```bash
#!/bin/bash
              export PATH=$PATH:/usr/local/bin/:/usr/bin
              # Safety feature: exit script if error is returned, or if variables not set.
              # Exit if a pipeline results in an error.
              set -ue
              set -o pipefail
              ## Automatic EBS Volume Snapshot Creation & Clean-Up Script
              #
              # Written by Casey Labs Inc. (https://www.caseylabs.com)
              # Contact us for all your Amazon Web Services Consulting needs!
              # Script Github repo: https://github.com/CaseyLabs/aws-ec2-ebs-automatic-
              snapshot-bash
              #
              # Additonal credits: Log function by Alan Franzoni; Pre-req check by Colin
              Johnson
              #
              # PURPOSE: This Bash script can be used to take automatic snapshots of your
              Linux EC2 instance. Script process:
              # - Determine the instance ID of the EC2 server on which the script runs
              # - Gather a list of all volume IDs attached to that instance
              # - Take a snapshot of each attached volume
              # - The script will then delete all associated snapshots taken by the script
              that are older than 7 days
              #
              # DISCLAIMER: This script deletes snapshots (though only the ones that it
              creates).
              # Make sure that you understand how the script works. No responsibility
              accepted in event of accidental data loss.
              #
              ## Variable Declartions ##
              # Get Instance Details
              instance_id=$(wget -q -O- http://169.254.169.254/latest/meta-data/instance-
              id)
              region=$(wget -q -O- http://169.254.169.254/latest/meta-
              data/placement/availability-zone | sed -e 's/\([1-9]\).$/\1/g')
              # Set Logging Options
              logfile="/var/log/ebs-snapshot.log"
              logfile_max_lines="5000"
              # How many days do you wish to retain backups for? Default: 7 days
              retention_days="7"
              retention_date_in_seconds=$(date +%s --date "$retention_days days ago")
              ## Function Declarations ##
              # Function: Setup logfile and redirect stdout/stderr.
              log_setup() {
                  # Check if logfile exists and is writable.
                  ( [ -e "$logfile" ] || touch "$logfile" ) && [ ! -w "$logfile" ] && echo
```

```
"ERROR: Cannot write to $logfile. Check permissions or sudo access." && exit
1
    tmplog=$(tail -n $logfile_max_lines $logfile 2>/dev/null) && echo
"${tmplog}" > $logfile
    exec > >(tee -a $logfile)
    exec 2>&1
}
# Function: Log an event.
log() {
    echo "[$(date +"%Y-%m-%d"+"%T")]: $*"
}
# Function: Confirm that the AWS CLI and related tools are installed.
prerequisite_check() {
        for prerequisite in aws wget; do
                hash $prerequisite &> /dev/null
                if [[ $? == 1 ]]; then
                        echo "In order to use this script, the executable
\"$prerequisite\" must be installed." 1>&2; exit 70
                fi
        done
}
# Function: Snapshot all volumes attached to this instance.
snapshot_volumes() {
        for volume_id in $volume_list; do
                log "Volume ID is $volume_id"
                # Get the attched device name to add to the description so we
can easily tell which volume this is.
                device_name=$(aws ec2 describe-volumes --region $region --
output=text --volume-ids $volume_id --query
'Volumes[0].{Devices:Attachments[0].Device}')
                # Take a snapshot of the current volume, and capture the
resulting snapshot ID
                snapshot_description="$(hostname)-$device_name-backup-$(date
+%Y-%m-%d)"
                snapshot_id=$(aws ec2 create-snapshot --region $region --
output=text --description $snapshot_description --volume-id $volume_id --
query SnapshotId)
                log "New snapshot is $snapshot_id"

                # Add a "CreatedBy:AutomatedBackup" tag to the resulting
snapshot.
                # Why? Because we only want to purge snapshots taken by the
script later, and not delete snapshots manually taken.
                aws ec2 create-tags --region $region --resource $snapshot_id -
-tags Key=CreatedBy,Value=AutomatedBackup
        done
```

```
}
# Function: Cleanup all snapshots associated with this instance that are
older than $retention_days
cleanup_snapshots() {
        for volume_id in $volume_list; do
                snapshot_list=$(aws ec2 describe-snapshots --region $region --
output=text --filters "Name=volume-id,Values=$volume_id"
"Name=tag:CreatedBy,Values=AutomatedBackup" --query Snapshots[].SnapshotId)
                for snapshot in $snapshot_list; do
                        log "Checking $snapshot..."
                        # Check age of snapshot
                        snapshot_date=$(aws ec2 describe-snapshots --region
$region --output=text --snapshot-ids $snapshot --query Snapshots[].StartTime
| awk -F "T" '{printf "%s\n", $1}')
                        snapshot_date_in_seconds=$(date "--date=$snapshot_date"
+%s)
                        snapshot_description=$(aws ec2 describe-snapshots --
snapshot-id $snapshot --region $region --query Snapshots[].Description)
                        if (( $snapshot_date_in_seconds <=
$retention_date_in_seconds )); then
                                log "DELETING snapshot $snapshot. Description:
$snapshot_description ..."
                                aws ec2 delete-snapshot --region $region --
snapshot-id $snapshot
                        else
                                log "Not deleting snapshot $snapshot.
Description: $snapshot_description ..."
                        fi
                done
        done
}
## SCRIPT COMMANDS ##
log_setup
prerequisite_check
# Grab all volume IDs attached to this instance
volume_list=$(aws ec2 describe-volumes --region $region --filters
Name=attachment.instance-id,Values=$instance_id --query Volumes[].VolumeId --
output text)
snapshot_volumes
cleanup_snapshots
```