If you want to back up an AWS EC2 instance, you should create snapshots of EBS volumes, which are stored with the help of Amazon Simple Storage Service (S3).

Snapshots can capture all data within EBS volumes and create their exact copies.

Over the last decade, the sheer amount of data in the world has grown exponentially, thus making it hard for some organizations to manage and store critical pieces of information daily, let alone protect it from unexpected data loss because of hardware failure, software corruption, accidental deletion, malicious attack, or an unpredictable disaster.

More issues may arise still when it comes to managing AWS EC2 environments and protecting data stored in the cloud.

In short, AWS EC2 backup instances, you should choose one of the following options:

1. Take an EBS snapshot.
2. Create a new AMI.
3. Design an AWS EC2 Backup plan.
4. Automate AWS EC2 backup with a third-party solution.

AWS Backup is a rather new addition to the rich set of AWS services and tools and is worth your attention.

AWS Backup is a valuable tool which can **help you automatically back up and protect your data and applications in the AWS cloud as well as on-premises IT environments.**

If you want to learn how to back up AWS EC2 instances, read these notes which describes the different backup strategies available for the AWS EC2 environment.

How to Back Up AWS EC2 Instances

AWS is a high-performance, constantly evolving cloud computing platform that allows you to store data and applications in the cloud environment.

AWS can provide you with the tools you need to create EC2 instances which act as virtual servers with varying **CPU, memory, storage, and networking capacity.**

Currently, there are three ways to back up AWS EC2 instances**: taking EBS snapshots, creating AMIs, or designing an AWS Backup plan.**

Let us take a closer look at each of these approaches and see how they differ.

Taking EBS Snapshots

If you want to back up an AWS EC2 instance, you should create snapshots of **EBS volumes,** which are stored with the help of **Amazon Simple Storage Service (S3).**

Snapshots can capture all data within EBS volumes and create their exact copies. Moreover, these EBS snapshots can then be copied and transferred to another AWS region to ensure safe and reliable storage of critical data.

Thus, in case of a disaster or accidental data loss, you can be sure that you have a backup copy securely stored in a remote location which you can use for restoring critical data.

Prior to running **AWS EC2 backup,** it is **recommended that you stop the instance or at least detach an EBS volume which is about to** be backed up.

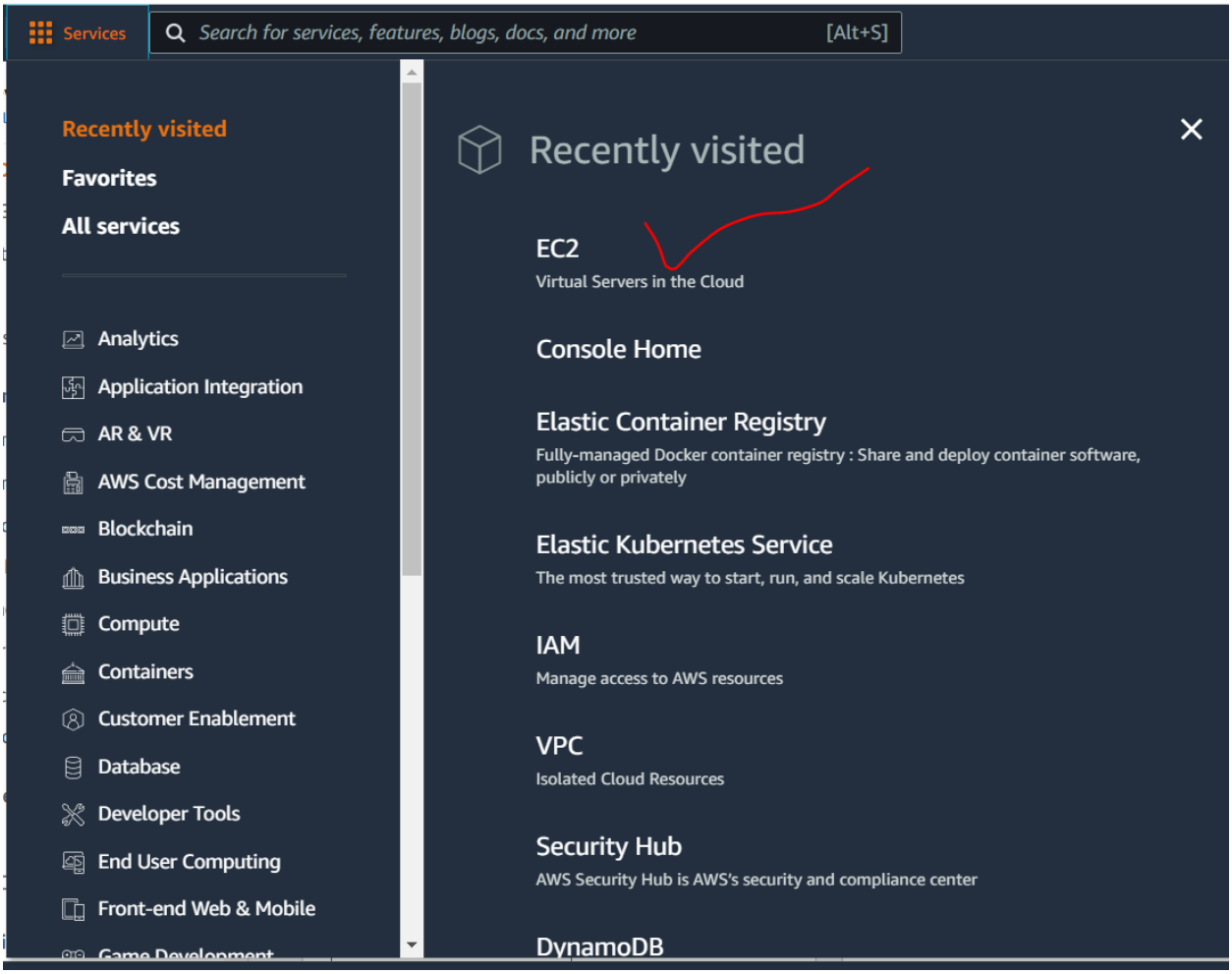
This way, you can prevent failure or errors from occurring and affecting the newly created snapshots.

Please note that, for security purposes, some sensitive information has been removed.

