Aim -

To study and Implement Storage as a Service using Own Cloud/ AWS, Glaciers.

Theory -

What is Storage as a Service(STaaS)?

Storage as a Service (STaaS) is a cloud computing model that provides users with on-demand access to storage resources over the internet. In STaaS, users can store, access, and manage data in a remote storage infrastructure maintained by a third-party service provider, eliminating the need for users to invest in and maintain their own storage hardware and infrastructure.

Advantages:

- Cost Efficiency: STaaS follows a pay-as-you-go pricing model, where users only pay for the storage resources they consume. This eliminates the need for upfront investment in hardware and infrastructure, reducing capital expenditure and enabling cost-effective scaling based on demand.
- Scalability: STaaS platforms offer scalability features that allow users to easily scale their storage resources up or down based on changing requirements. This elasticity enables organizations to accommodate growing data volumes and fluctuating storage needs without the need for manual intervention.
- Reduced Management Overhead: STaaS abstracts the complexity of managing storage infrastructure, including hardware provisioning, software installation, data replication, backup, and maintenance. This reduces administrative overhead and allows organizations to focus on their core business activities rather than storage management tasks.
- High Availability and Data Durability: Many STaaS providers offer built-in redundancy, data replication, and backup capabilities to ensure high availability and data durability. This helps protect against data loss and ensures continuous access to data even in the event of hardware failures or disasters.
- Accessibility and Anywhere Access: STaaS enables users to access and manage their data from anywhere with an internet connection, providing flexibility and convenience for remote work, collaboration, and data sharing across distributed teams and locations.

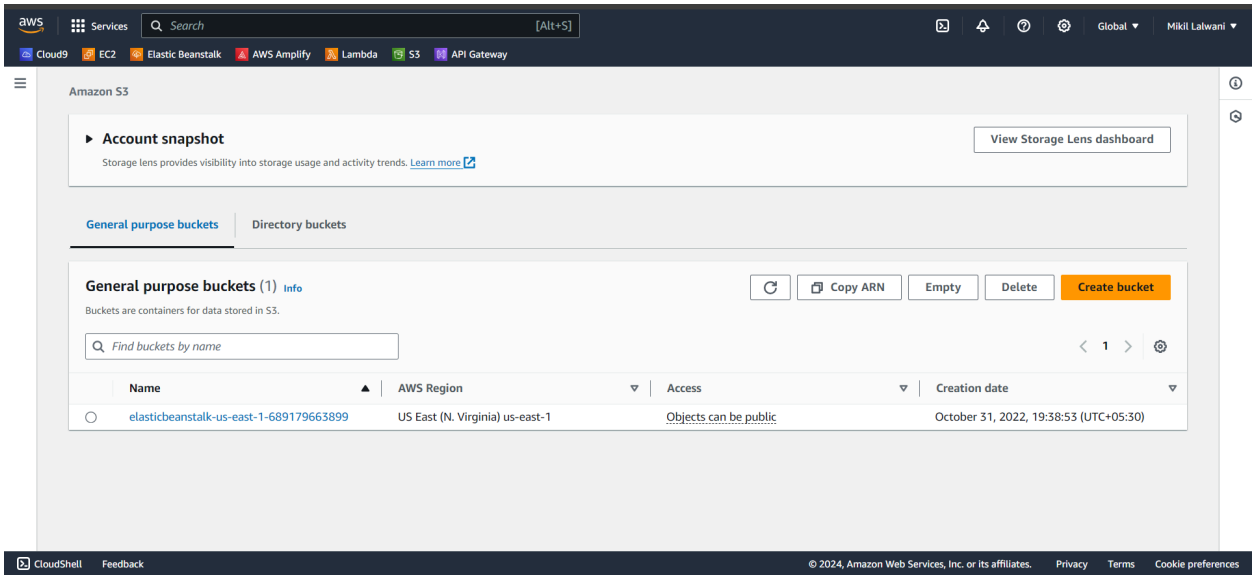
Disadvantages:

- Security and Privacy Concerns: Storing sensitive data in a third-party cloud environment raises concerns about data security and privacy. Organizations must carefully evaluate the security measures implemented by STaaS providers to ensure compliance with regulatory requirements and protect their data from unauthorized access and breaches.

