing**.
Elastic Cloud Server (ECS)	Operating system: CentOS	1	For details about the billing modes and billing standards, see **Billing**.
EVS disk	Data disk: 10 GB	4	For details about the billing modes and billing standards, see **Billing for Disks**.

1.3 Operation Process

The process of managing EVS disks using LVM is as follows:

1. **Installing LVM**
2. **Creating a Logical Volume Using LVM**
3. **Creating and Mounting a File System**
4. Expanding the Logical Volume and File System
   a. **Extending the Logical Volume Using the Unallocated Space**
   b. **Extending the Logical Volume by Expanding Capacity of an EVS Disk**
   c. **Extending the Volume Group by Adding an EVS Disk**

Figure 1-2 Using LVM to manage EVS disks


Start
Install LVM.
Creating logical volumes.
Create and mount file systems.
Extend the logical volumes and file systems.
End
Extend the logical volume directly if the unallocated space in the volume group is sufficient.
Extend the volume group and then logical volumes if the unallocated space in a volume group is insufficient.

1.4 Implementation Procedure

1.4.1 Installing LVM

Scenarios

By default, LVM is not installed in the ECS operating system (OS), and you need to manually install it. This section shows how to check whether LVM is installed on your ECS and how to install LVM.

In this section, CentOS 7.5 64bit is used as the sample OS. The method for formatting an EVS disk varies depending on the OS running on the server. This section is used for reference only.

Prerequisites

You have an ECS and have bound an EIP to the ECS.

Procedure

Step 1 Log in to the ECS as user **root**.

Step 2 Run the following command to check whether LVM is installed:

rpm -qa |grep lvm2

```
[root@ecs-lvmtest ~]# rpm -qa |grep lvm2
lvm2-libs-2.02.177-4.el7.x86_64
lvm2-2.02.177-4.el7.x86_64
```

- If the preceding information is displayed, LVM has been installed. Go to **Creating a Logical Volume Using LVM**.
- If the preceding information is not displayed, LVM is not installed. Go to **Step 3**.

Step 3 Run the following command and follow the prompts to install LVM:

yum install lvm2

Information similar to the following is displayed:

```
......
Installed:
  lvm2.x86_64 7:2.02.177-4.el7

Dependency Installed:
  device-mapper-event.x86_64 7:1.02.146-4.el7                   device-mapper-event-libs.x86_64 
7:1.02.146-4.el7
  device-mapper-persistent-data.x86_64 0:0.7.3-3.el7             lvm2-libs.x86_64 7:2.02.177-4.el7

Dependency Updated:
  device-mapper.x86_64 7:1.02.146-4.el7                       device-mapper-libs.x86_64 7:1.02.146-4.el7

Complete!
```

When **Complete!** is displayed, LVM is successfully installed.

----End

1.4.2 Creating a Logical Volume Using LVM

Scenarios

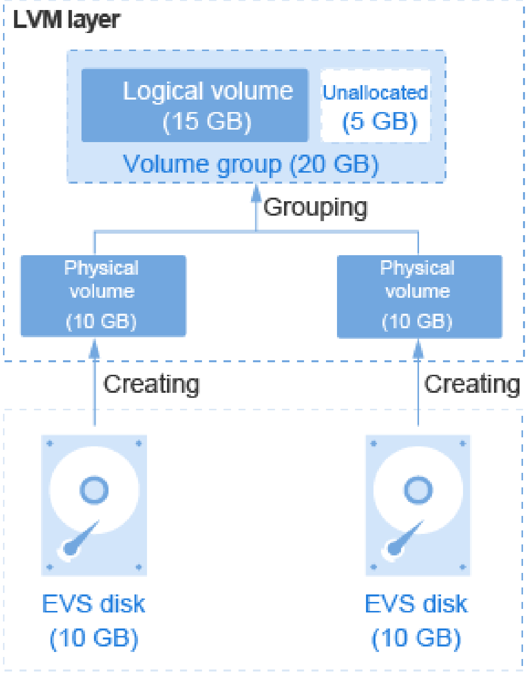
This section shows how to create a 15 GB logical volume based on two 10 GB EVS disks.

NOTE

Logical volumes can be created based on EVS disks with different specifications.

The process includes creating physical volumes, create a volume group, and create a logical volume.

Figure 1-3 Process of creating an LVM logical volume



Prerequisites

Two EVS disks have been attached to the ECS where LVM is installed.

Procedure

Step 1 Log in to the ECS as user **root**.

Step 2 Run the following command to view and take note of the device names:

fdisk -l | grep /dev/vd | grep -v vda

Information similar to the following is displayed:

```
[root@ecs-lvmtest ~]# fdisk -l | grep /dev/vd | grep -v vda
Disk /dev/vdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Disk /dev/vdc: 10.7 GB, 10737418240 bytes, 20971520 sectors
```

In the command output, two EVS disks are attached to the ECS, and the device names are **/dev/vdb** and **/dev/vdc**.

Step 3 Create physical volumes using EVS disks.

1. Run the following command to create physical volumes using EVS disks:

   pvcreate *Device name 1 Device name 2 Device name 3*...

   Parameter description:

   Device name: indicates the disk device name. If multiple physical volumes need to be created in a batch, specify multiple device names and separate them with spaces.

   In this example, run the following command:

   pvcreate /dev/vdb /dev/vdc

   Information similar to the following is displayed:

```
[root@ecs-lvmtest ~]#  pvcreate /dev/vdb /dev/vdc
  Physical volume "/dev/vdb" successfully created.
  Physical volume "/dev/vdc" successfully created.
```

2. Run the following command to query details of the physical volumes:

   pvdisplay

   Information similar to the following is displayed:

```
[root@ecs-lvmtest ~]# pvdisplay
  "/dev/vdc" is a new physical volume of "10.00 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/vdc
  VG Name
  PV Size               10.00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               dypyLh-xjIj-PvG3-jD0j-yup5-O7SI-462R7C

  "/dev/vdb" is a new physical volume of "10.00 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/vdb
  VG Name
  PV Size               10.00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               srv5H1-tgLu-GRTl-Vns8-GfNK-jtHk-Ag4HHB
```

   In the command output, the system has two new physical volumes named **/dev/vdc** and **/dev/vdb**.