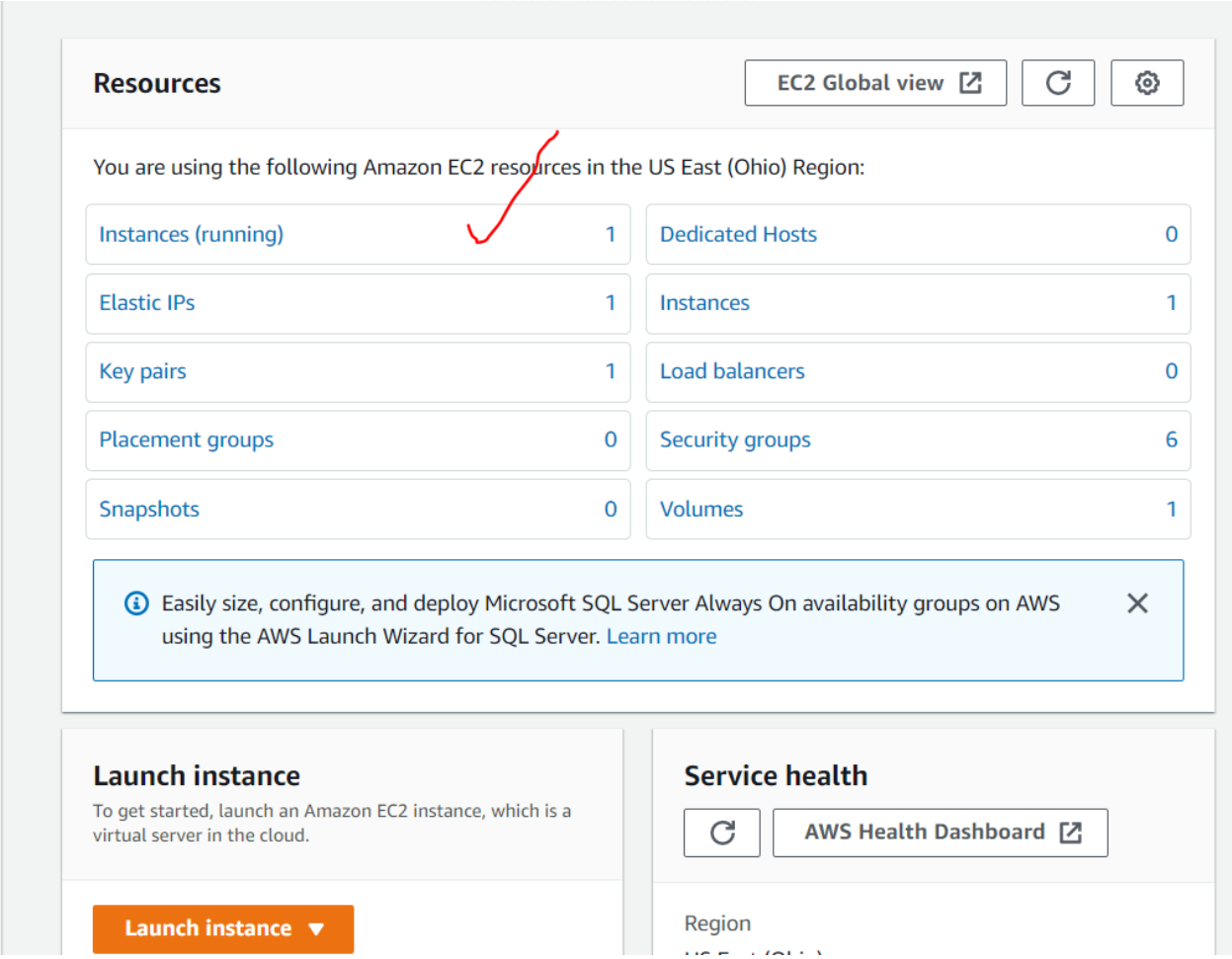
To back up AWS EC2 instance, you need to take the following steps:

1. Sign into your AWS account to open the AWS console.

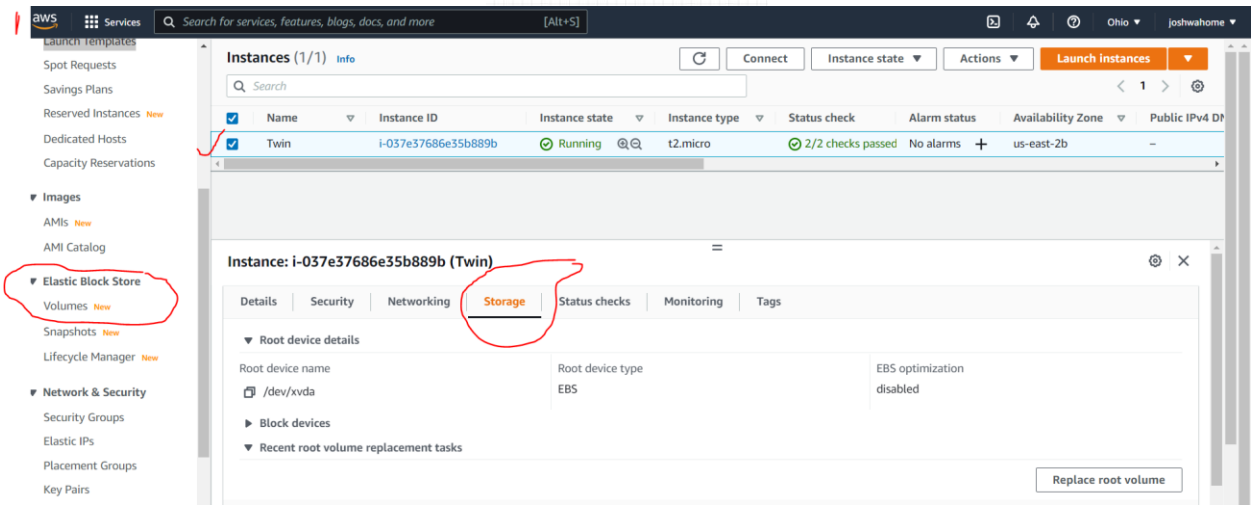
2. Select **Services** in the top bar and click **EC2** to launch the EC2 Management Console.



3. Select **Running Instances** and choose the instance you would like to back up.

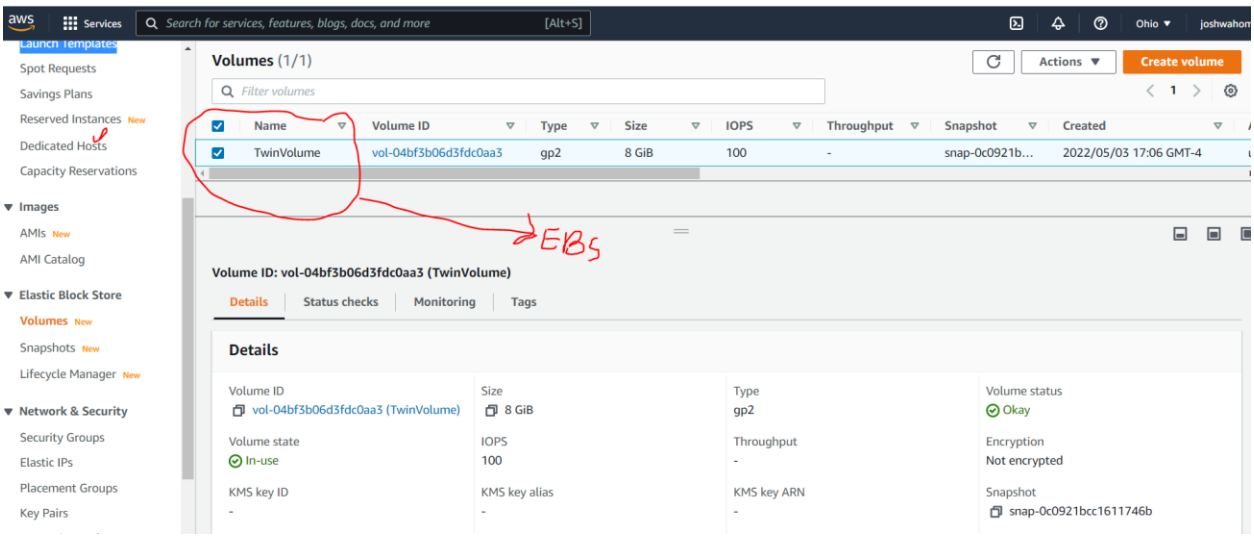


4. In the bottom pane, you can view the central technical information about the instance. In the **Description** tab, find the **Storage** section and select the **Volumes on your left-hand side.**