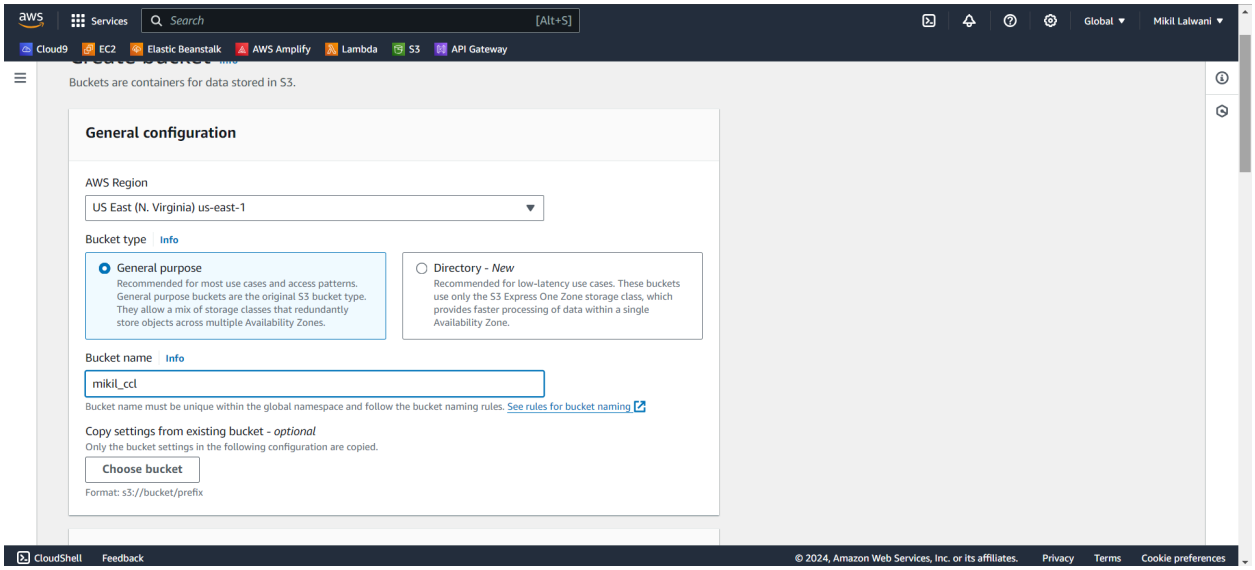
- Dependency on Internet Connectivity: STaaS relies on internet connectivity for accessing and managing storage resources, which may pose challenges in environments with limited or unreliable internet connectivity. Organizations should consider the potential impact of internet outages or disruptions on their data access and storage operations.
- Data Transfer Costs: Transferring large volumes of data to and from STaaS platforms may incur additional data transfer costs, particularly for organizations with high data volumes or frequent data migrations. Organizations should carefully consider data transfer costs when evaluating the total cost of ownership of STaaS solutions.
- Vendor Lock-In: Migrating data between different STaaS providers or transitioning from STaaS to an on-premises storage deployment may be challenging and costly, leading to vendor lock-in. Organizations should consider the long-term implications of vendor lock-in and evaluate strategies to mitigate this risk.
- Performance and Latency: Performance and latency issues may arise in STaaS environments, particularly in multi-tenant deployments where resources are shared among multiple users. Organizations should assess the performance characteristics of STaaS offerings and ensure they meet their performance requirements before migration.

