



Discipline

Configure a RDS with Postgres SQL and create 2 table in it.

CS5002

Activity-13

Sheikh Muhammed Tadeeb (AU19B1014)

❖ Problem Statement:

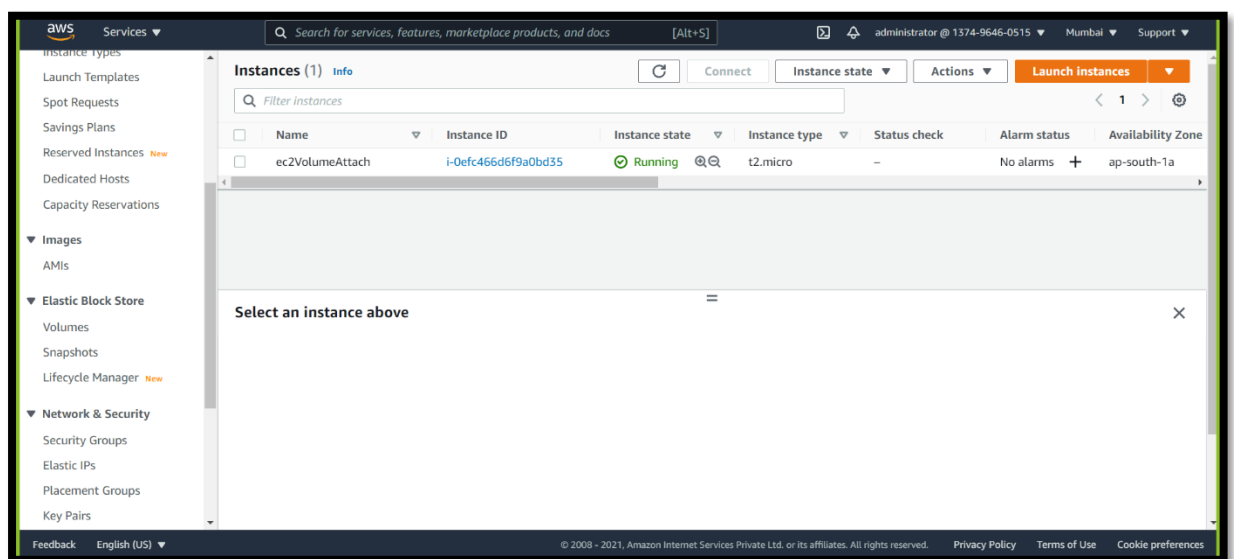
Create a Ec2 instance and attached a EBS Volume of 50 GB Size.

Mount that Volume on Linux machine.

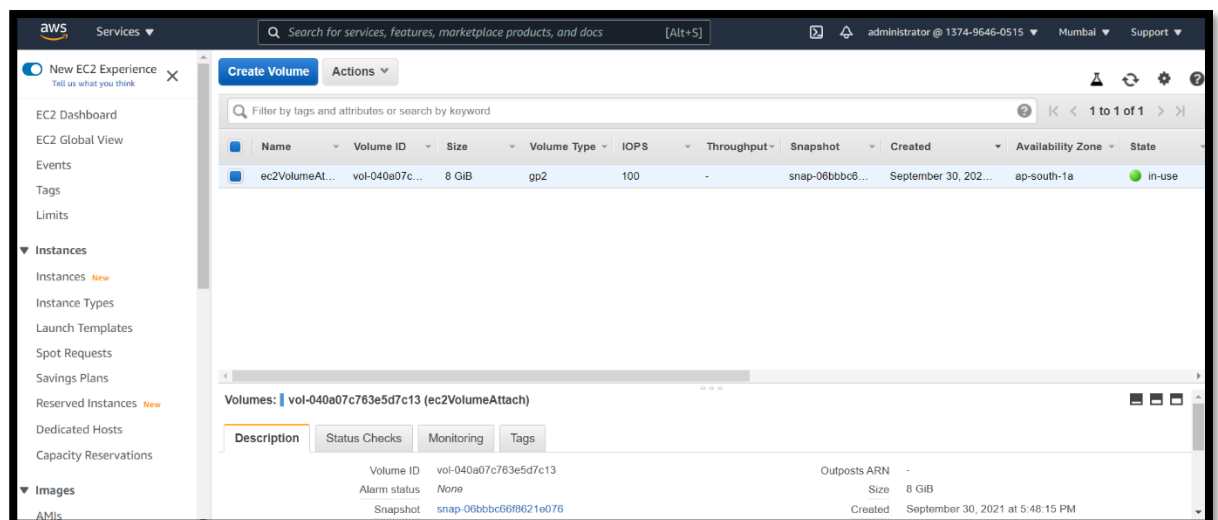
Provide the out of the Volume mounted on your Linux machine using df -h command.