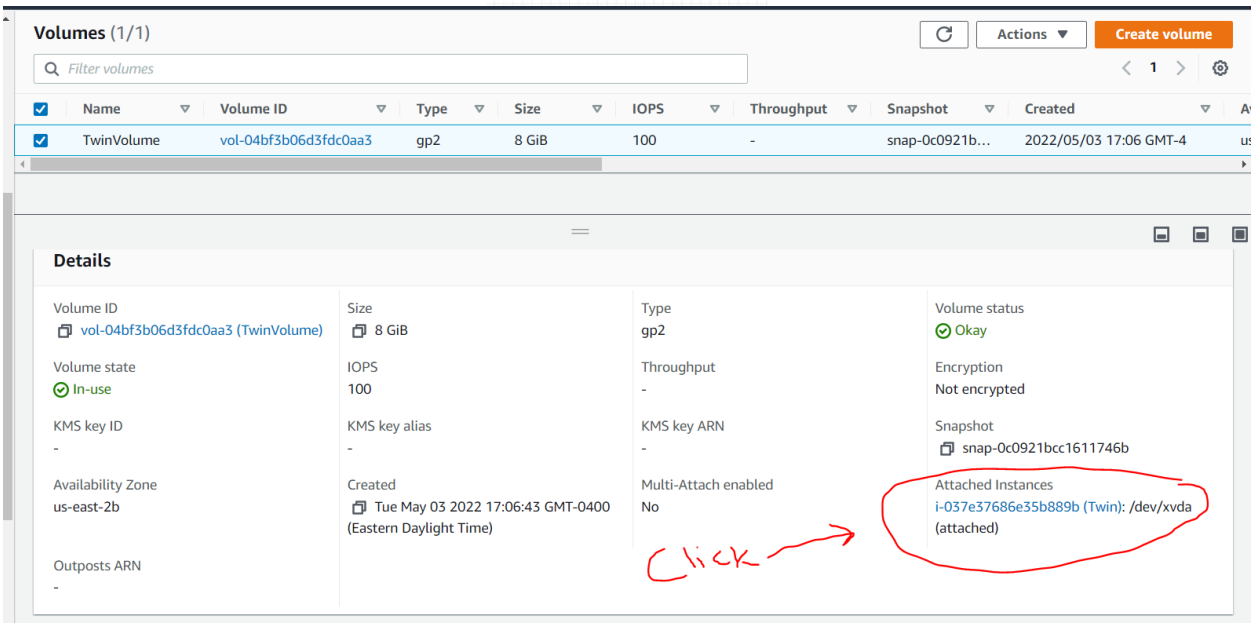


5. In the left-hand side of your console, click on Volume.

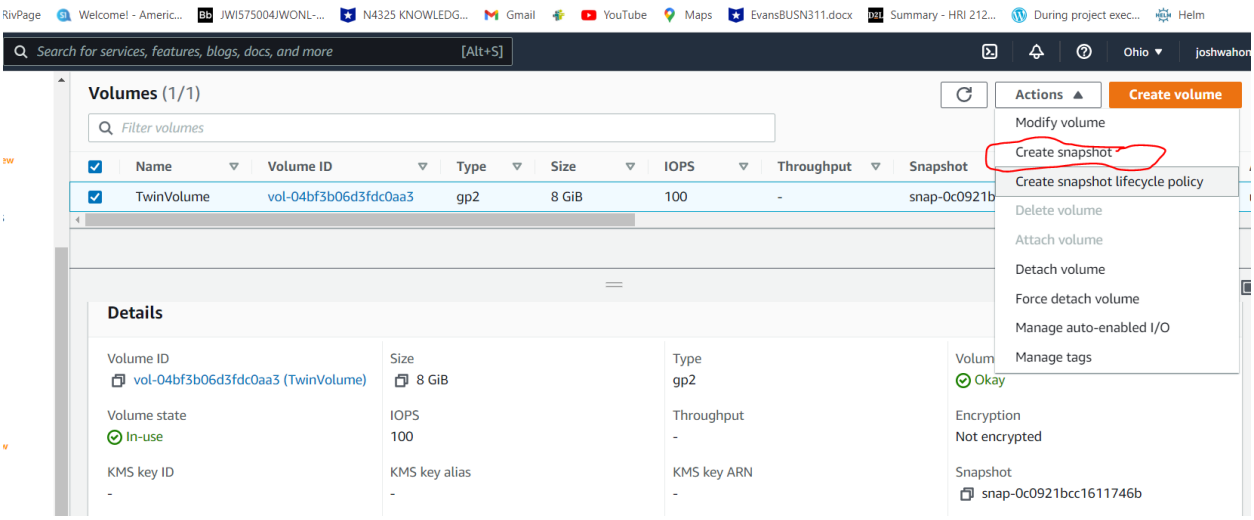
6. The Volumes section should open.



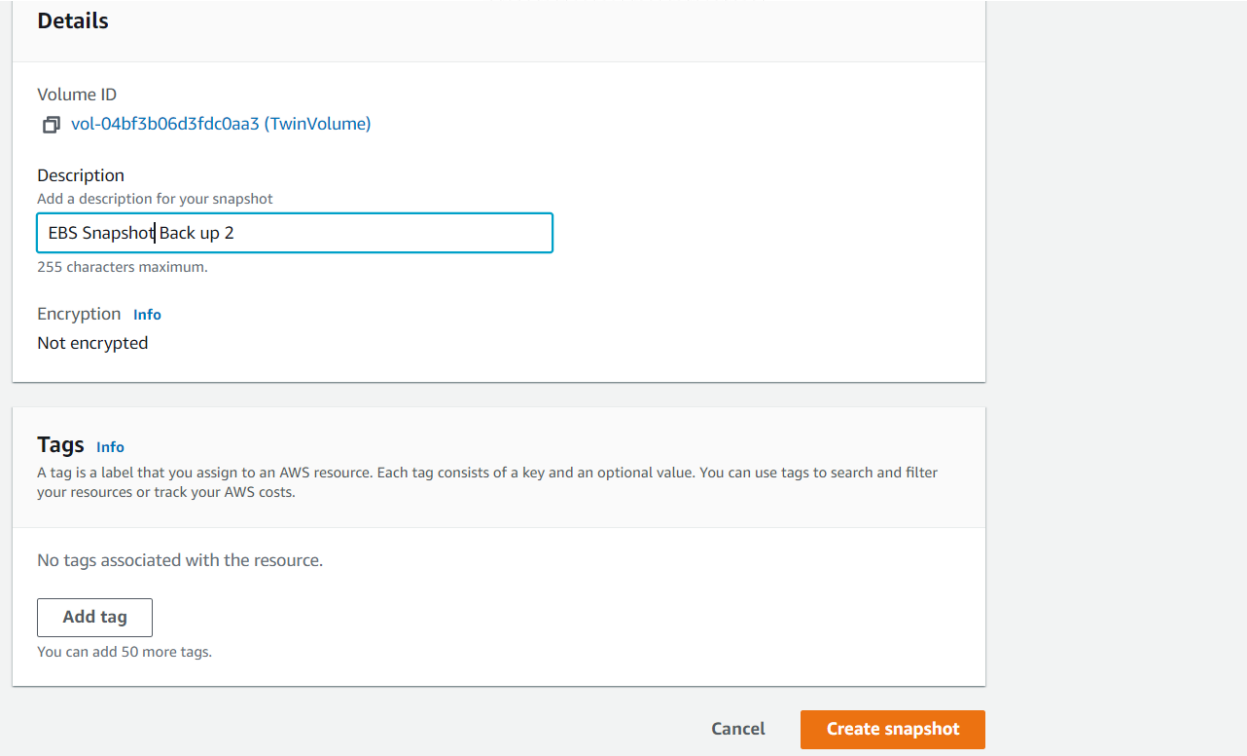
Click on the highlighted link and should take you back to the instance that owns this volume.



