|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | AWS Lab 4 | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | Weizhen Chen |  | | | |
|  | | | | —CCNP—Jeffery Mason &Michael Hansen |  | | | |
|  | | |  | | |  | | |

# **Lab 4: Working with EBS**

**Amazon Elastic Block Store (Amazon EBS)**

Amazon EBS volumes are network-attached and persist independently from the life of an instance. They are highly available, highly reliable volumes that can be leveraged as an Amazon EC2 instances boot partition or attached to a running Amazon EC2. With boot partition, Amazon EC2 instances can be stopped and subsequently restarted, where you only pay for the storage resources used while maintaining your instance's state. It also offers greatly improved durability over local Amazon EC2 instance stores because Amazon EBS volumes are automatically replicated on the backend. Amazon EBS provides the ability to create point-in-time consistent snapshots of your volumes that are then stored in Amazon Simple Storage Service (Amazon S3) and automatically replicated across multiple Availability Zones.

**Amazon EBS Features**

* **Persistent storage:** Volume lifetime is independent of any particular Amazon EC2 instance.
* **General purpose:** Amazon EBS volumes are raw, unformatted block devices that can be used from any operating system.
* **High performance:** Amazon EBS volumes are equal to or better than local Amazon EC2 drives.
* **High reliability:** Amazon EBS volumes have built-in redundancy within an Availability Zone.
* **Designed for resiliency:** The AFR (Annual Failure Rate) of Amazon EBS is between 0.1% and 1%.
* **Variable size:** Volume sizes range from 1 GB to 16 TB.
* **Easy to use:** Amazon EBS volumes can be easily created, attached, backed up, restored, and deleted.

**The purpose**

To learn how to create an Amazon EBS volume, attach it to an instance, apply a file system to the volume, and then take a snapshot backup. By the end you will be able create an Amazon EBS volume, attach and mount your volume to an EC2 instance, create a snapshot of your volume, create a new volume from your snapshot, attach and mount the new volume to your EC2 instance

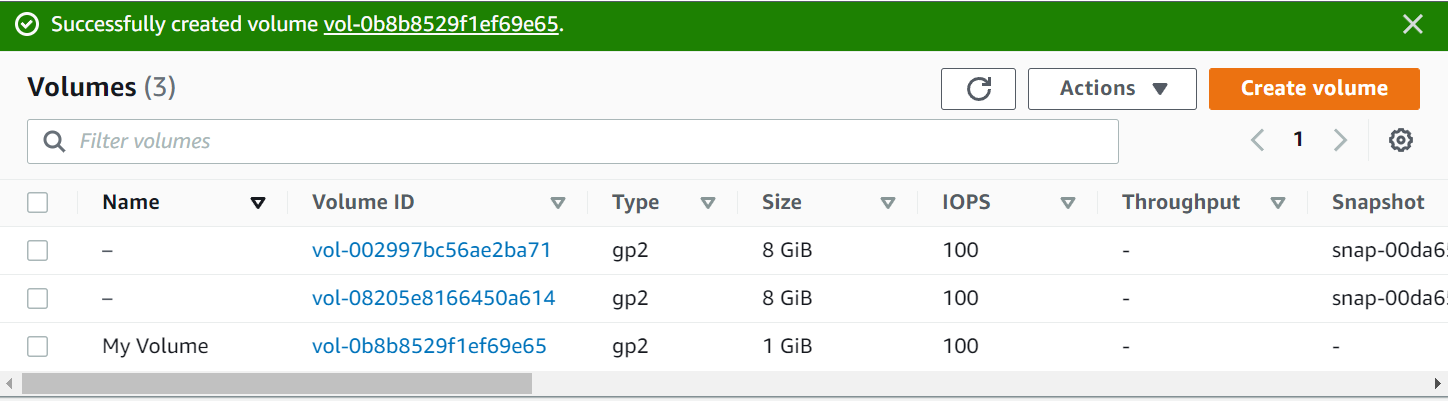
**Task 1: Create a New EBS Volume**

1. In the **AWS Management Console**, on the **Services** menu, click **EC2**.
2. In the navigation pane on the left, choose Instances

