AWS Rollback Solutions

AWS CodeDeploy:

AWS CodeDeploy is a deployment service that automates the release process for applications running on EC2 instances, on-premises servers, or Lambda. For a single EC2 instance, CodeDeploy can be configured to automatically roll back to a previous application revision if a deployment fails or a specified CloudWatch alarm is triggered. This helps ensure stability by quickly restoring the last known good version without manual intervention.

We can even write a quick upload script to handle all this automatically — it can bundle your files, create the appspec.yml, upload everything to S3, and trigger the deployment — making every upload rollback-friendly without you having to do it all manually. Each revision includes your app files and an appspec.yml file that defines how the deployment should happen. When you push a new revision to S3, GitHub, or Bitbucket and deploy it through CodeDeploy, that point becomes a trackable revision, allowing you to later roll back to it if needed.

AMIs and EBS Snapshots:

AMIs (Amazon Machine Images) and EBS snapshots are two ways to create rollback points for a single EC2 instance. An AMI captures the entire instance — OS, app, and settings — so you can launch a new instance exactly like the old one if something goes wrong. EBS snapshots back up the volume (like the disk), letting you restore or swap out a broken volume without replacing the whole instance. Both are great for quickly recovering from bad updates or accidental changes.

An AMI is essentially a full backup of your EC2 instance at a specific point in time. It includes the operating system, configuration settings, installed applications, and attached EBS volumes. When you create an AMI from an EC2 instance, AWS saves everything needed to launch a brand-new instance with the exact same setup. For rollback purposes, you can create an AMI before making any major changes (like updating software or modifying system configs). If something breaks or doesn't work as expected, you can quickly launch a new EC2 instance from that AMI and be back to a working version in minutes. You can even assign the same Elastic IP to the new instance so users won’t notice the switch. AMIs can be stored indefinitely and versioned manually (e.g., myapp-v1.2.3), giving you full control over your rollback points. This makes them perfect for safe updates, test environments, and disaster recovery.