Step 4 Create a volume group for the physical volumes.

1. Run the following command to create a volume group:

   vgcreate *Volume group name Physical volume name 1 Physical volume name 2 Physical volume name 3...*

   Parameter description:

   – *Volume group name*: Specify a volume group name, for example, **vgdata**.

   – *Physical volume name*: Specify the name of a physical volume to be added to the volume group. Multiple names are separated with spaces.

   In this example, run the following command:

   vgcreate vgdata /dev/vdb /dev/vdc

   Information similar to the following is displayed:

```
[root@ecs-lvmtest ~]# vgcreate vgdata /dev/vdb /dev/vdc
  Volume group "vgdata" successfully created
```

2. Run the following command to query details of the volume group:

   vgdisplay

   Information similar to the following is displayed:

```
[root@ecs-lvmtest ~]# vgdisplay
  --- Volume group ---
  VG Name               vgdata
  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               19.99 GiB
  PE Size               4.00 MiB
  Total PE              5118
  Alloc PE / Size       0 / 0
  Free  PE / Size       5118 / 19.99 GiB
  VG UUID               NLkZV7-hYYE-0w66-tnlt-Y6jL-Ik7S-76w4P6
```

Step 5 Create a logical volume in the volume group.

1. Run the following command to create a logical volume:

   lvcreate -L *Logical volume size* **-n** *Logical volume name Volume group name*

   Parameter description:

   – *Logical volume size*: Specify a value smaller than the volume group's available space, either in MB or GB.

   – *Logical volume name*: Specify a volume name, for example, **lvdata1**.

   – *Volume group name*: Specify the name of the volume group where the logical volume belongs.

   In this example, run the following command:

   lvcreate -L 15GB -n lvdata1 vgdata

   Information similar to the following is displayed:

```
[root@ecs-lvmtest ~]# lvcreate -L 15GB -n lvdata1 vgdata
  Logical volume "lvdata1" created.
```

2. Run the following command to query details of the logical volume:

   lvdisplay

   Information similar to the following is displayed:

```
[root@ecs-lvmtest ~]# lvdisplay
  --- Logical volume ---
  LV Path                /dev/vgdata/lvdata1
  LV Name                lvdata1
  VG Name                vgdata
  LV UUID                c7mNcF-CdPW-5PLD-1gVj-QZpB-nHfy-PHXchV
  LV Write Access        read/write
  LV Creation host, time ecs-lvmtest.novalocal, 2018-11-29 11:28:18 +0800
  LV Status              available
  # open                 0
  LV Size                15.00 GiB
  Current LE             3840
  Segments               2
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           252:0
```

----End

1.4.3 Creating and Mounting a File System

Scenarios

After the logical volume is created, you need to create a file system on the logical volume and mount the file system on the corresponding directory. This section shows how to create an ext4 file system on a logical volume and mount the file system on **/Data1**.

Procedure

Step 1 Log in to the ECS as user **root**.

Step 2 Run the following command to create a file system:

mkfs.*File system format Logical volume path*

In this example, run the following command:

mkfs.ext4 /dev/vgdata/lvdata1

Information similar to the following is displayed:

```
[root@ecs-lvmtest ~]# mkfs.ext4 /dev/vgdata/lvdata1
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
983040 inodes, 3932160 blocks
196608 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2151677952
120 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

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

Step 3 Run the following command to create a mounting directory:

mkdir *Mounting directory*

In this example, run the following command:

mkdir /Data1

Step 4 Run the following command to mount the file system on the directory:

mount *Logical volume path Mounting directory*

In this example, run the following command:

mount /dev/vgdata/lvdata1 /Data1

Step 5 Run the following command to query the file system mounting information:

mount | grep *Mounting directory*

In this example, run the following command:

mount | grep /Data1

Information similar to the following is displayed:

```
[root@ecs-lvmtest ~]# mount | grep /Data1
/dev/mapper/vgdata-lvdata1 on /Data1 type ext4 (rw,relatime,data=ordered)
```

In the command output, **dev/mapper/vgdata-lvdata1** indicates the file system path. Take note of this path, which will be used in **Step 6**.

Step 6 Perform the following operations to enable automatic mounting of the file system at the system start:

If this is not configured, you need to manually mount the file system every time the ECS is restarted.

1. Run the following command to query the file system UUID:

   blkid *File system path*

   In this example, run the following command to query the UUID of **dev/mapper/vgdata-lvdata1**:

   blkid /dev/mapper/vgdata-lvdata1

   Information similar to the following is displayed:

   ```
   [root@ecs-lvmtest ~]# blkid /dev/mapper/vgdata-lvdata1
   /dev/mapper/vgdata-lvdata1: UUID="c6a243ce-5150-41ac-8816-39db54d1a4b8" TYPE="ext4"
   ```

   In the command output, the UUID is **c6a243ce-5150-41ac-8816-39db54d1a4b8**.

2. Run the following command to open the **/etc/fstab** file:

   vi /etc/fstab

   Information similar to the following is displayed:

   ```
   [root@ecs-lvmtest ~]# vi /etc/fstab