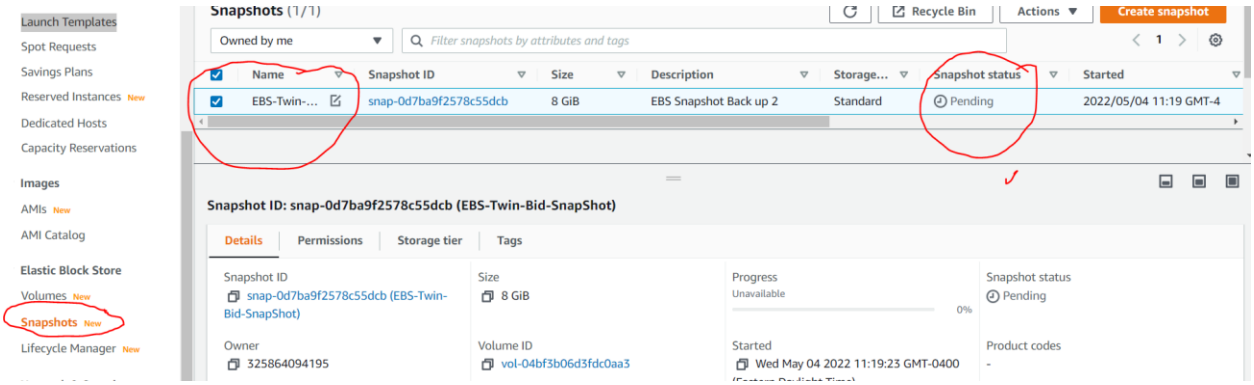
7. Click on Actions and select **Create Snapshot.**



8. Click on Create Snapshot and give your EBS snapshot a name.



9. Confirm the snapshot has been created

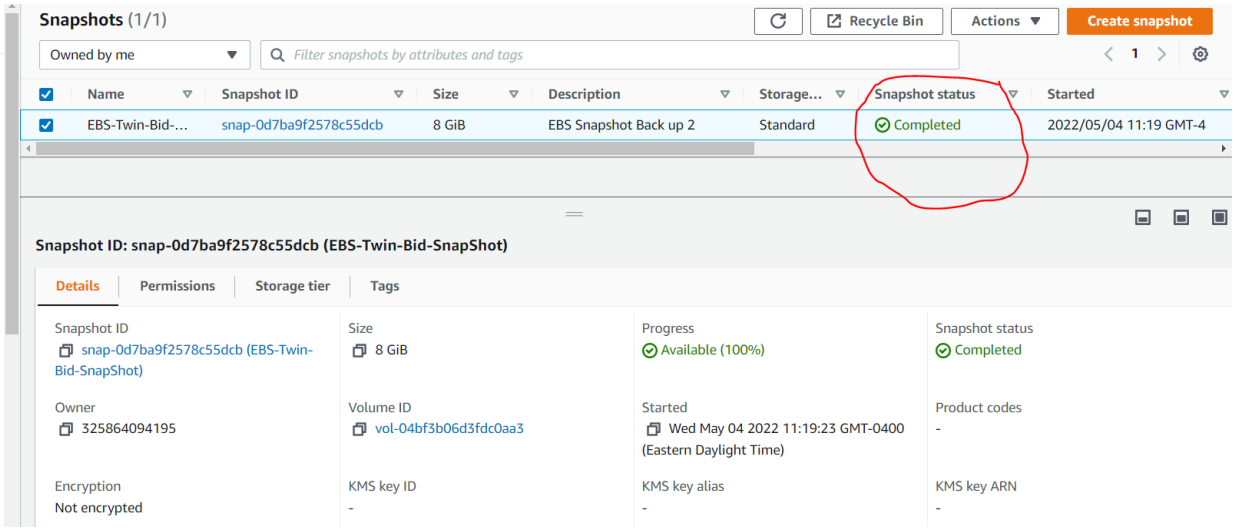


May take a few minutes to create. After the snapshot creation is complete, you can find your new snapshot by selecting the **Snapshots** section in the left pane.

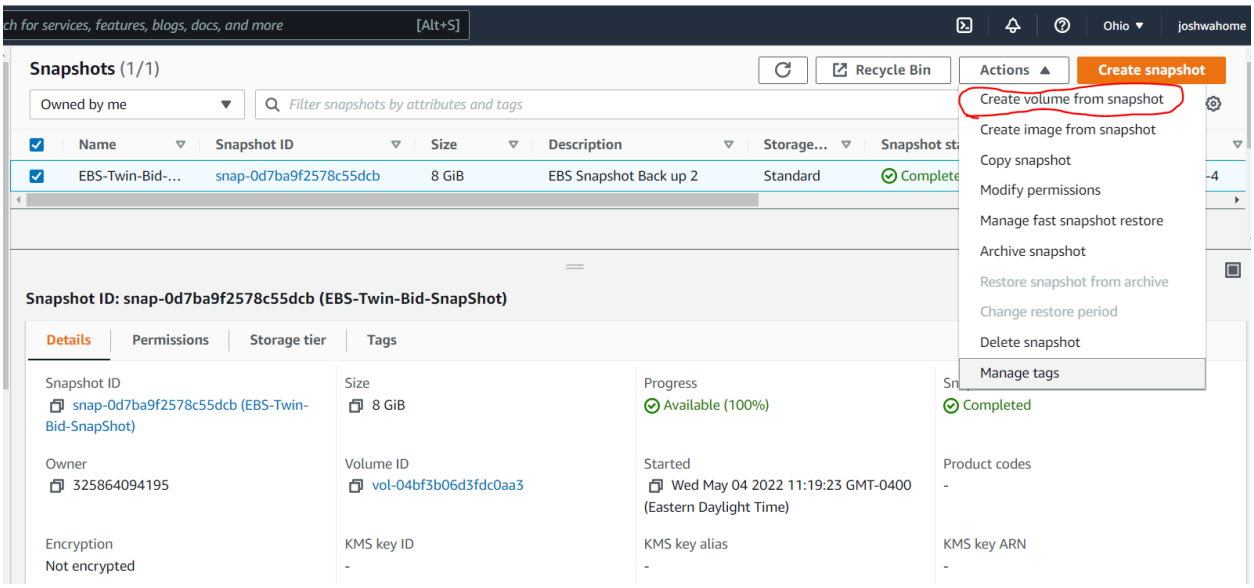
As you can see, we have successfully **created a point-in-time copy of the EBS volume,** which can later be used to restore your EC2 instance.

Next - **We need to create a volume from the snapshot backup of the EBS volume.**

