

Changing Termination behaviour of instances

To change termination settings, highlight instance.

→ Instances

- Actions
 - instance settings
 - ⇒ change termination protection
 - check enables
 - save

Now you hit terminate it was not terminated.

- To terminate highlight
 - same untick enable
 - save it.

Volumes

→ Volumes

We need create volume where server is present Because we need to attach volume to server.

→ create volume

- select volume type (General purpose SSD (gp2))
- size (10GB)
- create volume.
- name it
- Attach to instance

→ click on volume

- Actions
- Attach volume
 - select instance in Instance
 - Attach volume.
 - Now its status change into in use

→ Gitbash

- Sudo su - root
- lsblk

(xvda is a Root Volume)
(xvdat is a attached volume)

File system - in volume

3rd Session of aws 32/1/24

file system is nothing but how you store data and How retrieve data

⇒ **Creating volume**

→ **volumes**

- create volume
- General purpose (gp2) (for root use only gp2, gp3, io2, io3 only)
- size (896)
- IOPS (3000) (input output per second) (Numbers of reads/writes per second)
- Availability zone Cross server zone
- tag (name)
- ⇒ create volume

→ **Attach volume.**

- Action
- Attach volume
- ⇒ instance (select which instance do you want)
- Attach volume.
- once it was attached it shows in use

→ **Gitbash**

→ how many volumes do you want to find

lsblk

- in the terminal
- List of block devices

xvda is xvdt is a root volume. In GUI it was sda1
xvdf extra volume we attached

- it in Linux if you want use as a mount volume
- we need to format it with Ext3 or ext4
- sudo su-root
- command mkfs -t exty dev/xvdf

2) To show it created or not

blkid /dev/xlf

3) if you want to use this volume

- we need to mount to an directory.
- mkdir mydir
- mounting to an directory
- Mount /dev/xvdt mydir

4) to see mounted or not

lsblk

(or)

5) this shows only mounted Volumes

df -h

6) create some files in volume

- cd mydir
- touch f1 f2 f3 f4

➔ Now data will not present in mydir it will show only
⇒ actually, it presents in volume.

8) to unmount this volume from server

umount /dev/xvdf

9) check unmount or not

df -h

check files

➔ cd mydir

Detach volume from instance in Aws.

➔ check lsblk in gitbash

Now attach this volume into another server for checking files are present not.

➔ **Instances**

- ➔ create instance.
- ➔ security group select pre-existing
- ➔ subnet (select your volume location)
- ➔ create instance

➔ **Gitbash**

➔ lsblk

3) attaching volume to this instance.

➔ **volumes**

- ➔ actions
- ➔ attach volumes

➔ **Gitbash**

4) check its mounted or not

lsblk

➔ do you think we will format again (No)

5) Just create an directory.

Mkdir dir.

6) check formatted or not

blkid /dev/xvdf

7) Now mount to dir2 directory then only we add files etc...

mount /dev/sxvat dire2

Check: - df -h

8) check files present or not from old instance.

➔ Cd dir2

➔ ls -ltr

Snapshots

- ➔ snapshot is a pointing time backup.
- ➔ if your volume is corrupted then how we will recover data then we use snapshot
- ➔ snapshot is A pointing time backup of volume
- ➔ we can't attach volume directly to any EC2 instance
- ➔ we need to create a volume out of this snapshot
- ➔ Here snapshots have files only we can before you snapped

Creating snapshot

- ➔ take a volume
 - Actions
 - create snapshot
 - ⇒ Description (write what do you want)
 - Add tag (Game)
 - create snapshot

Snapshots we create any zone.

Then if you want detach Volume from instance

Let's detach it

⇒ Gitbash

- unmount /dev/xvdf
- lsblk (Check unmount or not)

⇒ Detaching volume in volume section.

- Actions
- detach volume

Volumes

→ Volumes

- We need create volume where server is present Because we need to attach volume to server.
- With EBS volumes, you can create, attach, detach, and delete volumes as needed. These volumes can be formatted with a file system and used as a storage device for databases, application data, or file storage.

→ create volume

- select volume type (General purpose SSD (gp2))
- size (10GB)
- create volume.
- name it
- Attach to instance

→ click on volume

- Actions
- Attach volume
 - select instance in Instance
 - Attach volume.
 - Now its status change into in use

→ Gitbash

- Sudo su - root (xvda is a Root Volume)
- lsblk (xvdat is a attached volume)

Creating volume from snapshot

→ Snapshots

- Actions
- create volume from, snapshot
- Availability zone (instance zone)
- Create volume.

Attach volume to instance

→ volumes

- Actions
- Attach volume.
- instance (select another)
- Attach volume

→ Gitbash

- sudo su - root
- lsblk
- mount /dev/xvdat dir
- cd dir
- ls

Copying snapshot

- ➔ Copy Snapshot one region to another region
 - ➔ In present region instance take a snapshot of server
- ➔ **snapshots**
 - ➔ Actions
 - ➔ copy snapshot
 - (it takes to copy snapshot section)*
 - ➔ Destination Region (select what do you want),
 - ⇒ copy snapshot
 - Now switch in to Ohio Region (what do you want)**
- ➔ **snapshot**
 - ➔ Actions
 - ➔ create volume from snapshot
 - ➔ 8gb
 - ➔ Availability zone (us East)
 - ➔ Create volume.
- ➔ **instances**
 - ➔ New instance
 - ➔ ubuntu
 - ➔ Networking settings
 - ➔ subnet
 - ➔ us East 2a
 - ➔ Launch instance
- ➔ **volumes**
 - ➔ Actions
 - ➔ Attach volume
 - ➔ instance (select which is you want it)
 - ➔ Attach volume
- ➔ **Git bash**
 - ➔ Lsblk
 - ➔ sudo su-root
 - ➔ Mkdir d1
 - ➔ mount /dev/xudt d1
 - ➔ Cd d1
 - ➔ Ls-altr