Amazon Web Services (AWS), you are likely familiar with EBS, Elastic Block Store. It is a block-level storage service for EC2 instances, which means it is a persistent storage device that you can attach to your EC2 instances and use like a physical hard drive.

What are AWS EBS Snapshots

EBS snapshots are point-in-time copies of your EBS volumes that you can use to back up your data. By creating a backup (an EBS snapshot) of your disk (the EBS volume), you can restore a new EBS volume from that snapshot.

EBS snapshots are incremental, which means they only capture the data that has changed since the last snapshot. For example, if you have a 100 GB volume and you take a snapshot, then make a small change to the volume and take another snapshot, the second snapshot will only contain the data that has changed since the first snapshot. This makes EBS snapshots more efficient and cost-effective than full-volume backups. The size of an EBS snapshot is calculated based on the amount of data stored in the volume at the time the snapshot was taken.

There are two categories of EBS snapshots: Standard snapshots and Archive snapshots. Standard snapshots are stored in Amazon S3 and are designed for fast recovery of data. They are the default type of snapshot and are suitable for most use cases. Archive snapshots, on the other hand, are stored in Amazon S3 Glacier and are designed for long-term data retention. They are more cost-effective than Standard snapshots, but have retrieval times of a few hours (because they are retrieved from Glacier instead of S3).

We will use Python 2.7 scripts, lambda, IAM role, and cloud watch event schedule for this setup.

need to create a tag named “backup” with the value true for all the instances for which you need a backup.

A snapshot creation python script with the necessary parameters.

An IAM role with snapshot create, modify, and delete access.

A lambda function.

[AWS Step Functions](https://aws.amazon.com/step-functions/) serves just this purpose―to help you coordinate your functions and microservices. Step Functions enables you to simplify your effort and pull the error handling, retry logic, and workflow logic out of your Lambda code. Step Functions integrates with Lambda to provide a mechanism for building complex serverless applications. Now, you can kick off a Step Functions state machine based on a CloudWatch event.

target Step Functions in a CloudWatch Events rule. This allows you to have event-driven snapshot management based on snapshot completion events firing in CloudWatch Event rules.