The EBS Volume snapshot is completed



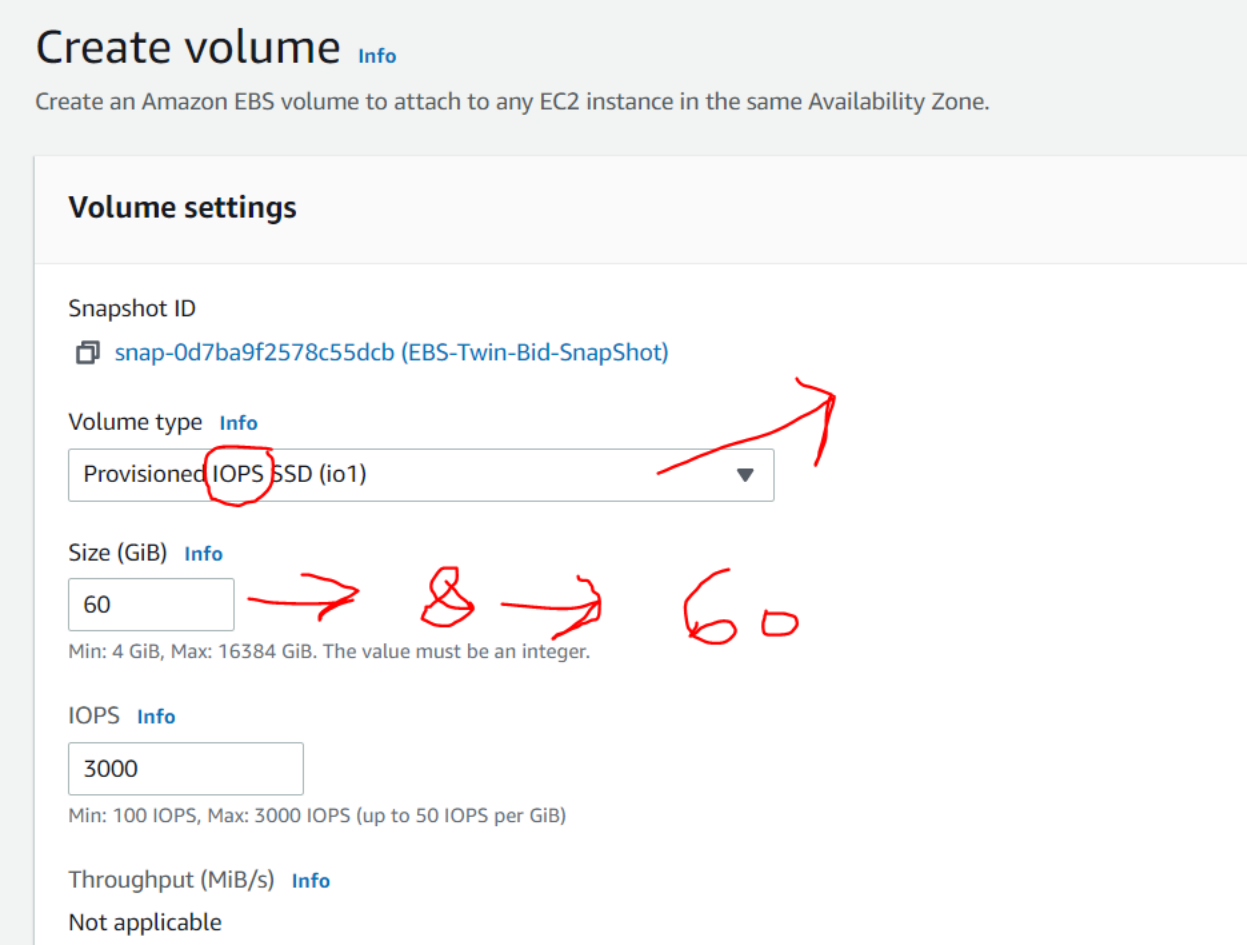
Let create a volume of the snapshot – Disaster Recovery Plan



For this purpose, you need to select the snapshot of the backed-up volume, press the **Actions** button above, and click **Create Volume**.

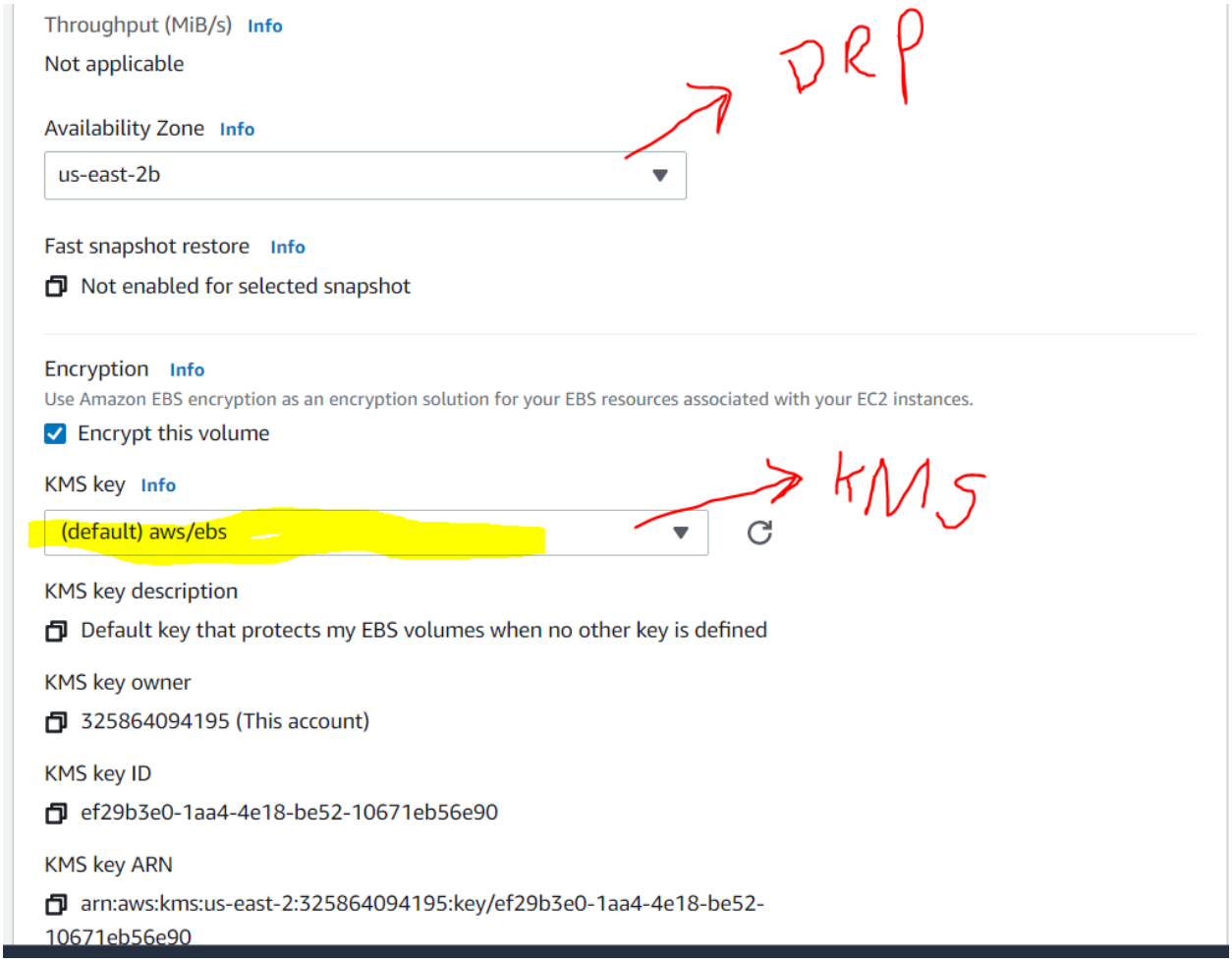
Following the prompts, configure the volume details (volume type, size, IOPS, availability zone, tags).

Then, click **Create Volume** for the new volume to be created, which can later be added to the AWS EC2 instance of your choice.

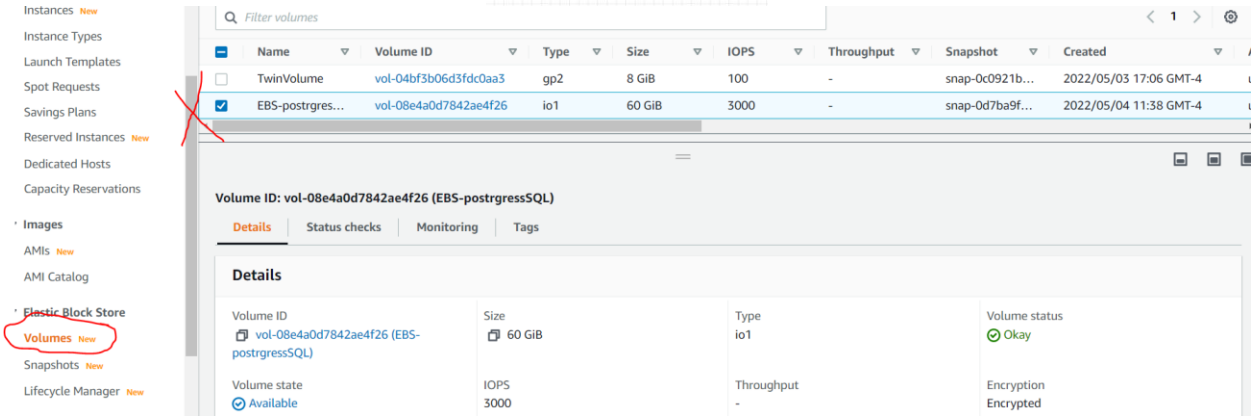


Change the Volume type to: IOPS SSD (Allow high throughput).

