

Storage Troubleshooting Case Studies for CloudOps Engineers

10 Real-world Storage and Filesystem Scenarios with Commands, Logs, and Solutions

Case 1: Disk Full Due to Log Overflow

Problem: Application downtime caused by a full root filesystem.

Investigation:

```
$ df -h  
/dev/xvda1 40G 39G 0G 100% /  
$ du -sh /var/log/* | sort -h  
5.1G /var/log/journal  
$ tail -n 10 /var/log/messages  
Error writing to disk: No space left on device
```

Root Cause: Excessive log rotation in /var/log/journal filled the root partition.

Resolution: Cleared old logs using `journalctl --vacuum-time=7d` and resized disk volume.

Prevention: Implement logrotate and set disk usage alerts for critical mounts.

Case 2: Inode Exhaustion on Application Partition

Problem: Disk usage appeared low but writes were failing with 'No space left on device'.

Investigation:

```
$ df -i  
/dev/xvdb1 1000000 1000000 0 100% /data  
$ find /data -type f | wc -l  
999999
```

Root Cause: Application created millions of small temporary files consuming all inodes.

Resolution: Removed unused temporary files and recreated filesystem with larger inode table.

Prevention: Add monitoring for inode utilization and cleanup cron jobs for temp files.

Case 3: LVM Snapshot Filled Up

Problem: Application I/O froze suddenly on production server.

Investigation:

```
$ lvs  
snap-lv vg0 -wi-ao---- 100%  
$ dmesg | tail  
device-mapper: snapshots: Snapshot is full
```

Root Cause: Snapshot volume reached 100% utilization causing writes to freeze.

Resolution: Removed stale snapshots and resized active ones using `lvextend`.

Prevention: Monitor LVM snapshot usage and automate cleanup.

Case 4: NFS Mount Hung After Remote Reboot

Problem: Web servers froze while accessing shared content from NFS mount.

Investigation:

```
$ df -h | grep nfs  
server:/data 500G 300G 200G 60% /mnt/nfs  
$ tail -f /var/log/messages  
nfs: server not responding, still trying
```

Root Cause: NFS mount was not configured with soft timeout and retransmission limits.

Resolution: Remounted with `soft,timeo=10,retrans=3` options.

Prevention: Configure NFS mounts in /etc/fstab with resilience options for HA.

Case 5: Filesystem Corruption Detected

Problem: System boot dropped to emergency mode after power outage.

Investigation:

```
$ fsck -n /dev/xvdb1
Inode 20345 has invalid mode.
$ dmesg | grep EXT4-fs
EXT4-fs error (device xvdb1): ext4_find_entry: reading directory
```

Root Cause: Unexpected power loss corrupted ext4 journal entries.

Resolution: Ran `fsck -y /dev/xvdb1` from recovery mode to repair filesystem.

Prevention: Deploy UPS protection and enable journaling on critical partitions.

Case 6: High Disk I/O Latency

Problem: Application response time degraded after storage migration.

Investigation:

```
$ iostat -x 1 3
avgqu-sz await svctm %util
32.0 210.3 12.0 99.8
```

Root Cause: Underlying block device had misaligned partitions causing latency.

Resolution: Repartitioned disks with proper alignment using parted (align-check optimal).

Prevention: Use cloud block storage benchmarks pre-deployment for performance validation.

Case 7: iSCSI Target Disconnects

Problem: Database node lost access to iSCSI LUN intermittently.

Investigation:

```
$ dmesg | grep iscsi
connection1:0: detected conn error (1011)
$ iscscliadm -m session
tcp: [1] 10.0.1.5:3260,1 iqn.2023-04.db01 (non-flash)
```

Root Cause: Unstable network path between initiator and target.

Resolution: Increased iSCSI timeout values and enabled multipath I/O.

Prevention: Implement iSCSI multipathing and network redundancy.

Case 8: RAID Array Degraded

Problem: Alerts received for degraded RAID1 array.

Investigation:

```
$ cat /proc/mdstat
md0 : active raid1 sda1[0] sdb1[1](F)
$ mdadm --detail /dev/md0
State : degraded
```

Root Cause: One disk in RAID array failed.

Resolution: Replaced faulty disk and rebuilt array using `mdadm --add /dev/sdb1`.

Prevention: Regularly test disk SMART data and maintain spares for quick replacement.

Case 9: Mount Options Causing Performance Drop

Problem: Read-heavy workload performed poorly after migration.

Investigation:

```
$ mount | grep data
```

```
/dev/xvdbl on /data type ext4 (rw,relatime,data=ordered)
$ iostat -xm 1
await values high during peak hours
```

Root Cause: Filesystem mounted with 'data=ordered' caused frequent journal flushes.

Resolution: Remounted with `data=writeback` for performance-critical workloads.

Prevention: Review mount options and use tuned profiles for workload optimization.

Case 10: Read-only Filesystem After Kernel Panic

Problem: System remounted root filesystem as read-only unexpectedly.

Investigation:

```
$ dmesg | grep EXT4-fs
EXT4-fs (xvda1): Remounting filesystem read-only
$ mount | grep '/'
/dev/xvda1 on / type ext4 (ro,errors=remount-ro)
```

Root Cause: Kernel detected I/O errors and remounted root filesystem as read-only.

Resolution: Performed disk checks, restored from backup, and rebooted cleanly.

Prevention: Enable proactive SMART monitoring and review hardware health logs.