The instance name lab should be launched and have the Availability Zone of *us-east-1a*

1. Choose Create Volume The configure
   * **Volume Type:** *General Purpose SSD (gp2)*
   * **Size (GiB):** 1. **NOTE**: You may be restricted from creating large volumes.
   * **Availability Zone:** Select the same availability zone as your EC2 instance.
   * Choose **Add Tag**
   * In the Tag Editor, enter:
     1. **Key:** Name
     2. **Value:** My Volume

* Then **Create Volume**

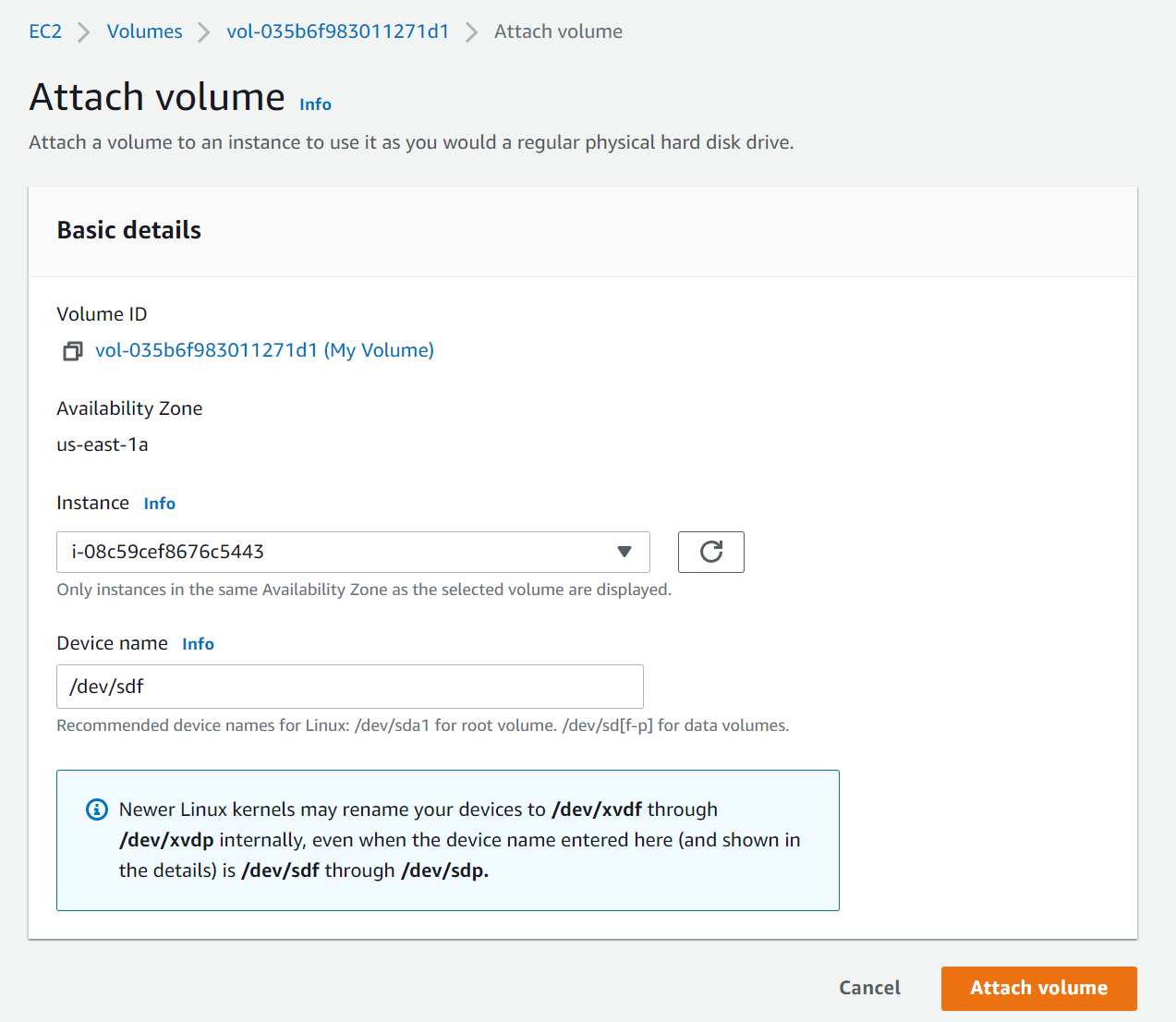


## **Task 2: Attach the Volume to an Instance**

1. Select the Volume created (**My Volume**)
2. Click on the **Actions** tab and choose **Attach volume**
3. Choose the **Instance** field that is labeled lab