Change the Size from 8 to 60 GiB (This is the advantage of using volumes because they are dynamic).



- You should choose a different region for disaster recovery plan satisfaction.
- You must encrypt your volume to ensure it is secure
- You can create a new KMS to encrypt the volume or use the default KMS.



At this point, you have created two things

EBS Snapshot

EBS Snapshot Volume

You have done enough for back up and you can proceed with your activities.

Restoring from an Amazon EBS snapshot or an AMI

To reduce the recovery time and impact to dependent applications and processes, your restore process must consider the resource that it is replacing. For best results, regularly test your restore process in lower environments (for example, non-production) to verify that your process meets your recovery point objective (RPO) and recovery time objective (RTO) and that the restore process works as expected. Consider how the restore process will impact applications and services that depend on the instance you are restoring, and then coordinate the restore as necessary. Try to automate and test the restore process as much as possible to reduce the risk of you restore process failing or being implemented inconsistently.

Data from an Amazon EBS snapshot is asynchronously loaded into an EBS volume. If an application accesses the volume where the data is not loaded, there is higher latency than normal while the data is loaded from Amazon S3. To avoid this impact for latency-sensitive applications, you can pre-warm your data from a

snapshot into an EBS volume. For an additional charge, Amazon EBS support fast snapshot restore which reduces the need to pre-warm your data.

Your workload architecture impacts your restore procedure. For example, if you use Elastic Load Balancing, with multiple instances servicing traffic, you can take a failed or impaired instance out of service. Then you can restore a new instance to replace it while the other instances continue to service traffic without disruption to users.

The following restore processes described are for instances that are not using Elastic Load Balancing.

Restoring an EBS volume from an Amazon EBS snapshot

You can restore a non-root volume attached to an existing EC2 instance by creating a volume from a snapshot and attaching it to your instance. You can use the console, the AWS CLI, or the API operations to create a volume from an existing snapshot. You can then mount the volume to the instance by using the operating system.

If you are replacing a volume that must use the same mount point, unmount that volume so that you can mount the new volume in its place. To unmount the volume, first stop any processes that are using the volume.

For example, follow these steps to restore a volume to an earlier point-in-time backup by using the console:

1. On the Amazon EC2 console, on the **Elastic Block Store** menu, choose **Snapshots**.
2. Search for the snapshot that you want to restore and select it.
3. Choose **Actions**, and then choose **Create Volume**.
4. Create the new volume in the same Availability Zone as your EC2 instance.
5. On the Amazon EC2 console, select the instance.

6. In the instance details, make note of the device name that you want to replace in the **Root device** entry or **Block Devices** entries.
7. Attach the volume. The process differs for root volumes and non-root volumes.

   For root volumes:

   . Stop the EC2 instance.

   a. On the **EC2 Elastic Block Store Volumes** menu, select the root volume that you want to replace.

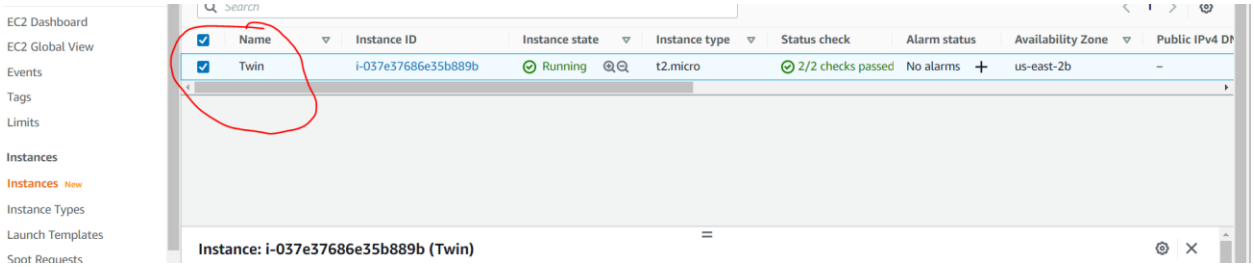
   b. Choose **Actions**, and then choose **Detach Volume**.

   c. On the **EC2 Elastic Block Store Volumes** menu, select the new volume.

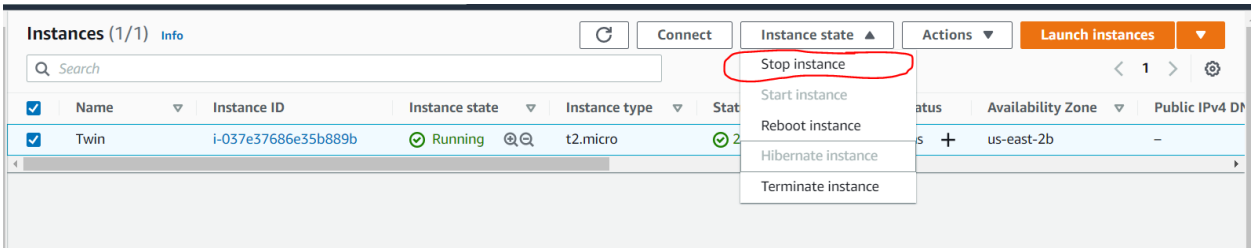
   d. Choose **Actions**, and then choose **Attach Volume**.

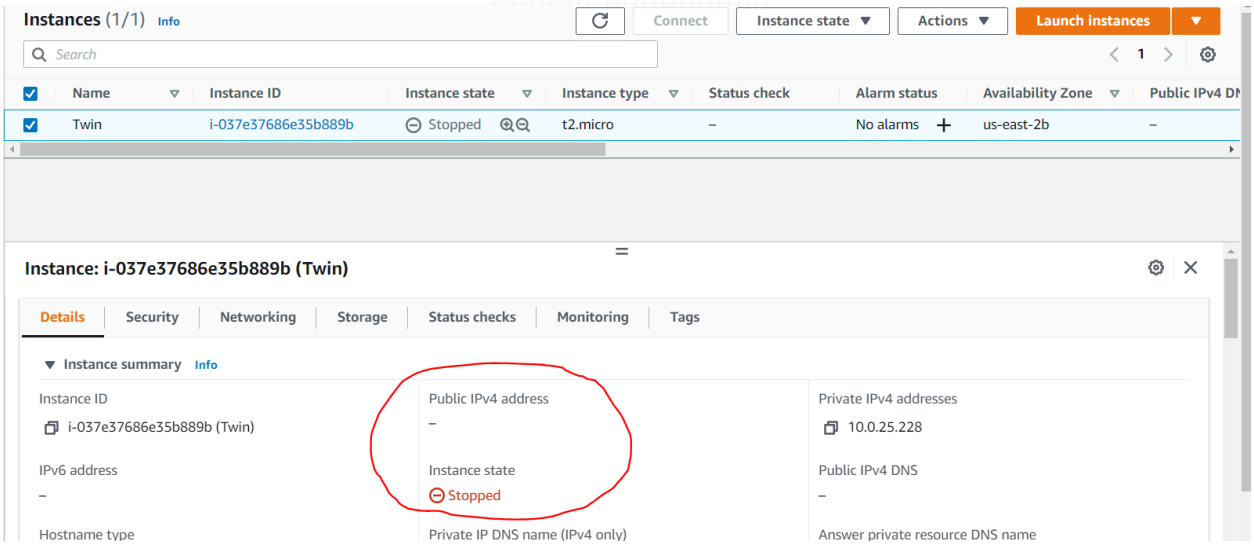
   e. Select the instance that you want to attach the volume to, and use the same device name that you noted earlier.

