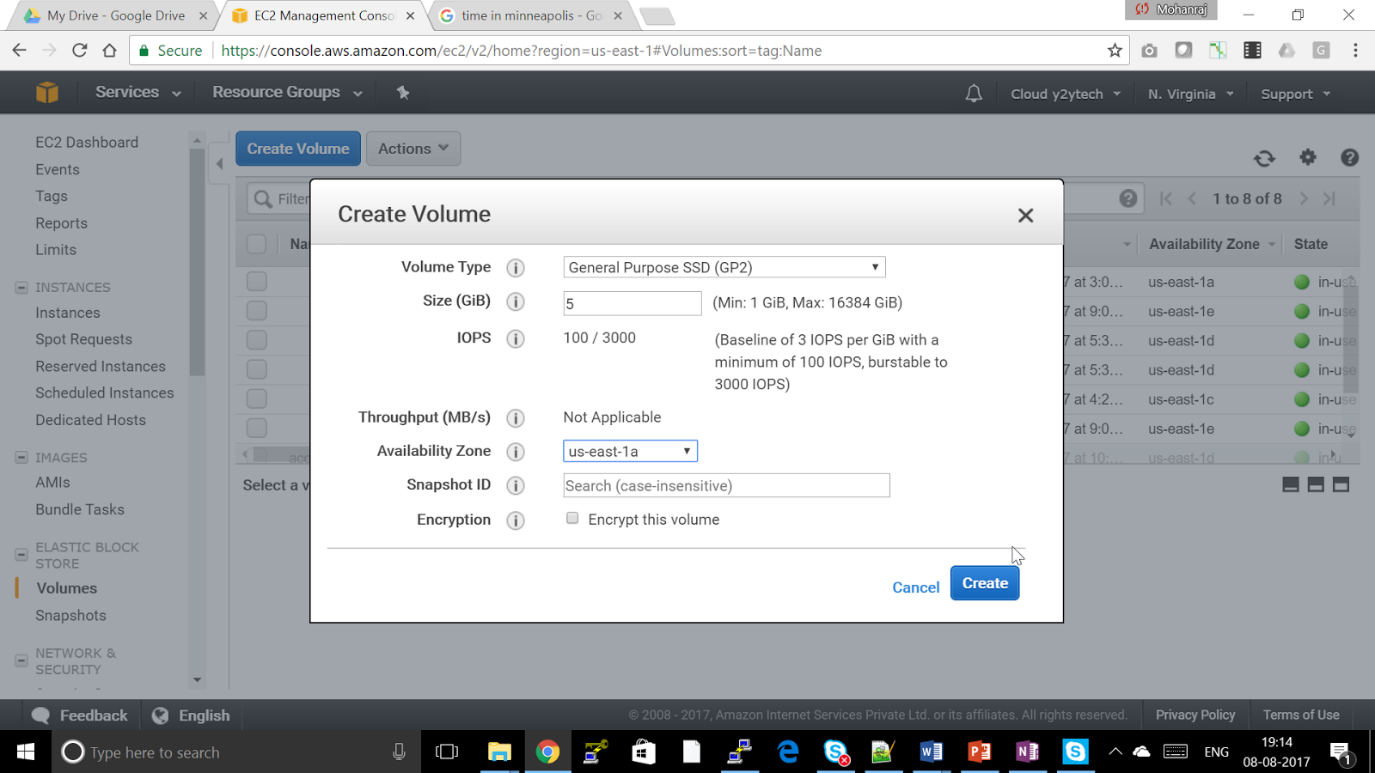
**To attach an EBS volume to an instance using the console**

1. Open the Amazon EC2 console.
2. Click **Volumes** in the navigation pane.
3. Click on Create Volume

* Give Volume Type ( General Purpose SSD, Provisioned IOPS, Magnetic)
* Volume Size in GB
* IOPS if you choose Provisioned IOPS volume Type
* Availability Zone
* If the Volume needs to be created from Snapshot give Snapshot ID
* If you need Encrypt the data at Rest.

Click Create to create the volume



1. Select a volume and then click **Attach Volume**.
2. In the **Attach Volume** dialog box, start typing the name or ID of the instance to attach the volume to in the **Instance** box, and select it from the list of suggestion options (only instances that are in the same Availability Zone as the volume are displayed).
3. You can keep the suggested device name, or enter a different supported device name.

**Important**

The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different from the name that Amazon EC2 recommends.

1. Click **Attach**.

**To make an EBS volume available for use on Linux**

1. Connect to your instance using SSH.
2. Depending on the block device driver of the kernel, the device might be attached with a different name than what you specify. For example, if you specify a device name of/dev/sdh, your device might be renamed /dev/xvdh
3. Use the **lsblk** command to view your available disk devices and their mount points (if applicable) to help you determine the correct device name to use.

