**# PART 1 - EXTEND ROOT VOLUME**

# Launch an Amazon Linux 2 instance with default ebs volume and ssh to it.

# List block devices (lsblk) and file system disk space usage (df) of the instance.

# Only root volume should be listed with the default capacity.

**lsblk**  #List block devices

**df -h**  #file system disk space usage

**sudo file -s /dev/xvda1**  #Check file system of the root volume's partition.

**sudo growpart /dev/xvda 1** #Extend partition 1 on the modified volume and occupy all newly #

**PART 2 - CREATE NEW EBS VOLUME, ATTACH IT AND MODIFY**

# Create a new volume "in the SAME AZ with the INSTANCE" (2 GB for this demo).

# Attach the new volume to the instance and then list block storages again.

# Root volume and the newly created second volumes should be listed.avaiable space

**lsblk**

**df -h**

**sudo file -s /dev/xvdf** #Check if the attached volume is already formatted or not. It should be NOT FORMATTED.

**sudo mkfs -t ext4 /dev/xvdf** #Format the new volüme

**sudo file -s /dev/xvdf** #Check the format of the volume again after formatting.

**sudo mkdir /mnt/2nd-vol** #Mount the new volume to the mounting point path.

**lsblk** #Check if the attached volume is mounted to the mounting point path.

**df -h** #Show the available space, on the mounting point path.

**ls -lh /mnt/2nd-vol/** #Check if there is data on it or not.

**cd /mnt/2nd-vol**

**sudo touch file.txt** #Create a new file to show persistence in later steps.

**ls**

# Modify the new volume in aws console, and enlarge capacity to double its size (2GB >> 4GB).

# Check if the attached volume shows the new capacity.

**lsblk**

**df -h** #Show the real capacity used currently at mounting path, old capacity should be displayed.

**sudo resize2fs /dev/xvdf** #Resize the EXT4 file system on the new volume to cover all available space.

**df -h** #Show the real capacity used currently at mounting path, new capacity should reflect the modified volume size.

**ls -lh /mnt/2nd-vol/** #Show that the data still persists on the newly enlarged volume.

**sudo reboot now** #Show that mounting point path is gone when instance reboots.

**lsblk** #Show the new volume is still attached, but not mounted.

**sudo file -s /dev/xvdf** #Check if the attached volume is formatted or not.

**sudo mount /dev/xvdf** /mnt/2nd-vol/ #Mount the new volume to the mounting point path again.

# show the used and available capacity is same as the disk size.

**lsblk**

**df -h**

**ls -lh /mnt/2nd-vol/** # if there is data on it, check if the data still persists.

# PART 3 - CREATE NEW EBS VOLUME (AND PARTITION), ATTACH IT AND MODIFY

# List volumes to show current status, Root and second volumes should be listed.

**Lsblk**

**df -h**

# Create third volume (4 GB for this demo) in the same AZ with the instance.

# Attach the new volume and list volumes again.

# Root, second and third volumes should be listed

**lsblk**

**df -h**

# Show the current partitions ("fdisk -l /dev/xvda" for specific partition).

**sudo fdisk -l**

# Check all available fdisk commands and using "m".

**sudo fdisk /dev/xvdg**

# n -> add new partition (with 2G size)

# p -> primary

# Partition number: 1

# First sector: default - use Enter to select default

# Last sector: +2g (you can also type sector)

# n -> add new partition (with rest of the size-2G)

# p -> primary

# Partition number: 2

# First sector: default - use Enter to select default

# Last sector: default - use Enter to select default

# w -> write partition table

# Show new partitions

**lsblk**

# Check if the newly created partitons are formatted or not. They should be NOT FORMATTED.

**sudo file -s /dev/xvdg1**

**sudo file -s /dev/xvdg2**

# Format the new partitions.

**sudo mkfs -t ext4 /dev/xvdg1**

**sudo mkfs -t ext4 /dev/xvdg2**

# Create mounting point paths for the new volume.

**sudo mkdir /mnt/3rd-vol-part1**

**sudo mkdir /mnt/3rd-vol-part2**

# Mount the partitions to the mounting point paths.

**sudo mount /dev/xvdg1 /mnt/3rd-vol-part1/**

**sudo mount /dev/xvdg2 /mnt/3rd-vol-part2/**

# List volumes to show current status, all volumes and partittions should be listed.

**lsblk**

# Show the used and available capacities related with volumes and partitions.

**df -h**

# Create a new file to show persistence in later steps.

**sudo touch /mnt/3rd-vol-part2/guilewashere2.txt**

**ls -lh /mnt/3rd-vol-part2/**

# Modify the new (3rd) volume, and enlarge capacity to double its size (4GB >> 8GB).

# Check if the attached volume is showing the new capacity.

**lsblk**

# Show the real capacity used currently at mounting path, old capacity should be shown.

**df -h**

# Extend the partition 2 and occupy all newly avaiable space.

**sudo growpart /dev/xvdg 2**

# ​Show the real capacity used currently at mounting path, updated capacity should be shown.

lsblk

# Resize and extend the file system.

**sudo resize2fs /dev/xvdg2**

# show the newly created file to show persistence

**ls -lh /mnt/3rd-vol-part2/**

# reboot and show that configuration is gone

**sudo reboot now**

PART 4 - AUTOMOUNT EBS VOLUMES AND PARTITIONS ON REBOOT

# Back up the /etc/fstab file.

**sudo cp /etc/fstab /etc/fstab.bak**

# Open /etc/fstab file and add the following info to the existing.(UUID's can also be used)

**sudo nano /etc/fstab # sudo vim /etc/fstab >>> for vim**

/dev/xvdf /mnt/2nd-vol ext4 defaults,nofail 0 0

/dev/xvdg1 /mnt/3rd-vol-part1 ext4 defaults,nofail 0 0

/dev/xvdg2 /mnt/3rd-vol-part2 ext4 defaults,nofail 0 0

# more info for fstab >> https://wiki.debian.org/fstab

# Reboot and show that configuration exists (NOTE)

**sudo reboot now**

# List volumes to show current status, all volumes and partittions should be listed

**lsblk**

# Show the used and available capacities related with volumes and partitions

**df -h**

**# Check if the data still persists.**

**ls -lh /mnt/2nd-vol/**

**ls -lh /mnt/3rd-vol-part2/**

# NOTE: You can use "sudo mount -a" to mount volumes and partitions after editing fstab file without rebooting.

# https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/recognize-expanded-volume-linux.html

# https://www.tecmint.com/fdisk-commands-to-manage-linux-disk-partitions/