Container Service

FAQ

FAQ

General

Swarm FAQs

How to increase data disks for Container Service Docker?

Docker data is stored on disks by using the union file system. If the number of containers or images needing to be run on the machines is continuously increased, the disk size might not meet the requirements. In this situation, increase the data disks to expand the storage space for Docker data directory.

Docker data directory

For Docker, the container and image data is stored in the /var/lib/docker directory by default. You can check the currently occupied disk size of this directory by running the du command.

For example:

```
du -h --max-depth=0 /var/lib/docker
7.9G /var/lib/docker
```

Change Docker data disk

Many Docker images are big. Therefore, several images might occupy large disk space, which leads to

insufficient disk space. By increasing the data disks for the Docker data directory, the requirements of increasing images or containers continuously can be met.

Purchase ECS data disk and mount to machines needing expansion

Log on to the Elastic Compute Service (ECS) console to purchase the cloud disk with corresponding configurations.

Click **Instances** in the left-side navigation pane.

Select the region and then click the instance name or click **Manage** at the right of the instance.

Click **Instance Disks** in the left-side navigation pane.

Click **Attach Disk** in the upper-right corner.

Select the purchased disk and record the mount point /dev/xvd* or /dev/vd*.

Determine the mount point by running the cd command. The mount point of the I/O optimized instance is /dev/vd*.

Log on to the machine and format the mounted disk

Run ls -l /dev/xvd* or ls -l /dev/vd* on the machine to view the disk ID, which is consistent with your recorded one.

Partition the disk by running the fdisk command. Then, format the disk by using mkfs.ext4. For more information, see Linux _ Format and mount a data disk.

For example:

```
 root@iZbp16h1ijt5er5wempg4sZ:~# ls -l /dev/vd*
brw-rw---- 1 root disk 253, 0 Jan 5 17:44 /dev/vda
brw-rw---- 1 root disk 253, 1 Jan 5 17:44 /dev/vda1
brw-rw---- 1 root disk 253, 16 Jan 5 17:55 /dev/vdb
root@iZbp16h1ijt5er5wempg4sZ:~# fdisk -S 56 /dev/vdb

Welcome to fdisk (util-linux 2.27.1).
Changes will remain in memory only, until you decide to write them.
```

```
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x44e128c4.

Command (m for help): n
Partition type
p primary (0 primary, 0 extended, 4 free)
e extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-41943039, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-41943039, default 41943039):

Created a new partition 1 of type 'Linux' and of size 20 GiB.

Command (m for help): wq
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

root@iZbp16h1ijt5er5wempg4sZ:~# ll /dev/vd*
brw-rw---- 1 root disk 253, 0 Jan 5 17:44 /dev/vda
brw-rw---- 1 root disk 253, 1 Jan 5 17:44 /dev/vda1
brw-rw---- 1 root disk 253, 16 Jan 5 17:58 /dev/vdb
brw-rw---- 1 root disk 253, 17 Jan 5 17:58 /dev/vdb1 ## Add partition
root@iZbp16h1ijt5er5wempg4sZ:~# mkfs.ext4 /dev/vdb1 ## Format
mke2fs 1.42.13 (17-May-2015)
Creating filesystem with 5242624 4k blocks and 1310720 inodes
Filesystem UUID: cef1625c-7533-4308-bc44-511580e3edc8
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
```

Move Docker data to a new disk

Stop Docker daemon first to guarantee the data integrity in the process of moving Docker data. Run the command service docker stop to stop the Docker daemon.

Move the data in the Docker default data directory to a backup directory. For example, if the backup directory is /var/lib/docker_data, run the command mv /var/lib/docker /var/lib/docker_data.

Mount the newly formatted disk to the /var/lib/docker directory. Run the command echo

"/dev/vdb1 /var/lib/docker ext4 defaults 0 0" >>/etc/fstab && mkdir /var/lib/docker && mount -a.

Move the backed up Docker data to the new disk. Run the command mv /var/lib/docker_data/* /var/lib/docker/.

Start Docker daemon and check data location

Start Docker daemon and run the command service docker start.

Run the command df. You can see /var/lib/docker is mounted to the new disk.

```
root@iZbp16h1ijt5er5wempg4sZ:/# df -h
Filesystem Size Used Avail Use% Mounted on
udev 2.0G 0 2.0G 0% /dev
tmpfs 396M 7.1M 389M 2% /run
/dev/vda1 40G 2.7G 35G 8% /
tmpfs 2.0G 476K 2.0G 1% /dev/shm
tmpfs 5.0M 0 5.0M 0% /run/lock
tmpfs 2.0G 0 2.0G 0% /sys/fs/cgroup
tmpfs 396M 0 396M 0% /run/user/0
/dev/vdb1 20G 2.1G 17G 12% /var/lib/docker ## This directory is mounted to the new disk.
....
```

Run the command docker ps to check if containers are lost. Restart the related containers as required, for example, the containers without configuring the restart:always label.

How to troubleshoot log issues?

If the extension label label aliyun.log_store_xxx: xxx is added in the application but no logs are collected in Log Service, follow these steps to troubleshoot the issue:

Note: Troubleshoot the issue by following these steps and do not skip the steps.

1. Check whether or not Logstore is successfully created

The application is not successfully deployed if Logstore is not created. Check if any error message

about deployment is in the application **Events**.

2. Check the ilogtail version

Run the command docker ps|grep ilogtail on the machine and determine the version of the image ilogtail according to the output. If the version is 0.11.6, upgrade the system services to the latest version (currently the latest version is 0.13.4). After the upgrade, query the logs in the Log Service console after the application generates new logs.

3. Check the ilogtail logs

Run the command docker exec -it <ilogtail container ID> cat /usr/local/ilogtail/ilogtail.LOG and determine what the issue is according to the ilogtail logs. Common possible reasons include:

Network is not connected. Determine whether the network is connected or not by running the following command:

```
Classic:
telnet logtail.cn-<region>-intranet.log.aliyuncs.com 80

VPC:
telnet logtail.cn-<region>-vpc.log.aliyuncs.com 80

Internet:
telnet logtail.cn-<region>.log.aliyuncs.com 80
```

AccessKey is not configured.

Unauthorized ErrorMessage:no authority, denied by ACL appears in the logs if the main account does not configure the AccessKey. Create the AccessKey for the main account first. Check whether or not the main account configures the AccessKey even if Unauthorized ErrorMessage:no authority, denied by ACL does not appear in the logs.

4. Check whether or not the machine IP is in the Log Service machine group

1. Log on to the Log Service console.
2. Click the name of the log project that corresponds to the cluster. The project name rule is acslog-project-<first 10 letters of the cluster ID>.
3. On the project details page, click **Logtail Machine Group** in the left-side navigation pane.
4. Click **Machine Status** at the right of the machine group and check if the IP of the current

machine is in the IP list.

5. Check whether or not the main account configures the AccessKey

Make sure at least one enabled AccessKey is under the main account.

6. Check whether or not the log file has contents

Enter the business application container and determine whether or not logs are actually generated. For stdout logs, use the docker logs command directly.

7. Check whether or not the cluster is a swarm mode cluster

If the cluster is a swarm mode cluster, the log feature currently does not support applications deployed by using Docker Compose V1 and V2.