The **Device field** should be /dev/sdf

1. Choose Attach volume, the state is now in-use

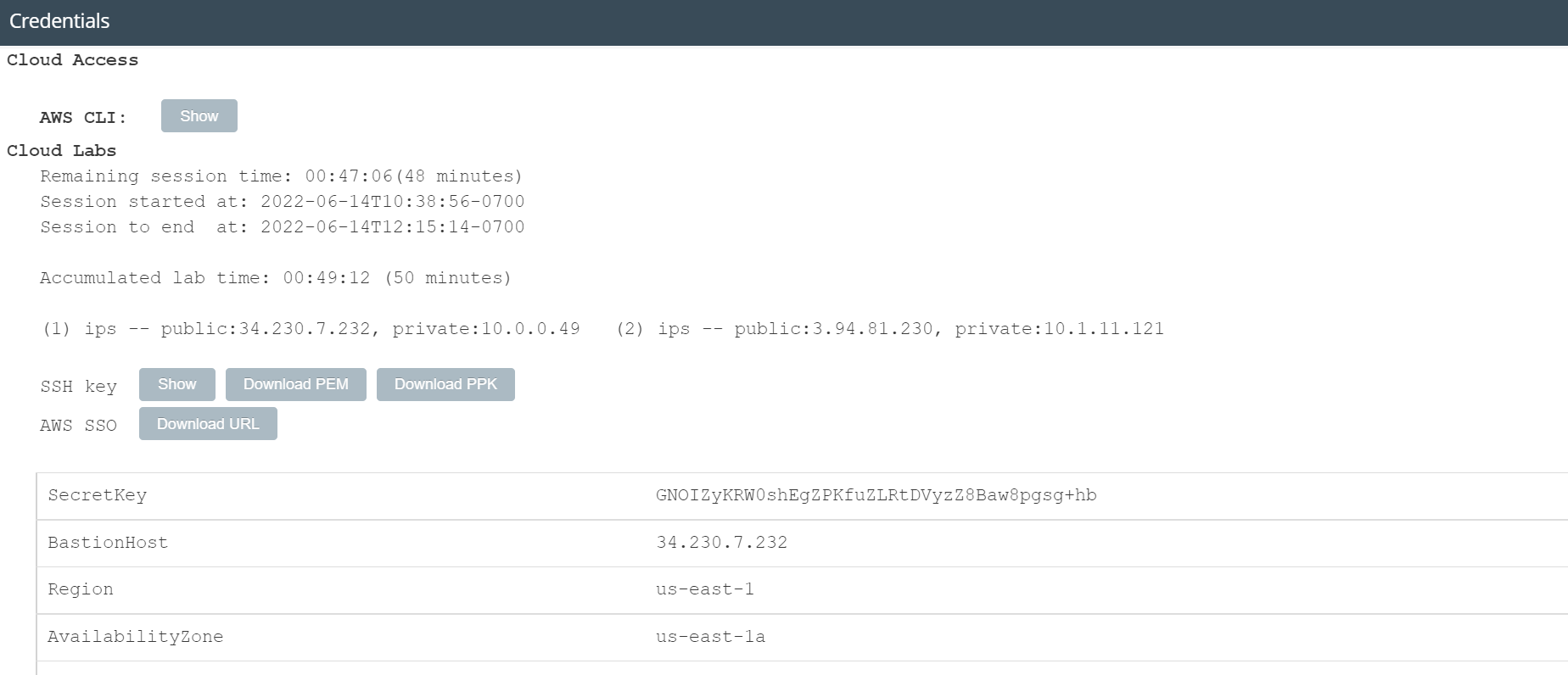


## **Task 3: Connect to Your Amazon EC2 Instance**

**Windows Users: Using SSH to Connect**

1. Download **labsuser.ppk**

* On canvas choose the **Details** menus and choose **Show**
* Choose the **Download PPK** button and save the file
* Then exit the panel



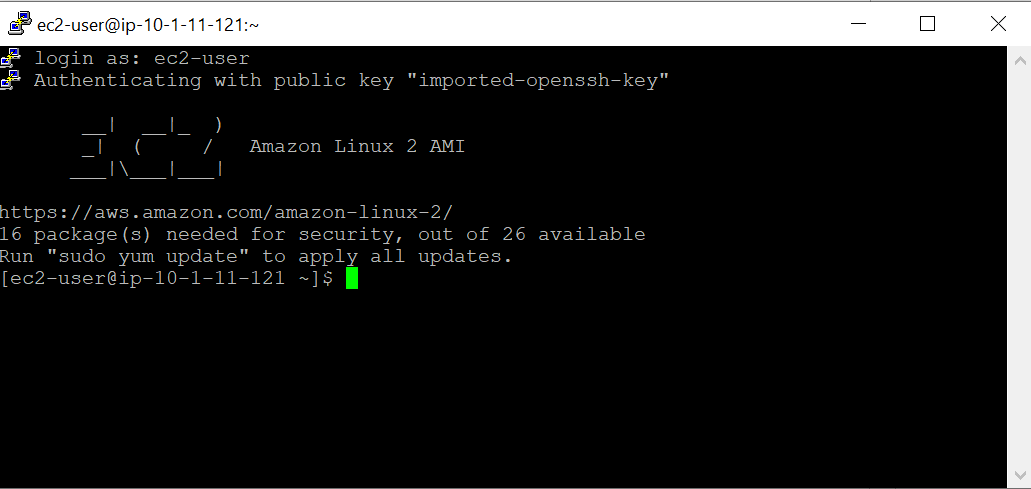
1. Open putty.exe
2. Configure not to time out

* Choose **Connection**