[ec2-user ~]$ **lsblk**

NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT

xvdf  202:80   0  100G  0 disk

xvda1 202:1    0    8G  0 disk /

The output of **lsblk** removes the /dev/ prefix from full device paths. In this example,/dev/xvda1 is mounted as the root device (note the MOUNTPOINT is listed as /, the root of the Linux file system hierarchy), and /dev/xvdf is attached, but it has not been mounted yet.

1. Determine whether you need to create a file system on the volume. New volumes are raw block devices, and you need to create a file system on them before you can mount and use them. Volumes that have been restored from snapshots likely have a file system on them already; if you create a new file system on top of an existing file system, the operation overwrites your data. Use the **sudo file -s *device*** command to list special information, such as file system type.
2. [ec2-user ~]$ **sudo file -s /dev/xvdf**

/dev/xvdf: data

If the output of the previous command shows simply data for the device, then there is no file system on the device and you need to create one. You can go on to [Step 4](https://www.google.com/url?q=http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-using-volumes.html%23create_file_system_step&sa=D&ust=1502885731053000&usg=AFQjCNGlbynjf66TRl49tVJEKbmNXa6l9g). If you run this command on a device that contains a file system, then your output will be different.

[ec2-user ~]$ **sudo file -s /dev/xvda1**

/dev/xvda1: Linux rev 1.0 ext4 filesystem data, UUID=1701d228-e1bd-4094-a14c-8c64d6819362 (needs journal recovery) (extents) (large files) (huge files)

In the previous example, the device contains Linux rev 1.0 ext4 filesystem data, so this volume does not need a file system created (you can skip [Step 4](https://www.google.com/url?q=http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-using-volumes.html%23create_file_system_step&sa=D&ust=1502885731054000&usg=AFQjCNFrsI7SohzmevJmPEBpGLr891bmjA) if your output shows file system data).

1. Create Physical Volume, Volume Group and LVM

$ pvcreate /dev/xvdf

$ vgcreate datavg /dev/xvdf

$ lvcreate -n datavol -L +<size in GB>G datavg

1. Use the following command to create an ext4 file system on the volume. Substitute the device name (such as /dev/xvdf) for *device\_name*. Depending on the requirements of your application or the limitations of your operating system, you can choose a different file system type, such as ext3 or XFS.

[ec2-user ~]$ **sudo mkfs -t xfs */dev/datavg/datavol***

1. Use the following command to create a mount point directory for the volume. The mount point is where the volume is located in the file system tree and where you read and write files to after you mount the volume. Substitute a location for *mount\_point*, such as /data.

[ec2-user ~]$ **sudo mkdir *mount\_point***

1. Use the following command to mount the volume at the location you just created.

[ec2-user ~]$ **sudo mount *device\_name* *mount\_point***

1. (Optional) To mount this EBS volume on every system reboot, add an entry for the device to the /etc/fstab file.
2. Create a backup of your /etc/fstab file that you can use if you accidentally destroy or delete this file while you are editing it.

[ec2-user ~]$ **sudo cp /etc/fstab /etc/fstab.orig**

1. Open the /etc/fstab file using any text editor, such as **nano** or **vim**.

**Note**

You need to open the file as root or by using the **sudo** command.

1. Add a new line to the end of the file for your volume using the following format.

*device\_name*  *mount\_point*  *file\_system\_type*  *fs\_mntops*  *fs\_freq*  *fs\_passno*

The last three fields on this line are the file system mount options, the dump frequency of the file system, and the order of file system checks done at boot time. If you don't know what these values should be, then use the values in the example below for them (defaults,nofail 0 2). For more information on/etc/fstab entries, see the **fstab** manual page (by entering **man fstab** on the command line). For example, to mount the ext4 file system on the device/dev/xvdf at the mount point /data, add the following entry to /etc/fstab.

**Note**

If you ever intend to boot your instance without this volume attached (for example, so this volume could move back and forth between different instances), you should add the nofail mount option that allows the instance to boot even if there are errors in mounting the volume. Debian derivatives, such as Ubuntu, must also add thenobootwait mount option.

/dev/xvdf       /data   ext4    defaults,nofail        0       2

1. After you've added the new entry to /etc/fstab, you need to check that your entry works. Run the **sudo mount -a** command to mount all file systems in/etc/fstab.

[ec2-user ~]$ **sudo mount -a**

If the previous command does not produce an error, then your /etc/fstab file is OK and your file system will mount automatically at the next boot. If the command does produce any errors, examine the errors and try to correct your/etc/fstab.

**Warning**

Errors in the /etc/fstab file can render a system unbootable. Do not shut down a system that has errors in the /etc/fstab file.