   #
   # /etc/fstab
   # Created by anaconda on Tue Nov  7 14:28:26 2017
   #
   # Accessible filesystems, by reference, are maintained under '/dev/disk'
   # See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
   #
   UUID=27f9be47-838b-4155-b20b-e4c5e013cdf3 /                       ext4    defaults        1 1
   UUID=2b2000b1-f926-4b6b-ade8-695ee244a901 /boot                   ext4    defaults        1 2
   ```

3. Press **i** to enter editing mode.

4. Move the cursor to the end of the file and press **Enter**. Then add the following information:

   ```
   UUID=c6a243ce-5150-41ac-8816-39db54d1a4b8 /Data1                  ext4    defaults        0 0
   ```

   The file content is described as follows:

   – Column 1: indicates the UUID. Enter the UUID queried in **1**.

   – Column 2: indicates the file system's mounting directory. Enter mounting directory **/Data1** created in **Step 3**.

   – Column 3: indicates the file system format. Enter file system format **ext4** configured in **Step 2**.

   – Column 4: indicates the mounting option. In this example, **defaults** is used.

   – Column 5: indicates the backup option. Enter either **1** (the system automatically backs up the file system) or **0** (does not back up the file system). In this example, **0** is used.

   – Column 6: indicates the scanning option. Enter either **1** (the system automatically scans the file system at system start) or **0** (does not scan the file system). In this example, **0** is used.

5. Press **Esc**, enter **:wq!**, and press **Enter**.

   The system saves the modifications and exits the vi editor.

Step 7 Perform the following operations to verify automatic mounting:

1. Run the following command to unmount a file system:

   umount *Logical volume path*

   In this example, run the following command:

   umount /dev/vgdata/lvdata1

2. Run the following command to reload all the content in the **/etc/fstab** file:

   mount -a

3. Run the following command to query the file system mounting information:

   mount | grep *Mounting directory*

   In this example, run the following command:

   mount | grep /Data1

   If information similar to the following is displayed, the automatic mounting function takes effect:

   ```
   [root@ecs-lvmtest ~]# mount | grep /Data1
   /dev/mapper/vgdata-lvdata1 on /Data1 type ext4 (rw,relatime,data=ordered)
   ```

----End

1.4.4 Extending the Logical Volume Using the Unallocated Space

Scenarios

If the logical volume space becomes insufficient, you can extend the logical volume. This section shows how to add 4 GB space to a 15 GB logical volume, which no longer meets requirements.

NOTE

During the extension, ensure that the volume group has sufficient available space to extend the logical volume. If the volume group's available space is also insufficient, extend the volume group according to **Extending the Logical Volume by Expanding Capacity of an EVS Disk** or **Extending the Volume Group by Adding an EVS Disk**.

Procedure

Step 1 Log in to the ECS as user **root**.

Step 2 Run the following command to extend the logical volume:

lvextend -L +*Additional capacity Logical volume path*

Parameter description:

- *Additional capacity*: Specify a value smaller than the volume group's available space, either in MB or GB.
- *Logical volume path*: Specify the path of the to-be-extended logical volume.

In this example, run the following command:

lvextend -L +4GB /dev/vgdata/lvdata1

Information similar to the following is displayed:

```
[root@ecs-lvmtest ~]# lvextend -L +4GB /dev/vgdata/lvdata1
  Size of logical volume vgdata/lvdata1 changed from 15.00 GiB (3840 extents) to 19.00 GiB (4864 extents).
  Logical volume vgdata/lvdata1 successfully resized.
```

This step only extends the logical volume. You also need to extend the size of the file system on this volume.

Step 3 Run the following command to extend the size of the file system:

resize2fs *Logical volume path*

In this example, run the following command:

resize2fs /dev/vgdata/lvdata1

Information similar to the following is displayed:

```
[root@ecs-lvmtest ~]# resize2fs /dev/vgdata/lvdata1
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/vgdata/lvdata1 is mounted on /Data1; on-line resizing required
old_desc_blocks = 4, new_desc_blocks = 28
The filesystem on /dev/vgdata/lvdata1 is now 3657728 blocks long.
```

Step 4 Run the following command to check whether the file system size increases:

df -h

Information similar to the following is displayed:

```
[root@ecs-lvmtest ~]# df -h
Filesystem                  Size  Used Avail Use% Mounted on
/dev/vda2                    39G  1.5G   35G   5% /
devtmpfs                    487M     0  487M   0% /dev
tmpfs                       496M     0  496M   0% /dev/shm
tmpfs                       496M  6.7M  490M   2% /run
tmpfs                       496M     0  496M   0% /sys/fs/cgroup
/dev/vda1                   976M  131M  779M  15% /boot
tmpfs                       100M     0  100M   0% /run/user/0
/dev/mapper/vgdata-lvdata1   19G   44M   18G   1% /Data1
```

In the command output, the size of file system **/dev/mapper/vgdata-lvdata1** increases by 4 GB.

----End

1.4.5 Extending the Logical Volume by Expanding Capacity of an EVS Disk

Scenarios

If the logical volume space becomes insufficient, you can extend the logical volume. This section describes how to add 10 GB space to a 19 GB logical volume by expanding the capacity of an EVS disk.

Procedure

Step 1 Expand the capacity of an EVS disk on the management console.

1. Log in to the management console.
2. Under **Storage**, click **Elastic Volume Service**. The disk list page is displayed.
3. Locate the to-be-expanded disk and expand the capacity.

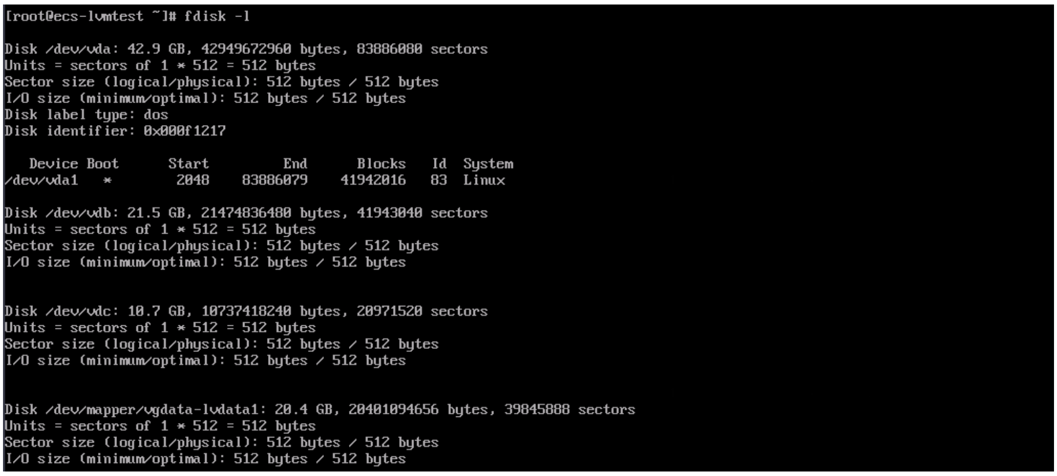
   For details, see **Expand Disk Capacity**.

Step 2 Log in to the ECS as user **root**.

Step 3 Run the following command to check whether the system has identified the added space:

fdisk -l

Information similar to the following is displayed:



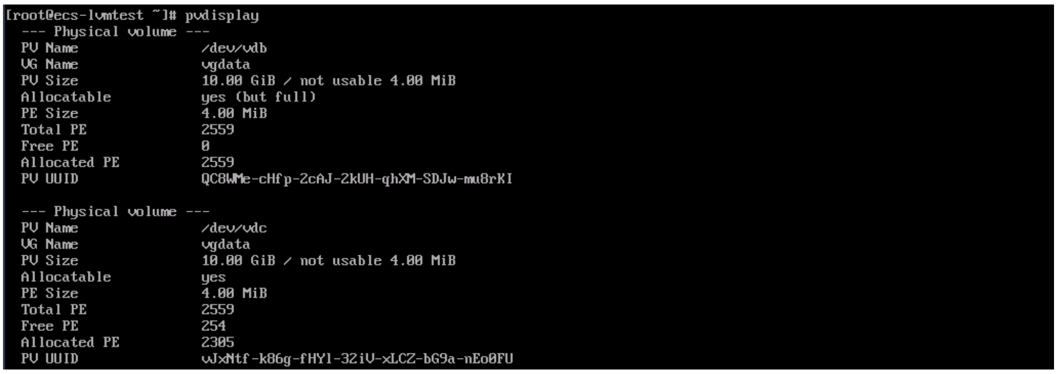
```
[root@ecs-lvmtest ~]# fdisk -l