* Set **Seconds between keepalives** to 30

1. Configure Putty session

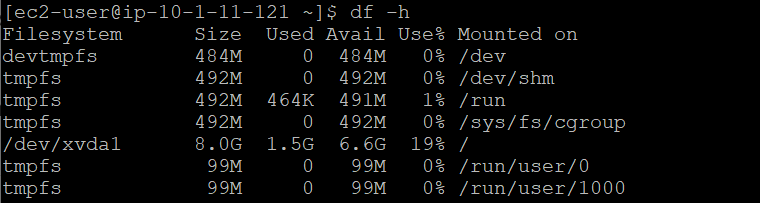
* Choose **Session**
* Paste **IPv4 Public IP address** of the EC2 instance
* In the **Connection** list, expand **SSH**
* Choose **Auth**
* Choose **Browse** and select the labsuser.ppk file
* Choose **Open**
* And choose **Yes**

1. When prompted **login as**, enter ec2-user



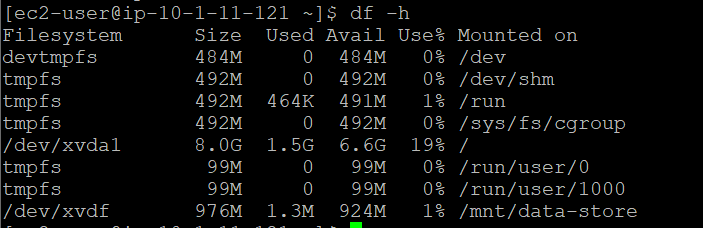
## **Task 4: Create and Configure Your File System**

