LVM snapshots

Classic snapshots, thin snapshots, consistency and restore choices

What a snapshot captures

A snapshot provides a point-in-time block-device view of an LV. It is useful for short-term rollback, consistent backup workflows and testing changes. It is not a complete backup: snapshots normally depend on the same storage infrastructure as the origin, so a failed VG or disk can destroy both.

A block-level snapshot captures on-disk blocks, not necessarily a clean application transaction boundary. For databases and other stateful applications, coordinate with the application or filesystem freeze mechanism to obtain the required consistency level.

```
lvs -a -o lv_name,origin,segtype,lv_size,data_percent
```

Classic copy-on-write snapshots

Traditional thick LVM snapshots reserve a COW area at creation. As the origin changes, old blocks are preserved in that area so the snapshot continues to represent the earlier state. If the snapshot fills, it can become invalid. Its size must reflect expected changed-block volume and retention time, not simply the filesystem's used space.

The example creates a 10 GiB COW snapshot of a regular LV. Read it through its own block device; if mounting an ext4 snapshot, avoid conflicting filesystem identities in persistent mounts.

```
sudo lvcreate -s -L 10G -n documents-before vgdata/documents
lvs -a -o lv_name,origin,data_percent
sudo mkdir -p /mnt/documents-snapshot
sudo mount -o ro,noload /dev/vgdata/documents-before /mnt/documents-snapshot
```

Thin snapshots

Thin snapshots draw from the shared thin pool rather than a separately sized classic COW volume. This makes creating many snapshots convenient, but all snapshots compete for pool data and metadata. Heavy churn or long retention can consume substantial capacity. Thin snapshots have different merge and activation semantics from classic snapshots; consult the installed lvcreate and lvconvert manuals for the version in use.

```
sudo lvcreate -s -n vmdata-snap vgdata/vmdata
lvs -a -o lv_name,origin,pool_lv,data_percent,metadata_percent
```

Consistency and restoration

For a consistent snapshot, quiesce the application where possible. Filesystem freezing can prevent writes briefly while the snapshot is taken, but does not replace an application-specific checkpoint. Unfreeze promptly. For online databases, use the database vendor's backup mechanisms or documented snapshot coordination.

Restoring a classic snapshot can involve lvconvert --merge, often completing on the next activation if the origin is active. A merge discards changes made to the origin since the snapshot. Never issue it without confirming the target and keeping a separate recoverable backup.

```
sudo fsfreeze -f /srv/documents
```

```
# create snapshot promptly in a second session
sudo fsfreeze -u /srv/documents
# inspect before any merge:
lvs -a -o lv_name,origin,lv_attr
```

Lifecycle and monitoring

Snapshots should have explicit owners and expiry times. Monitor classic snapshot Data%, thin-pool Data% and Meta%, and available VG extents. Remove snapshots that are no longer needed; snapshot proliferation makes troubleshooting and capacity forecasting harder.

If a snapshot is being used as a backup source, verify that the resulting backup is complete and restorable before removing it. Snapshot creation alone is not evidence of a successful backup.

```
sudo lvremove vgdata/documents-before
lvs -a -o lv_name,origin,data_percent,metadata_percent
```

Reference scope: commands and defaults vary by distribution and software version. Consult the locally installed manual pages for exact syntax and behaviour.