Disk /dev/vda: 42.9 GB, 42949672960 bytes, 83886080 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000f1217

   Device Boot      Start         End      Blocks   Id  System
/dev/vda1   *        2048    83886079    41942016   83  Linux

Disk /dev/vdb: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/vdc: 10.7 GB, 10737418240 bytes, 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/mapper/vgdata-lvdata1: 20.4 GB, 20401094656 bytes, 39845888 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

The size of **/dev/vdb** has increased from 10 GB to 20 GB.

Step 4 Run the following command to view information of physical volumes:

pvdisplay

Information similar to the following is displayed:



```
[root@ecs-lvmtest ~]# pvdisplay
  --- Physical volume ---
  PV Name               /dev/vdb
  VG Name               vgdata
  PV Size               10.00 GiB / not usable 4.00 MiB
  Allocatable           yes (but full)
  PE Size               4.00 MiB
  Total PE              2559
  Free PE               0
  Allocated PE          2559
  PV UUID               QC8WMe-cHfp-2cAJ-2kUH-qhXM-SDJw-mu8rKI

  --- Physical volume ---
  PV Name               /dev/vdc
  VG Name               vgdata
  PV Size               10.00 GiB / not usable 4.00 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              2559
  Free PE               254
  Allocated PE          2305
  PV UUID               wJxNtf-k86g-fHYl-32iV-xLCZ-bG9a-nEo0FU
```

The size of /**dev/vdb** remains 10 GB, indicating that the size of the physical volume is not increased.

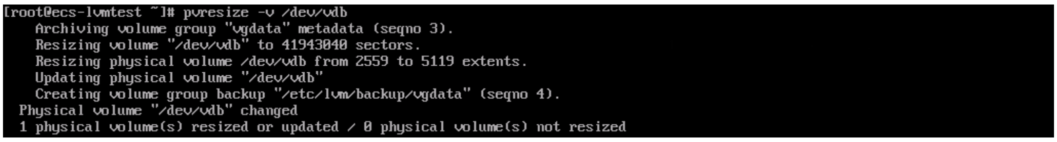
Step 5 Run the following command to extend the physical volume of the corresponding EVS disk:

pvresize -v *Disk device name*

In this example, run the following command:

pvresize -v /dev/vdb

Information similar to the following is displayed:



```
[root@ecs-lvmtest ~]# pvresize -v /dev/vdb
    Archiving volume group "vgdata" metadata (seqno 3).
    Resizing volume "/dev/vdb" to 41943040 sectors.
    Resizing physical volume /dev/vdb from 2559 to 5119 extents.
    Updating physical volume "/dev/vdb"
    Creating volume group backup "/etc/lvm/backup/vgdata" (seqno 4).
  Physical volume "/dev/vdb" changed
  1 physical volume(s) resized or updated / 0 physical volume(s) not resized
```

In the command output, the physical volume corresponding to **/dev/vdb** has been extended.

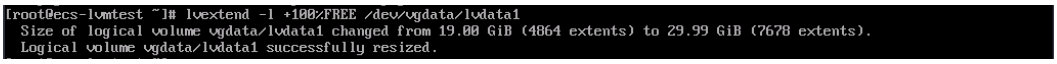
Step 6 Run the following command to extend the corresponding logical volume if needed:

lvextend -l +100%FREE *Logical volume path*

In this example, run the following command:

lvextend -l +100%FREE /dev/vgdata/lvdata1

Information similar to the following is displayed:



```
[root@ecs-lvmtest ~]# lvextend -l +100%FREE /dev/vgdata/lvdata1
  Size of logical volume vgdata/lvdata1 changed from 19.00 GiB (4864 extents) to 29.99 GiB (7678 extents).
  Logical volume vgdata/lvdata1 successfully resized.
```

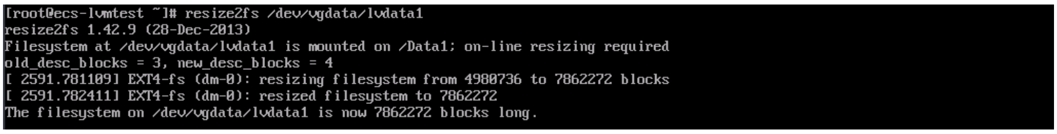
Step 7 Run the following command to extend the file system of the partition:

resize2fs *Logical volume path*

In this example, run the following command:

resize2fs /dev/vgdata/lvdata1

Information similar to the following is displayed:

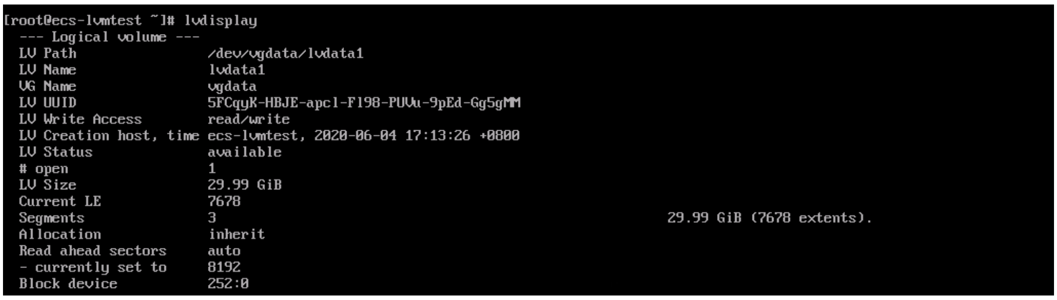


```
[root@ecs-lvmtest ~]# resize2fs /dev/vgdata/lvdata1
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/vgdata/lvdata1 is mounted on /Data1; on-line resizing required
old_desc_blocks = 3, new_desc_blocks = 4
[ 2591.781109] EXT4-fs (dm-0): resizing filesystem from 4980736 to 7862272 blocks
[ 2591.782411] EXT4-fs (dm-0): resized filesystem to 7862272
The filesystem on /dev/vgdata/lvdata1 is now 7862272 blocks long.
```

Step 8 Run the following command to view the capacity expansion result:

lvdisplay

Information similar to the following is displayed:



```
[root@ecs-lvmtest ~]# lvdisplay
  --- Logical volume ---
  LV Path                /dev/vgdata/lvdata1
  LV Name                lvdata1
  VG Name                vgdata
  LV UUID                5FCqyK-HBJE-apcl-F198-PUVu-9pEd-Gg5gMM
  LV Write Access        read/write
  LV Creation host, time ecs-lvmtest, 2020-06-04 17:13:26 +0800
  LV Status              available
  # open                 1
  LV Size                29.99 GiB
  Current LE             7678
  Segments               3                                                       29.99 GiB (7678 extents).
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           252:0
```

In the command output, the logical volume size (**LV Size**) is increased by 10 GB.

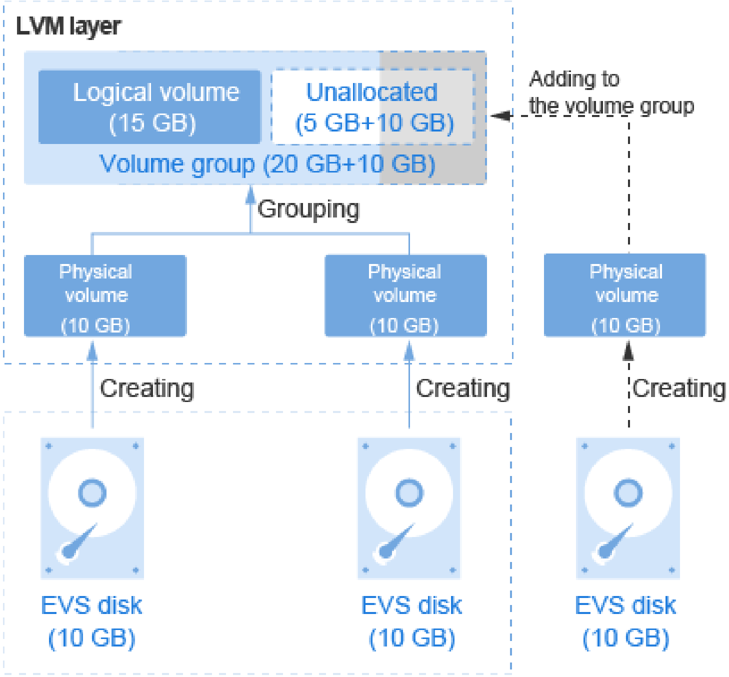
----End

1.4.6 Extending the Volume Group by Adding an EVS Disk

Scenarios

If the space of an LVM volume group no longer meets your needs, you can extend the volume group by adding new EVS disks, creating physical volumes, and adding the physical volumes to the volume group.

Figure 1-4 Example of extending a volume group



Procedure

Step 1 Create an EVS disk and attach it.

1. Log in to the management console.
2. Under **Storage**, click **Elastic Volume Service**. The disk list page is displayed.
3. Click **Buy Disk** and create a disk.

   For details, see **Purchasing an EVS Disk**.
4. In the disk list, locate the new disk and click **Attach** in the **Operation** column.
5. On the displayed page, select the target ECS and select a device name from the drop-down list. Ensure that the EVS disk and ECS reside in the same AZ.

   Return to the disk list page. The status of the disk is **Attaching**, indicating that the disk is being attached to the ECS. When the disk status changes to **In-use**, the disk is successfully attached.

Step 2 Log in to the ECS as user **root**.

Step 3 Run the following command to query the volume group size:

vgdisplay

Information similar to the following is displayed:

```
[root@ecs-lvmtest ~]# vgdisplay
 --- Volume group ---
 VG Name               vgdata
 System ID
 Format                lvm2
 Metadata Areas        2
 Metadata Sequence No  3
 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               19.99 GiB
 PE Size               4.00 MiB
 Total PE              5118
 Alloc PE / Size       4864 / 19.00 GiB
 Free  PE / Size       254 / 1016.00 MiB
 VG UUID               NLkZV7-hYYE-0w66-tnlt-Y6jL-Ik7S-76w4P6
```

In the command output, the **VG Size** value indicates the volume group size, which is **19.99 GiB**.

Step 4 Run the following command to view and take note of the device names:

fdisk -l | grep /dev/vd | grep -v vda

Information similar to the following is displayed:

```
[root@ecs-lvmtest ~]# fdisk -l | grep /dev/vd | grep -v vda
Disk /dev/vdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Disk /dev/vdc: 10.7 GB, 10737418240 bytes, 20971520 sectors
Disk /dev/vdd: 10.7 GB, 10737418240 bytes, 20971520 sectors
```

In the command output, the new EVS disk has been attached to the ECS, and the device name is **/dev/vdd**.