   We want to restore the root volume with the snapshot volume one we just created

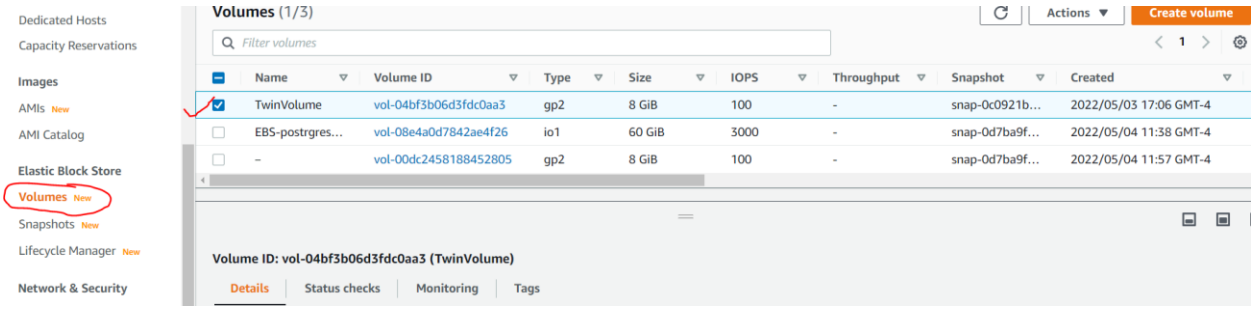


Step 1. Let us stop this instance

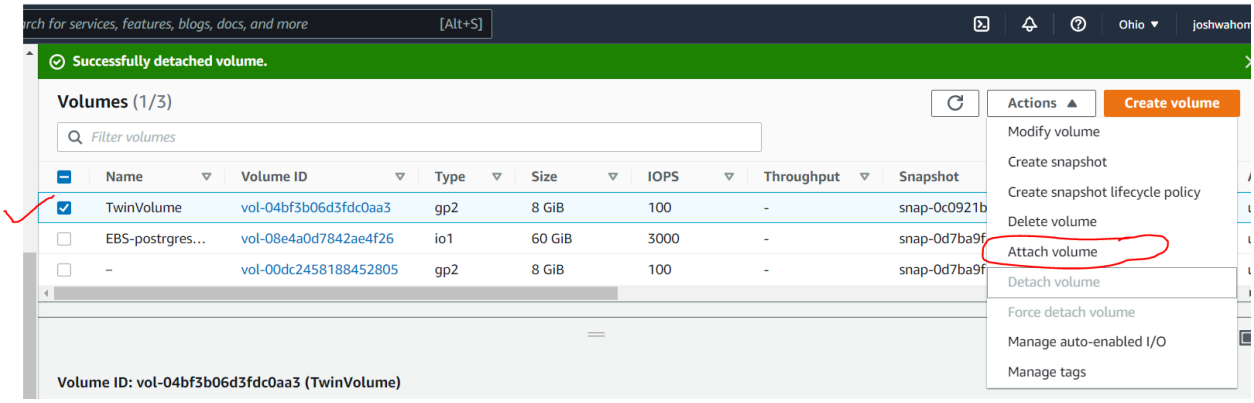




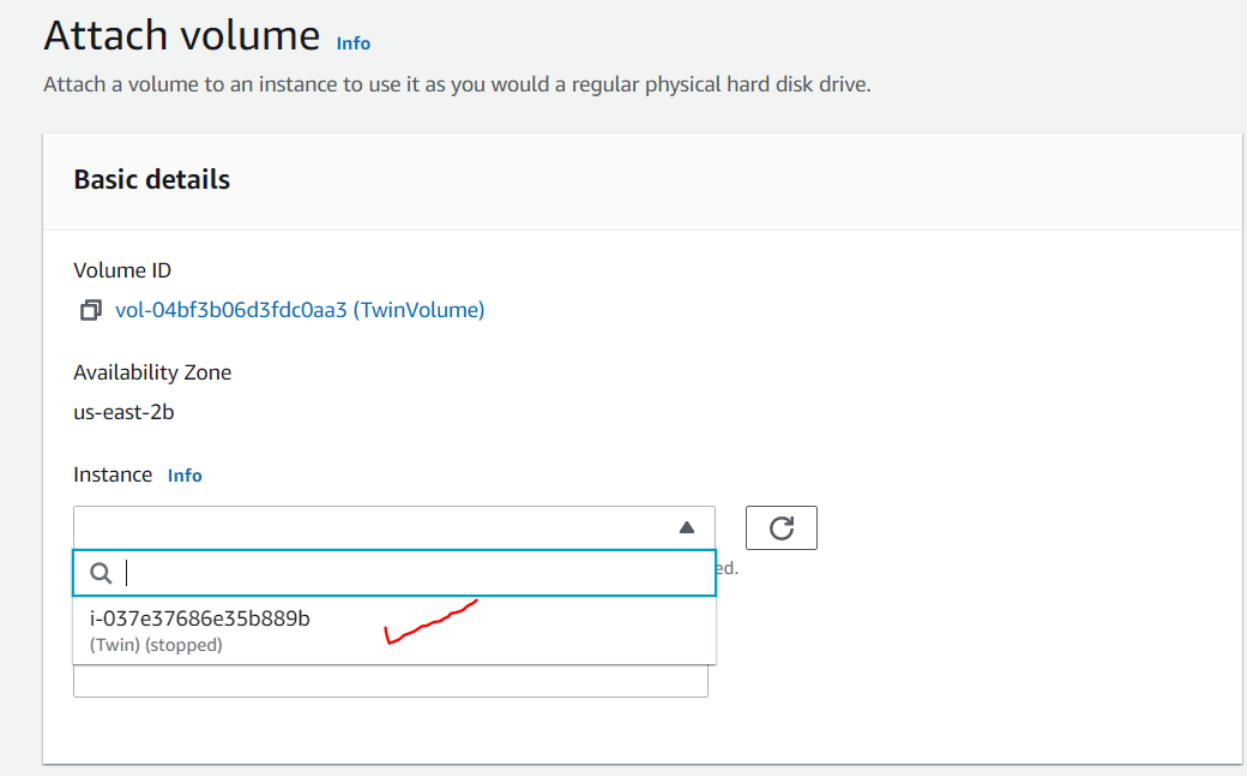
We want to attach the EBS Volume Snapshot we just created.



