Lab 2: Versioning & MFA Delete

What Worked as Expected

The objective of this lab was to configure S3 versioning, explore how version management works in practice, and understand the role of MFA Delete in securing object deletions. I started by creating an S3 bucket with versioning enabled and then uploaded an object to test how changes were tracked. After updating the file several times, I observed that multiple versions were being maintained automatically. When I attempted to delete the object, instead of being completely removed, a delete marker was created. This confirmed that versioning was working as expected, since I was able to restore the file simply by removing the delete marker.

What Was Challenging

One of the challenges I encountered was distinguishing between turning versioning on and off versus simply using the “Show versions” toggle in the S3 console, which initially caused some confusion. Another difficulty was enabling MFA Delete, since it cannot be done through the AWS console. It required the root user and the AWS CLI, along with careful steps to configure it properly. I also realized that MFA Delete depends on having a physical or virtual MFA device already linked to the account, which is an important prerequisite I had to keep in mind.

Real-World Application

From a real-world perspective, I see how versioning is extremely valuable for protecting data. It provides a safety net against accidental deletions, overwrites, or even malicious activities like ransomware by preserving the history of object changes. Adding MFA Delete enhances this further by requiring multi-factor authentication before any permanent deletions can happen. This ensures that even if an attacker gained access to credentials, they would still not be able to delete critical data without the MFA code. I can see this being especially important in industries such as finance, healthcare, or legal, where data integrity and recovery are crucial.

Cost Implications Observed

In terms of cost, I noticed that every object version is billed separately. This means that frequent updates to files could result in significant storage costs if versions are not managed properly. Without a lifecycle policy to automatically remove older versions, costs could escalate quickly in a production environment. On the other hand, MFA Delete itself does not add any extra charges, but it does introduce some operational overhead since all permanent deletions would require MFA authentication.