❖ Solution:

Step 1: We created an instance named Ec2-volume-attach for attaching volume.



Step 2: We are creating volume with general purpose SSD (gp3) type and size of 50 GB.



aws Services [Alt+S] administrator @ 1374-9646-0515 Mumbai Support

Volumes > Create Volume

Create Volume

Volume Type: General Purpose SSD (gp3) ⓘ

Size (GiB): 50 (Min: 1 GiB, Max: 16384 GiB) ⓘ

IOPS: 3000 (Min: 3000 IOPS, Max: 16000 IOPS) ⓘ

Throughput (MB/s): 125 (Min: 125 MB/s, Max: 1000 MB/s) ⓘ

Availability Zone*: ap-south-1a ⓘ

Snapshot ID: Select a snapshot ⓘ

Encryption: ☐ Encrypt this volume

Key (128 characters maximum) Value (256 characters maximum)

This resource currently has no tags

Choose the Add tag button or [click to add a Name tag](#)

Add Tag 50 remaining (Up to 50 tags maximum)

* Required Cancel Create Volume

Feedback English (US) © 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

aws Services [Alt+S] administrator @ 1374-9646-0515 Mumbai Support

Volumes > Create Volume

Create Volume

✓ Volume created successfully

Volume ID: vol-0b5ba11cc13292059

Close

Note: We can see the volume created named ec2VolumeAttach with the size of 50 GB.

Step 4: In this step, we are attaching a volume to the instance we have created for it and selecting the same region as the instance in volume. After that volume was attached successfully.

Attach Volume

Volume ⓘ vol-0b5ba11cc13292059 in ap-south-1a

Instance ⓘ in ap-south-1a

Device ⓘ
Linux Devices: /dev/sdf through /dev/sdp

Note: Newer Linux kernels may rename your devices to /dev/xvdf through /dev/xvdp internally, even when the device name entered here (and shown in the details) is /dev/sdf through /dev/sdp.

Cancel Attach

Step 5: we have successfully attached a volume but still, it is not available to use for any storage so we have opened an instance on putty and checked the partition through **df -h** and saw it is not available.

```
ec2-user@ip-172-31-45-112:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
  
  _|_  _|_  )  
  _|_ ( _|_ /  Amazon Linux 2 AMI  
  _|_ \ _|_  _|_  
  
https://aws.amazon.com/amazon-linux-2/  
11 package(s) needed for security, out of 35 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-45-112 ~]$ df -h  
Filesystem      Size  Used Avail Use% Mounted on  
devtmpfs        482M   0  482M   0% /dev  
tmpfs           492M   0  492M   0% /dev/shm  
tmpfs           492M 404K  492M   1% /run  
tmpfs           492M   0  492M   0% /sys/fs/cgroup  
/dev/xvda1      8.0G  1.5G   6.6G  19% /  
tmpfs           99M   0   99M   0% /run/user/1000  
[ec2-user@ip-172-31-45-112 ~]$
```