1. Use df -h command to view the storage available



1. Create an ext3 file system on the new volume: sudo mkfs -t ext3 /dev/sdf
2. Create a directory for mounting the new storage volume: sudo mkdir /mnt/data-store
3. Mount the new volume: sudo mount /dev/sdf /mnt/data-store
4. View the configuration file to see the setting on the last line: cat /etc/fstab
5. Use the df -h to view the available storage again:

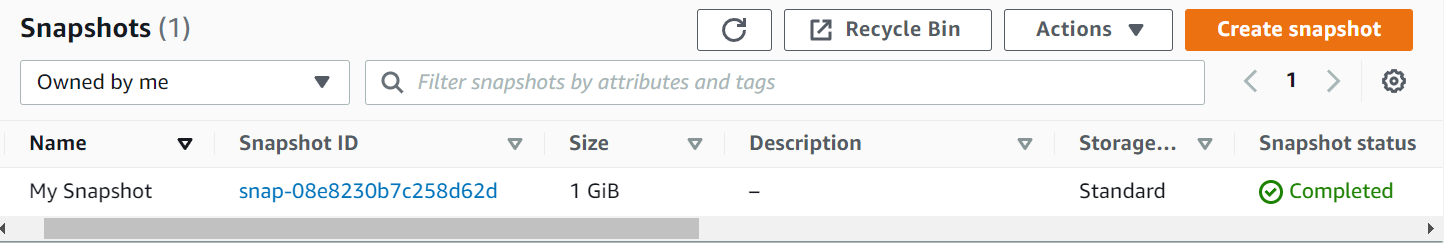
The output will now contain an additional line - */dev/xvdf*:



1. On your mounted volume, create a file and add some text to it: sudo sh -c "echo some text has been written > /mnt/data-store/file.txt"
2. Verify that the text has been written to your volume: cat /mnt/data-store/file.txt

## **Task 5: Create an Amazon EBS Snapshot**

1. In the **AWS Management Console**, choose **Volumes** and select the Volume created (**My Volume**)
2. In **Actions** menu, select **Create snapshot**
3. Choose **Add tag** then configure:
   1. **Key:** Name
   2. **Value:** My Snapshot
   3. Choose **Create snapshot**
4. In the left navigation pane, choose **Snapshots**.

