EBS Volume Management Documentation

Created: August 5, 2025 | Region: ap-southeast-1

Overview

This documentation covers EBS volume operations performed including instance creation, volume creation, XFS filesystem formatting, and attach/detach operations using both AWS Console and CLI.

EC2 Instances Created

Instance 1: i-017985543a46242c4

- **Type:** t2.micro
- **AZ:** ap-southeast-1a
- **Public DNS:** ec2-54-179-110-176.ap-southeast-1.compute.amazonaws.com
- **Root Volume:** vol-000bd9cefc53f0582 (8GB gp3)
- **Created:** 2025-08-05 13:25:25 UTC

Instance 2: i-0766077bd30eb8548

- **Type:** t2.micro
- **AZ:** ap-southeast-1a
- **Public DNS:** ec2-52-221-183-93.ap-southeast-1.compute.amazonaws.com
- **Root Volume:** vol-097b8e58596a5a7fb (8GB gp3)
- **Created:** 2025-08-05 13:24:21 UTC

EBS Volume Operations

```
 # 1. Volume Creation
aws ec2 create-volume \
    --size 10 \
    --volume-type gp3 \
    --availability-zone ap-southeast-1a \
    --tag-specifications 'ResourceType=volume,Tags=[{Key=Name,Value=test-volume}]'

# 2. Check Block Devices
lsblk

# Expected output:
# NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
# xvda    202:0    0   8G  0 disk
# └─xvda1 202:1    0   8G  0 part /
# xvdb    202:16   0  10G  0 disk    # <- New attached volume

# 3. Check Filesystem Type
sudo file -s /dev/xvdb

# Possible outputs:
# /dev/xvdb: data                     # <- No filesystem (raw)
# /dev/xvdb: SGI XFS filesystem       # <- XFS formatted
# /dev/xvdb: Linux rev 1.0 ext4       # <- ext4 formatted

#  XFS Filesystem Operations

# 1. Format Volume with XFS
sudo mkfs -t xfs /dev/xvdb

# Expected output:
# meta-data=/dev/xvdb              isize=512    agcount=4, agsize=655360 blks
#          =                       sectsz=512   attr=2, projid32bit=1
```

```
#          =                       crc=1        finobt=1, sparse=1, rmapbt=0
#          =                       reflink=1
# data     =                       bsize=4096   blocks=2621440, imaxpct=25
#          =                       sunit=0      swidth=0 blks
# naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
# log      =internal log           bsize=4096   blocks=2560, version=2
#          =                       sectsz=512   sunit=0 blks, lazy-count=1
# realtime =none                   extsz=4096   blocks=0, rtextents=0

# 2. Verify XFS Formatting
sudo file -s /dev/xvdb
# Output: /dev/xvdb: SGI XFS filesystem data

sudo xfs_info /dev/xvdb  # (only works if mounted)

# □ Mount Operations

# 1. Create Mount Point
sudo mkdir -p /mnt/test-mount

# 2. Mount Volume
sudo mount /dev/xvdb /mnt/test-mount

# Verify mount
df -h /mnt/test-mount
mount | grep test-mount

# 3. Fix Permissions
ls -ld /mnt/test-mount
sudo chown ec2-user:ec2-user /mnt/test-mount
sudo chmod 755 /mnt/test-mount

# 4. Test File Operations
echo "Hello XFS World" > /mnt/test-mount/testfile.txt
ls -la /mnt/test-mount/
cat /mnt/test-mount/testfile.txt

# 5. Unmount Volume
sudo umount /mnt/test-mount
df -h | grep test-mount  # Should return nothing

# □ AWS CLI Attach/Detach Operations

# 1. Attach Volume via CLI
aws ec2 attach-volume \
    --volume-id vol-xxxxxxxxx \
    --instance-id i-xxxxxxxxx \
    --device /dev/xvdb \
    --region ap-southeast-1

# Expected response:
# {
#     "AttachTime": "2025-08-05T13:30:00.000Z",
#     "Device": "/dev/xvdb",
#     "InstanceId": "i-xxxxxxxxx",
#     "State": "attaching",
#     "VolumeId": "vol-xxxxxxxxx"
# }

# 2. Check Attachment Status
aws ec2 describe-volumes \
    --volume-ids vol-xxxxxxxxx \
    --region ap-southeast-1 \
    --query 'Volumes[0].Attachments'
```

```
# 3. Detach Volume via CLI
aws ec2 detach-volume \
    --volume-id vol-xxxxxxxxx \
    --region ap-southeast-1

# Force detach (if needed)
aws ec2 detach-volume \
    --volume-id vol-xxxxxxxxx \
    --force \
    --region ap-southeast-1

# 🔄 Cross-Instance Volume Migration

# On Source Instance:
sudo umount /mnt/test-mount
aws ec2 detach-volume --volume-id vol-xxxxxxxxx

# Attach to Target Instance:
aws ec2 attach-volume \
    --volume-id vol-xxxxxxxxx \
    --instance-id i-target-instance \
    --device /dev/xvdb

# On Target Instance:
lsblk
sudo file -s /dev/xvdb
sudo mkdir -p /mnt/test-mount
sudo mount /dev/xvdb /mnt/test-mount
ls -la /mnt/test-mount/
cat /mnt/test-mount/testfile.txt

# 🖥 Console Operations

# Create Volume:
# EC2 Dashboard → Volumes → Create Volume
# Size: 10 GB
# Type: gp3
# AZ: ap-southeast-1a
# Tags: Name=test-volume

# Attach Volume:
# Select volume → Actions → Attach Volume
# Choose instance
# Device: /dev/xvdb

# Detach Volume:
# Select volume → Actions → Detach Volume
# Confirm detachment

# ⚠ Important Notes

# Best Practices:
# - Always unmount before detaching volumes
# - Check filesystem type before mounting
# - Fix permissions after mounting new volumes
# - Verify data integrity after cross-instance migration

# Common Issues:
# - Permission denied: Fix with chown and chmod
# - Device busy: Ensure no processes using the mount point
# - Mount fails: Check if filesystem exists with file -s

# XFS Specific:
# - Cannot shrink: XFS filesystems cannot be reduced in size
# - Can grow online: Use xfs_growfs to expand mounted XFS
# - Better performance: Generally faster than ext4 for large files
```

```
# 🔍 Troubleshooting Commands
sudo lsof +f -- /mnt/test-mount
sudo fuser -v /mnt/test-mount
sudo umount -f /mnt/test-mount
sudo umount -l /mnt/test-mount
aws ec2 describe-volumes --volume-ids vol-xxxxxxxxx
watch -n 2 'lsblk'
```