Step 5 Run the following command to create a physical volume using the new EVS disk:

pvcreate *Disk device name*

In this example, run the following command:

pvcreate /dev/vdd

Information similar to the following is displayed:

```
[root@ecs-lvmtest ~]# pvcreate /dev/vdd
 Physical volume "/dev/vdd" successfully created.
```

Step 6 Run the following command to extend the volume group by adding the physical volume to the volume group:

vgextend *Volume group name Physical volume name*

In this example, run the following command:

vgextend vgdata /dev/vdd

Information similar to the following is displayed:

```
[root@ecs-lvmtest ~]# vgextend vgdata /dev/vdd
 Volume group "vgdata" successfully extended
```

Step 7 Run the following command to query details of the volume group:

vgdisplay

Information similar to the following is displayed:

```
[root@ecs-lvmtest ~]# vgdisplay
 --- Volume group ---
 VG Name               vgdata
 System ID
 Format                lvm2
 Metadata Areas        3
 Metadata Sequence No  4
 VG Access             read/write
 VG Status             resizable
 MAX LV                0
 Cur LV                1
 Open LV               1
 Max PV                0
 Cur PV                3
 Act PV                3
 VG Size               <29.99 GiB
 PE Size               4.00 MiB
 Total PE              7677
 Alloc PE / Size       4864 / 19.00 GiB
 Free  PE / Size       2813 / <10.99 GiB
 VG UUID               NLkZV7-hYYE-0w66-tnlt-Y6jL-Ik7S-76w4P6
```

In the command output, 10 GB has been added to the **VG Size** volume group.

----End

2 RAID Array Creation with EVS Disks

2.1 Overview of Using EVS Disks to Create a RAID Array

Redundant Array of Independent Disks (RAID) is a technology that combines multiple physical disks into one or more logical units for the purposes of data redundancy and performance improvement.

NOTE

In this document, Elastic Volume Service (EVS) disks instead of physical disks are used to create RAID arrays. The working principles are the same.

This document uses CentOS 7.5 as the sample OS to describe how to create a RAID 10 array with four EVS disks. A RAID 10 array consists of RAID 0 and RAID 1 arrays. In this example, EVS disks are used to create a mirroring array (RAID 1) and then create a RAID 0 array to store data in stripes. At least four EVS disks are required. The resource information is as follows:

- Resource planning: **Resource Planning**
- Resource creation: **Creating an ECS** and **Creating and Attaching EVS Disks**

Introduction to Common RAID Arrays

Table 2-1 Introduction to common RAID arrays

RAID Level	Description	Read/Write Performance	Security	Disk Usage	Min. Number of Disks Required
RAID 0	RAID 0 stores data on multiple disks, implementing parallel read/write and providing the fastest read/write speed.	Parallel read/write from multiple disks achieves high performance.	Worst No redundancy capability. If one disk is damaged, the data of the entire RAID array is unavailable.	100%	2
RAID 1	RAID 1 implements data redundancy based on data mirroring. Half of the disk capacity in the RAID array is used, and the other half is used for mirroring to provide data backup.	Read performance: Same as a single disk Write performance: Data needs to be written into two disks. The write performance is lower than that of a single disk.	Highest Provides full backup of disk data. If a disk in the RAID array fails, the system automatically uses the data on the mirror disk.	50%	2
RAID 01	RAID 01 combines RAID 0 and RAID 1, in which half disks are first grouped into RAID 0 stripes and then used together with the other half to set up a RAID 1 array.	Read performance: Same as RAID 0 Write performance: Same as RAID 1	The security of RAID 01 is lower than that of RAID 10.	50%	4

RAID Level	Description	Read/Write Performance	Security	Disk Usage	Min. Number of Disks Required
RAID 10	RAID 10 combines RAID 1 and RAID 0, in which half disks are first set up as a RAID 1 array and then used together with the other half to create RAID 0 stripes.	Read performance: Same as RAID 0 Write performance: Same as RAID 1	The security performance of RAID 10 is the same as that of RAID 1.	50%	4
RAID 5	RAID 5 does not specify a dedicated parity disk and consists of block-level striping with parity information distributed among the disks.	Read performance: Same as RAID 0 Write performance: Because parity data needs to be written into disks, the write performance is lower than that of a single disk.	The security of RAID 5 is lower than that of RAID 10.	66.7%	3

2.2 Resource Planning

This topic describes the servers and disks planned for creating a RAID 10 array.

Servers

In this example, one Elastic Cloud Server (ECS) is created, and **Table 2-2** shows the parameter specifications.

Table 2-2 ECS parameter configurations

Parameter	Configuration Information
Name	ecs-raid10
Image	CentOS 7.5 64bit

Parameter	Configuration Information
Specifications	General computing and s2.medium.2 (1 vCPU and 2 GiB memory)
Elastic IP Address (EIP)	139.*XX.XX.XX*
Private IP Address	192.168.1.189

EVS Disks

Setting up RAID 10 requires at least 4 disks. Therefore, 4 EVS disks are created and attached to the ECS in this example.

2.3 Implementation Procedure

2.3.1 Creating an ECS

Scenarios

This section shows how to create an ECS. In this example, one ECS needs to be created. For details about the ECS parameter configurations, see **Resource Planning**.

Procedure

Step 1 Log in to the management console.

Step 2 Choose **Compute** > **Elastic Cloud Server**.

The **Elastic Cloud Server** page is displayed.

Step 3 Click **Buy ECS**.

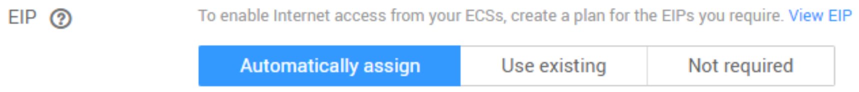
For details, see the *Elastic Cloud Server User Guide*.