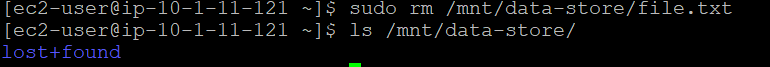


1. In your remote SSH session, delete the file that you created on your volume.

sudo rm /mnt/data-store/file.txt

1. Verify that the file has been deleted.

ls /mnt/data-store/



## **Task 6: Restore the Amazon EBS Snapshot**

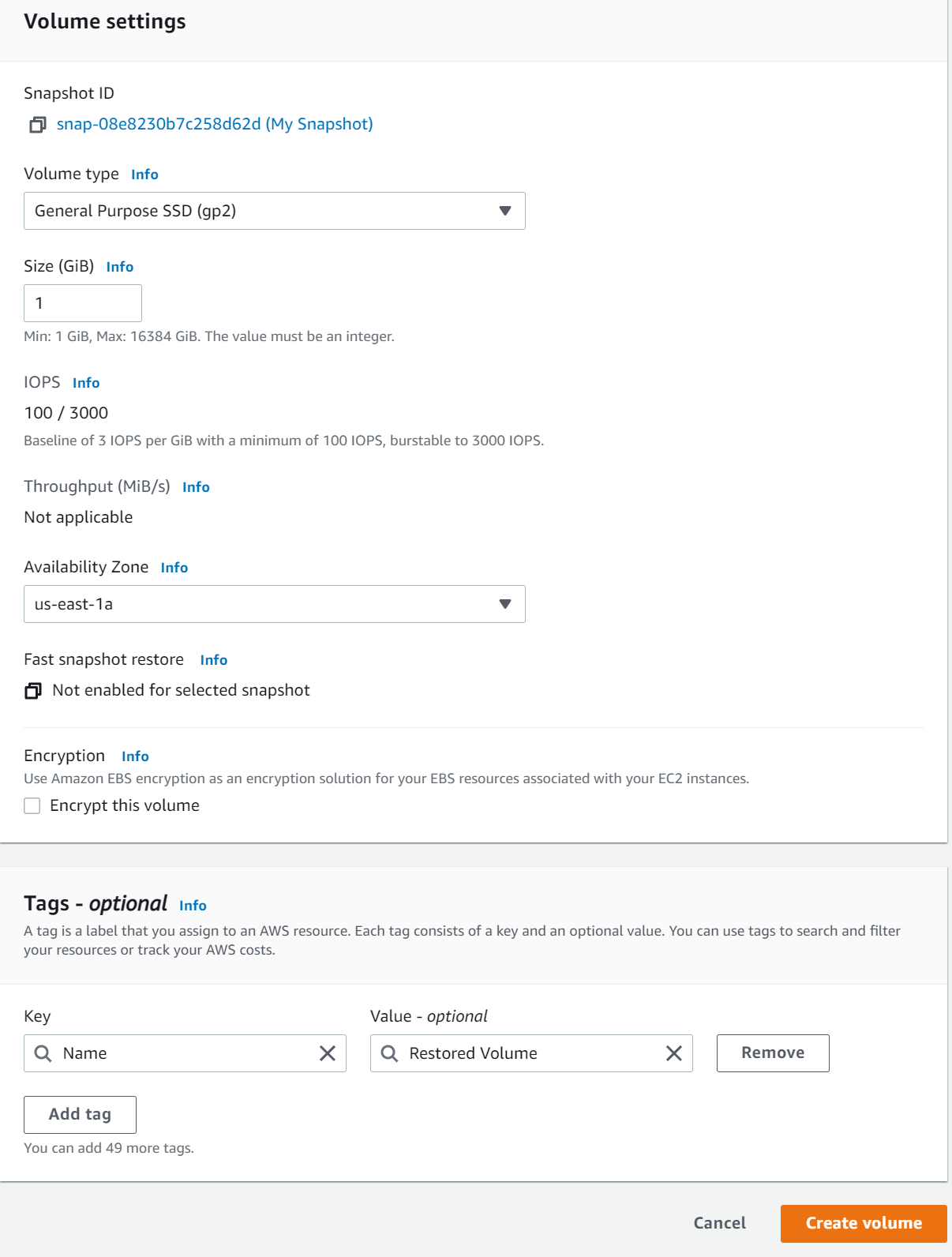
**Create a Volume Using Your Snapshot**

1. In the **AWS Management Console**, select **My Snapshot**
2. In the Actions menu, select **Create volume from snapshot**
3. Select the same availability for **Availability Zone**
4. Choose **Add tag** then configure:

