KVM Backup online

Script to backup KVM VMs to backup location online

In order to move or copy the VM to a different server, you need to extract the qcow2 file to a disk location:

```
#!/bin/bash

# ----------- Configuration -----------
VMNAME="yourvmname"
[[ $# -gt 0 ]] && VMNAME="$1"

BACKUPDIR="/backups/kvm/${VMNAME}"
TIMESTAMP=$(date +%F%H-%M-%S)
SNAPSHOTPREFIX="backupsnapshot${TIMESTAMP}"
VMXMLFILE="${BACKUPDIR}/${VMNAME}config${TIMESTAMP}.xml"
declare -a CREATEDSNAPSHOTS
# -------------------------------------

set -euo pipefail

# Cleanup handler in case of error or interrupt
cleanuponexit() {
  echo "[!] Cleanup triggered due to error or interruption."
  for SNAPINFO in "${CREATEDSNAPSHOTS[@]}"; do
  IFS='¦' read -r DEVICE DISK <<< "$SNAPINFO"
  SNAPPATH="${DISK}.snap"
  if [[ -f "$SNAPPATH" ]]; then
  echo "      Removing leftover snapshot: $SNAPPATH"
  rm -f "$SNAPPATH"
  fi
  # Try to pivot back if snapshot is still active
  echo "      Attempting to blockcommit $DEVICE..."
  virsh blockcommit "$VMNAME" "$DEVICE" --active --pivot 2>/dev/null ¦¦ true
  done
  echo "[!] Cleanup complete. Exiting."
}
trap cleanuponexit ERR INT

echo "[*] Starting backup of VM: $VMNAME"
mkdir -p "$BACKUPDIR"

# Step 0: Save VM XML
echo "[0] Saving VM configuration..."
virsh dumpxml "$VMNAME" > "$VMXMLFILE"
echo "      Saved to $VMXMLFILE"

# Step 1: Get disks
mapfile -t DISKENTRIES < <(virsh domblklist "$VMNAME" --details ¦ awk '$2 == "disk" {print $3 "¦" $4}')
if [[ ${#DISKENTRIES[@]} -eq 0 ]]; then
  echo "[!] No disks found for VM: $VMNAME"
  exit 1
fi

# Step 2: Snapshot
echo "[1] Creating snapshots..."
for ENTRY in "${DISKENTRIES[@]}"; do
  IFS='¦' read -r DEVICE DISK <<< "$ENTRY"
  SNAPPATH="${DISK}.snap"
```

```
  if [[ -f "$SNAPPATH" ]]; then
  echo " ⚠ Snapshot file already exists: $SNAPPATH (removing)"
  rm -f "$SNAPPATH"
  fi

  virsh snapshot-create-as --domain "$VMNAME" \\
  --name "$SNAPSHOTPREFIX" \\
  --no-metadata \\
  --atomic \\
  --disk-only \\
  --diskspec "$DEVICE",snapshot=external,file="$SNAPPATH"

  CREATEDSNAPSHOTS+=("${DEVICE}¦${DISK}")
  echo "      Created snapshot for $DEVICE ($DISK → $SNAPPATH)"
done

# Step 3: Compress with zstd
echo "[2] Backing up disks with zstd..."
for ENTRY in "${DISKENTRIES[@]}"; do
  IFS='¦' read -r DEVICE DISK <<< "$ENTRY"
  DISKBASENAME=$(basename "$DISK")
  BACKUPFILE="${BACKUPDIR}/${VMNAME}${DISKBASENAME}${TIMESTAMP}.qcow2.zst"
  echo " → Compressing $DISK to $BACKUPFILE"
  zstd -T0 -19 -o "$BACKUPFILE" "$DISK"
done

# Step 4: Blockcommit snapshots
echo "[3] Merging snapshots..."
for ENTRY in "${DISKENTRIES[@]}"; do
  IFS='¦' read -r DEVICE DISK <<< "$ENTRY"
  virsh blockcommit "$VMNAME" "$DEVICE" --active --pivot
  echo "      Committed snapshot for $DEVICE"
done

# Step 5: Remove snapshot files
echo "[4] Cleaning up snapshot files..."
for ENTRY in "${DISKENTRIES[@]}"; do
  IFS='¦' read -r DEVICE DISK <<< "$ENTRY"
  SNAPPATH="${DISK}.snap"
  if [[ -f "$SNAPPATH" ]]; then
  rm -f "$SNAPPATH"
  echo "      Removed $SNAPPATH"
  fi
done

# Reset trap
trap - ERR INT
echo "[✓] Backup completed successfully for VM: $VMNAME"
```

zstd -d yourvmdisk.qcow2.zst -o yourvmdisk.qcow2

and import the vm definition with:

virsh define yourvmconfig.xml

σ υ ν η μ μ έ ν α::

Μοναδικό αναγνωριστικό λύσης: #1014
Συγγραφέας :: Kosmas
Τελευταία ενημέρωση: 2026-05-11 14:25