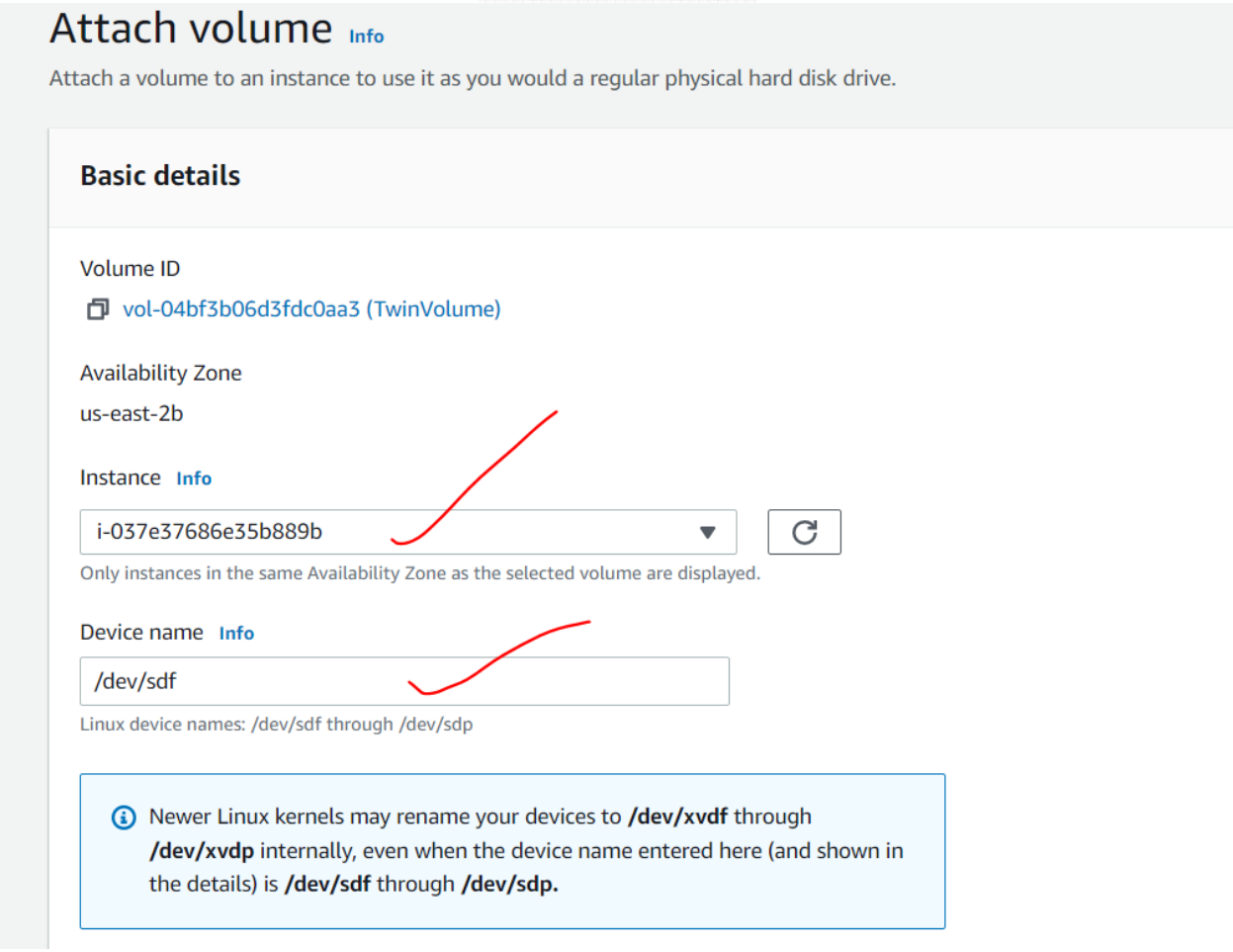
Click on Actions



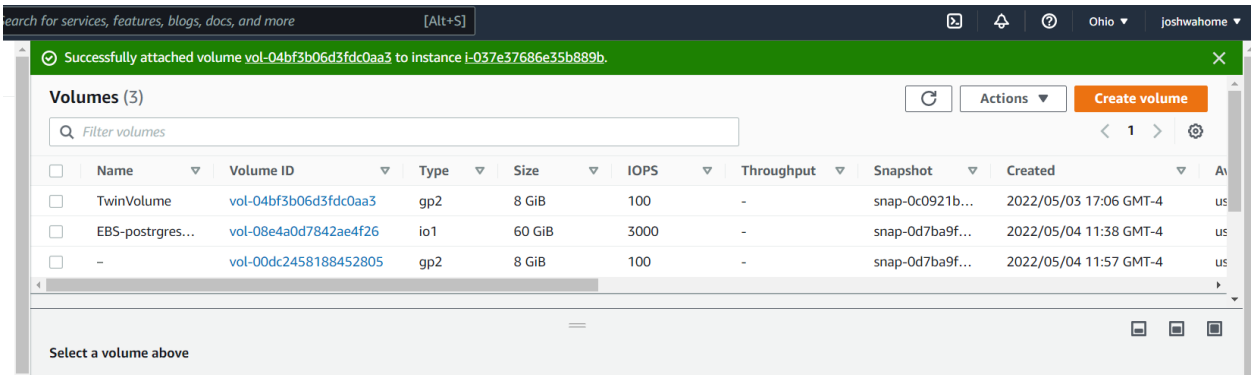
Select Attach volume



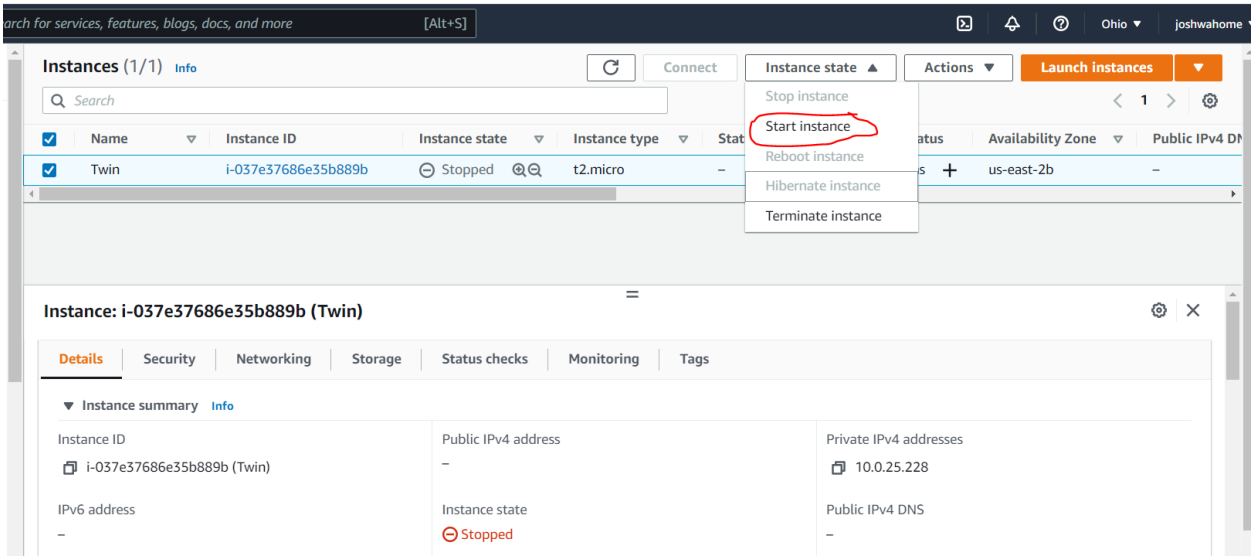
Selected the stopped instance



Attach the volume



Go back to the stopped instance and start it to pick the new attached volume (This should be a brand new instance)



When the instance completely starts, you will have succeeded in restoring the volume.

TRY FOR THE NON-ROOT VOLUMES

For non-root volumes:

f. On the **EC2 Elastic Block Store Volumes** menu, select the root volume that you want to replace.
g. Choose **Actions**, and then choose **Detach Volume**.
h. Attach the new volume by choosing it on the **EC2 Elastic Block Store Volumes** menu and then choosing **Actions**, **Attach Volume**. Select the instance that you want to attach it to, and then select an available device name.
i. Using the operating system for the instance, unmount the existing volume, and then mount the new volume in its place.

In Linux, you can use the umount command. In Windows, you can use a logical volume manager (LVM) such as the Disk Management system utility.

j. Detach any prior volumes that you may be replacing by choosing it on the **EC2 Elastic Block Store Volumes** menu and then choosing **Actions**, **Detach Volume**.

You can also use the AWS CLI in combination with operating system commands to automate these steps.