**Key:** Name

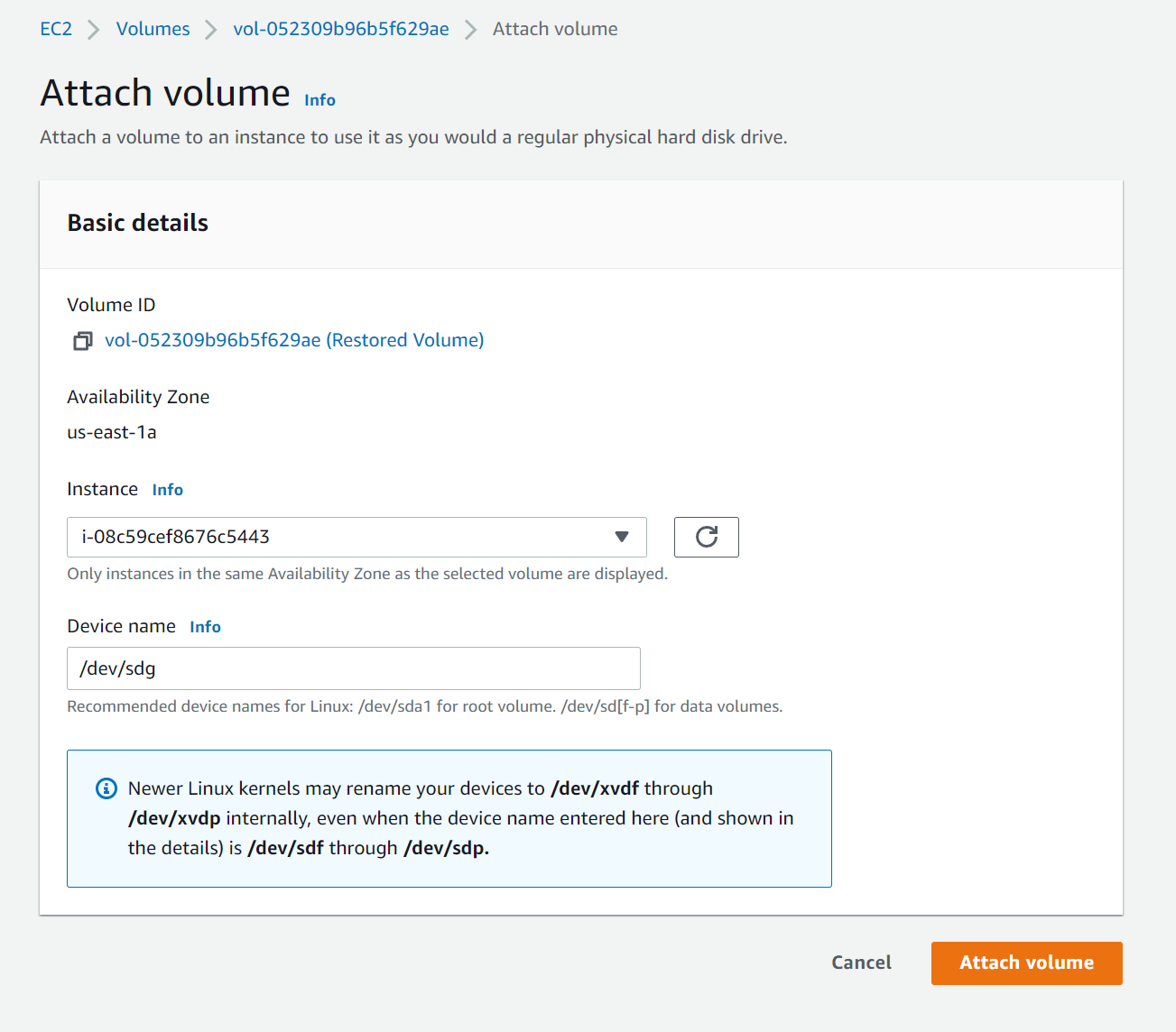
**Value:** Restored Volume

Choose **Create volume**



**Create a Volume Using Your Snapshot**

1. In the left navigation pane, choose **Volumes**
2. Select **Restored Volume**
3. In the **Actions** menu, select **Attach volume**
4. Choose the **Instance** field, then select the (Lab) instance that appears.
5. Choose **Attach volume**



**Create a Volume Using Your Snapshot**

1. Create a directory for mounting the new storage volume: sudo mkdir /mnt/data-store2
2. Mount the new volume: sudo mount /dev/sdg /mnt/data-store2
3. Verify that volume you mounted has the file that you created earlier. ls /mnt/data-store2/