Step 6: We have to mount it for use so we will check for partition whether it has data or not. So, we will use **file -s/dev/xvdf** command to check it and we found that the disk is empty.

```
ec2-user@ip-172-31-45-112:~  
tmpfs          99M   0   99M   0% /run/user/1000  
[ec2-user@ip-172-31-45-112 ~]$  
[ec2-user@ip-172-31-45-112 ~]$ lsblk  
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT  
xvda        202:0    0   8G  0 disk  
└─xvda1     202:1    0   8G  0 part /  
xvdf        202:80   0  50G  0 disk  
[ec2-user@ip-172-31-45-112 ~]$ lsblk -f  
NAME        FSTYPE LABEL UUID                                MOUNTPOINT  
xvda  
└─xvda1     xfs     /      55ee5a5f-d155-47e0-9121-e6f4522cb2bf /  
xvdf  
[ec2-user@ip-172-31-45-112 ~]$ sudo file -s/dev/xvdf  
file: invalid option -- '/'  
Usage: file [-bchikLlNnprsvz0] [--apple] [--mime-encoding] [--mime-type]  
        [-e testname] [-F separator] [-f namefile] [-m magicfiles] file ...  
        file -C [-m magicfiles]  
        file [--help]
```

Step 7: Now, we are formatting the partition using **mkfs** (make file system) format and extension as **ext4** to the partition.

```
PutTY (inactive)
xvdf
[ec2-user@ip-172-31-45-112 ~]$ sudo file -s/dev/xvdf
file: invalid option -- '/'
Usage: file [-bchikLlNnprsvz0] [--apple] [--mime-encoding] [--mime-type]
        [-e testname] [-F separator] [-f namefile] [-m magicfiles] file ...
        file -C [-m magicfiles]
        file [--help]
[ec2-user@ip-172-31-45-112 ~]$ sudo mkfs -t ext4 /dev/xvdf
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
3276800 inodes, 13107200 blocks
655360 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2162163712
400 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

Step 8: Now, we have to create a directory to mount the partition because the partition can be mounted only on the directory, so we mounted it using **mount /dev/xvdf /directory name** command.

```
[ec2-user@ip-172-31-45-112 ~]$ sudo mkdir /volume01
[ec2-user@ip-172-31-45-112 ~]$ sudo mount /dev/xvdf /volume01
[ec2-user@ip-172-31-45-112 ~]$ cd /volume01/
```

Step 9: So, we have mounted the partition successfully and it is ready for use and store data, we can check that using **df -h** command in the partition list and it is mounted on the directory we created.

```
[ec2-user@ip-172-31-45-112 ~]$ sudo mkdir /volume01
[ec2-user@ip-172-31-45-112 ~]$ sudo mount /dev/xvdf /volume01
[ec2-user@ip-172-31-45-112 ~]$ cd /volume01/
[ec2-user@ip-172-31-45-112 volume01]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        482M   0    482M   0% /dev
tmpfs           492M   0    492M   0% /dev/shm
tmpfs           492M 404K   492M   1% /run
tmpfs           492M   0    492M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  1.5G   6.6G  19% /
tmpfs           99M   0     99M   0% /run/user/1000
/dev/xvdf       50G   53M   47G   1% /volume01
[ec2-user@ip-172-31-45-112 volume01]$
```

Step 10: In Below snapshot we can see our attached volume is available in storage.

The screenshot displays the AWS Management Console interface. The left sidebar contains navigation options such as 'New EC2 Experience', 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity Reservations', 'Images', 'AMIs', and 'Elastic Block Store'. The main content area is titled 'Instances (1/1)' and shows a table with one instance, 'ec2VolumeAttach', with ID 'i-0efc466d6f9a0bd35', in a 'Running' state. Below this, a detailed view for the instance shows its root device name as '/dev/xvda' and type as 'EBS'. A section for 'Block devices' contains a table listing two attached volumes:

Volume ID	Device name	Volume size (GiB)	Attachment status	Attachment time	Encrypted	KMS key ID
vol-040a07c763e5d7c13	/dev/xvda	8	Attached	Thu Sep 30 2021 17:48:15 ...	No	-
vol-0b5ba11cc13292059	/dev/sdf	50	Attached	Thu Sep 30 2021 17:50:28 ...	No	-

The footer of the console shows the copyright notice '© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.' along with links for 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.