Configure the following parameters as planned:

- **Image**: Select **CentOS 7.5 64bit**.
- **EIP**: An EIP is mandatory if the ECS needs to access the public network. In this example, the multiple devices admin (mdadm) tool needs to be installed. Therefore, an EIP must be configured. Buy an EIP or configure an existing one based on the environment condition.

  Figure 2-1 shows how to buy a new EIP.

Figure 2-1 Configuring EIP



EIP ⓘ

To enable Internet access from your ECSs, create a plan for the EIPs you require. View EIP

Table 2-3 shows the ECS parameter configurations.

Table 2-3 ECS parameter configurations

ECS Parameter Configurations		Billing Mode	Quantity
Specifications	General computing | s2.medium.2 | 1 vCPU | 2 GiB	Pay-per-use	1
Image	CentOS 7.5 64bit		
System disk	High I/O, 40 GB		
VPC	vpc-1a55		
Security group	Sys-default		
NIC	subnet-1a55(192.168.1.0/24)		
EIP	Specifications: Static BGP Billing mode: By bandwidth (Bandwidth: 5 Mbit/s)		
ECS Name	ecs-raid10		

----End

2.3.2 Creating and Attaching EVS Disks

Scenarios

This section shows how to create four EVS disks in a batch and attach the disks to the ECS.

Procedure

Step 1 Log in to the management console.

Step 2 Under **Storage**, click **Elastic Volume Service**.

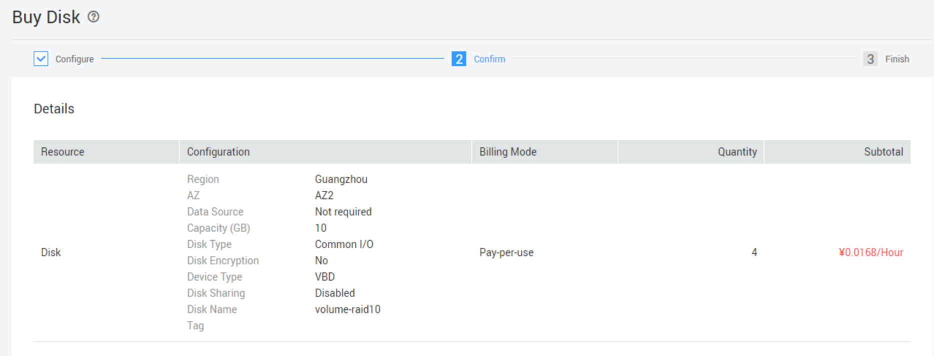
The disk list page is displayed.

Step 3 Click **Buy Disk** and create a disk.

For details, see section "Create an EVS Disk" in the *Elastic Volume Service User Guide*.

In this example, four EVS disks are created in a batch. Figure 2-2 shows the detailed parameter configurations.

Figure 2-2 EVS disk specifications



Buy Disk

☑ Configure — 2 Confirm — 3 Finish

Details

Resource	Configuration		Billing Mode	Quantity	Subtotal
Disk	Region AZ Data Source Capacity (GB) Disk Type Disk Encryption Device Type Disk Sharing Disk Name Tag	Guangzhou AZ2 Not required 10 Common I/O No VBD Disabled volume-raid10	Pay-per-use	4	¥0.0168/Hour

Step 4 Attach the disks to the ECS.

----End

2.3.3 Creating an EVS-based RAID Array Using mdadm

Scenarios

This section shows how to create a RAID 10 array using mdadm.

In this example, the ECS runs CentOS 7.5 64bit. Configurations vary depending on the OS running on the ECS. This section is used for reference only. For the detailed operations and differences, see the corresponding OS documents.

Procedure

Step 1 Log in to the ECS as user **root**.

Step 2 Run the following command to view and take note of the device names:

fdisk -l | grep /dev/vd | grep -v vda

Information similar to the following is displayed:

```
[root@ecs-raid10 ~]# fdisk -l | grep /dev/vd | grep -v vda
Disk /dev/vdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Disk /dev/vdc: 10.7 GB, 10737418240 bytes, 20971520 sectors
Disk /dev/vdd: 10.7 GB, 10737418240 bytes, 20971520 sectors
Disk /dev/vde: 10.7 GB, 10737418240 bytes, 20971520 sectors
```

In the command output, four disks are attached to the ECS, and the device names are **/dev/vdb**, **/dev/vdc**, **/dev/vdd**, and **/dev/vde**, respectively.

Step 3 Run the following command to install mdadm:

yum install mdadm -y

NOTE

mdadm is a utility to create and manage software RAID arrays on Linux. Ensure that an EIP has been bound to the ECS where mdadm is to be installed.

Information similar to the following is displayed:

```
[root@ecs-raid10 ~]# yum install mdadm -y
......
Installed:
  mdadm.x86_64 0:4.0-13.el7
```

```
Dependency Installed:
  libreport-filesystem.x86_64 0:2.1.11-40.el7.centos

Complete!
```

Step 4 Run the following command to create a RAID array using the four disks queried in **Step 2**:

mdadm -Cv *RAID array device name* **-a yes -n** *Disk quantity* **-l** *RAID level Device name of disk1 Device name of disk2 Device name of disk3 Device name of disk4*

Parameter description:

- *RAID array device name*: The value can be user-definable. In this example, **/dev/md0** is used.
- *Disk quantity*: Set this parameter based on the actual condition. In this example, RAID 10 is created, and at least four disks are required.

  The minimum number of disks required varies depending on the RAID level. For details, see **Overview of Using EVS Disks to Create a RAID Array**.
- *RAID level*: Set this parameter based on the actual condition. In this example, set it to RAID 10.
- *Device name of the disk*: Enter the device names of all the disks that will be used to create the RAID array. Multiple names are separated with spaces.

In this example, run the following command:

mdadm -Cv /dev/md0 -a yes -n 4 -l 10 /dev/vdb /dev/vdc /dev/vdd /dev/vde

Information similar to the following is displayed:

```
[root@ecs-raid10 ~]# mdadm -Cv /dev/md0 -a yes -n 4 -l 10 /dev/vdb /dev/vdc /dev/vdd /dev/vde
mdadm: layout defaults to n2
mdadm: layout defaults to n2
mdadm: chunk size defaults to 512K
mdadm: size set to 10476544K
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

Step 5 Run the following command to format the created RAID array:

mkfs.*File system format Device name of the RAID array*

In this example, run the following command:

mkfs.ext4 /dev/md0

Information similar to the following is displayed:

```
[root@ecs-raid10 ~]# mkfs.ext4 /dev/md0
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=128 blocks, Stripe width=256 blocks
1310720 inodes, 5238272 blocks
261913 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2153775104
160 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
```

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

Step 6 Run the following command to create a mounting directory:

mkdir *Mount point*

In this example, run the following command:

mkdir /RAID10

Step 7 Run the following command to mount the RAID array:

mount *RAID array device name Mounting directory*

In this example, run the following command:

mount /dev/md0 /RAID10

Step 8 Run the following command to view the mount result:

df -h

Information similar to the following is displayed:

```
[root@ecs-raid10 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/vda2        39G  1.5G   35G   5% /
devtmpfs        911M     0  911M   0% /dev
tmpfs           920M     0  920M   0% /dev/shm
tmpfs           920M  8.6M  911M   1% /run
tmpfs           920M     0  920M   0% /sys/fs/cgroup
/dev/vda1       976M  146M  764M  17% /boot
tmpfs           184M     0  184M   0% /run/user/0
/dev/md0         20G   45M   19G   1% /RAID10
```

Step 9 Perform the following operations to enable automatic mounting of the RAID array at the system start:

1. Run the following command to open the **/etc/fstab** file:

   vi /etc/fstab

2. Press **i** to enter editing mode.

   Information similar to the following is displayed:

   ```
   [root@ecs-raid10 ~]# vi /etc/fstab

   #
   # /etc/fstab
   # Created by anaconda on Tue Nov  7 14:28:26 2017
   #
   # Accessible filesystems, by reference, are maintained under '/dev/disk'
   # See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
   #
   UUID=27f9be47-838b-4155-b20b-e4c5e013cdf3 /                       ext4    defaults        1 1
   UUID=2b2000b1-f926-4b6b-ade8-695ee244a901 /boot                   ext4    defaults        1 2
   ```

3. Add the following information to the end of the file:

   ```
   /dev/md0                                  /RAID10                 ext4    defaults        0 0
   ```

4. Press **Esc**, enter **:wq!**, and press **Enter**.

   The system saves the modifications and exits the vi editor.

Step 10 Run the following command to view the RAID array information:

mdadm -D *RAID array device name*

In this example, run the following command:

mdadm -D /dev/md0

Information similar to the following is displayed:

```
[root@ecs-raid10 ~]# mdadm -D /dev/md0
/dev/md0:
           Version : 1.2
     Creation Time : Thu Nov  8 15:49:02 2018
        Raid Level : raid10
        Array Size : 20953088 (19.98 GiB 21.46 GB)
     Used Dev Size : 10476544 (9.99 GiB 10.73 GB)
      Raid Devices : 4
     Total Devices : 4
       Persistence : Superblock is persistent

       Update Time : Thu Nov  8 16:15:11 2018
             State : clean
    Active Devices : 4
   Working Devices : 4
    Failed Devices : 0
     Spare Devices : 0

            Layout : near=2
        Chunk Size : 512K

Consistency Policy : resync

              Name : ecs-raid10.novalocal:0  (local to host ecs-raid10.novalocal)
              UUID : f400dbf9:60d211d9:e006e07b:98f8758c
            Events : 19

    Number   Major   Minor   RaidDevice State
       0     253       16        0      active sync set-A   /dev/vdb
       1     253       32        1      active sync set-B   /dev/vdc
       2     253       48        2      active sync set-A   /dev/vdd
       3     253       64        3      active sync set-B   /dev/vde
```

----End

2.3.4 Configuring Auto Start of the EVS-based RAID Array at Server Startup

Scenarios

This section shows how to add RAID array information, such as the device name and UUID to the mdadm configuration file. In this case, the RAID array can be started by querying information in the configuration file when the system starts.

In this example, the ECS runs CentOS 7.5 64bit. Configurations vary depending on the OS running on the ECS. This section is used for reference only. For the detailed operations and differences, see the corresponding OS documents.

Procedure

Step 1 Log in to the ECS as user **root**.

Step 2 Run the following command to view the RAID array information:

mdadm --detail --scan

Information similar to the following is displayed:

```
[root@ecs-raid10 ~]# mdadm --detail --scan
ARRAY /dev/md0 metadata=1.2 name=ecs-raid10.novalocal:0 UUID=f400dbf9:60d211d9:e006e07b:98f8758c
```

Step 3 Perform the following operations to add information of the new RAID array to the mdadm file:

1. Run the following command to open the **mdadm.conf** file:

   vi /etc/mdadm.conf

2. Press **i** to enter editing mode.

3. Add the following information to the end of the file:

   ```
   DEVICE /dev/vdb /dev/vdc /dev/vdd /dev/vde
   ARRAY /dev/md0 metadata=1.2 name=ecs-raid10.novalocal:0
   UUID=f400dbf9:60d211d9:e006e07b:98f8758c
   ```

   Description:

   – DEVICE line: Indicates the device names of the disks that set up the RAID array. Multiple device names are separated with spaces.

   – ARRAY line: Indicates RAID array information. Input the RAID array information obtained in **Step 2**.

   NOTE

   The preceding information is used for reference only. Add RAID array information based on the site information.

4. Press **Esc**, enter **:wq!**, and press **Enter**.

   The system saves the modifications and exits the vi editor.

Step 4 Run the following command to check whether the **mdadm.conf** file is modified:

more /etc/mdadm.conf

Information similar to the following is displayed:

```
[root@ecs-raid10 ~]# more /etc/mdadm.conf
DEVICE /dev/vdb /dev/vdc /dev/vdd /dev/vde
ARRAY /dev/md0 metadata=1.2 name=ecs-raid10.novalocal:0 UUID=f400dbf9:60d211d9:e006e07b:98f8758c
```

If the information added in **Step 3** is displayed, the file is successfully modified.

----End