Procedure -

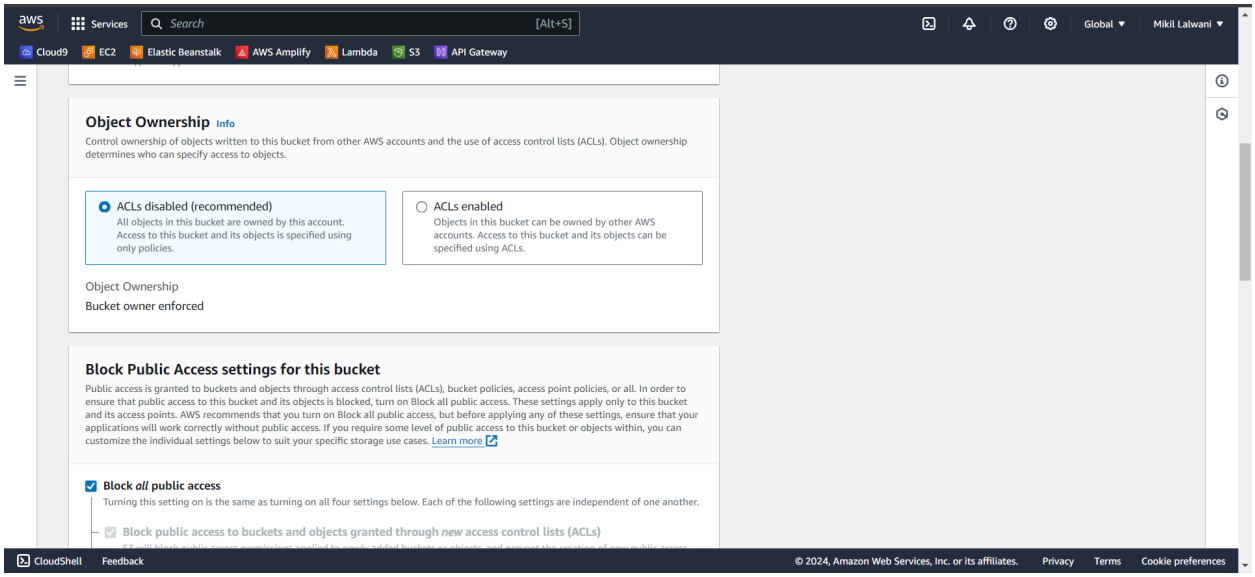
Step-1: click on create bucket



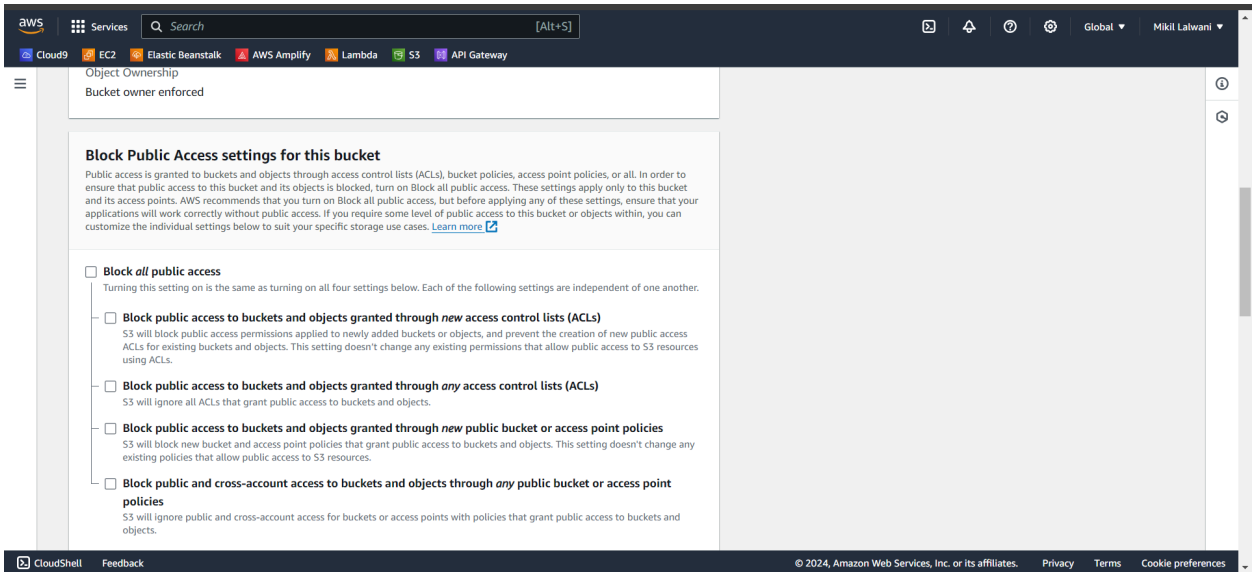
Step-2: Give Bucket name & select region for storage



Step-3: Keep object ownership setting as ACLs Disabled as by-default

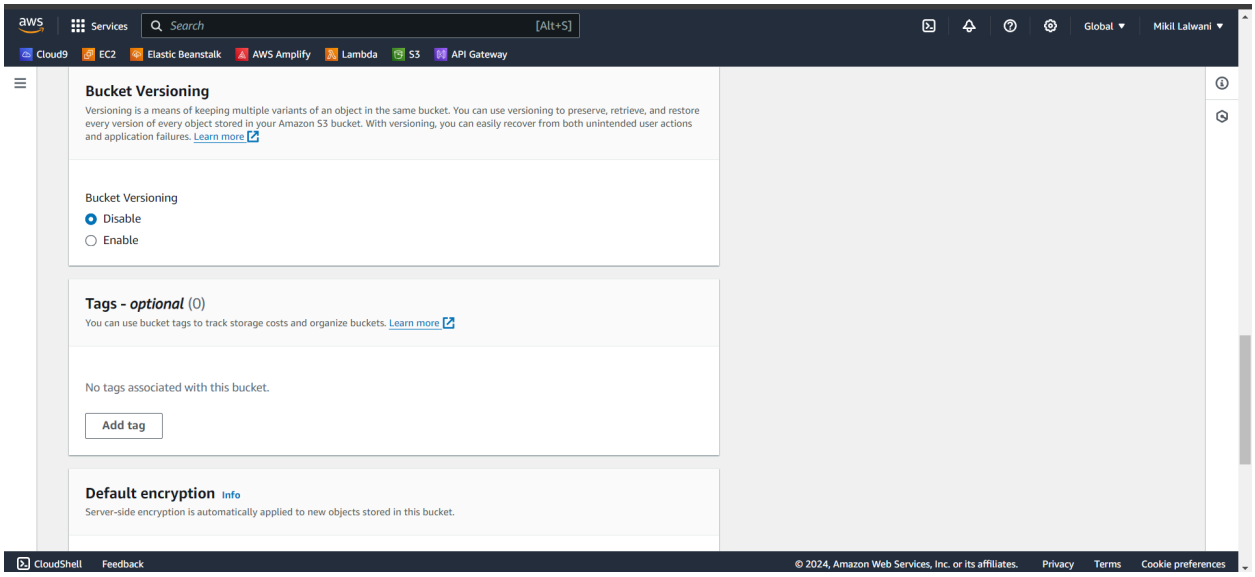


Step-4: Disable block all public access checkbox

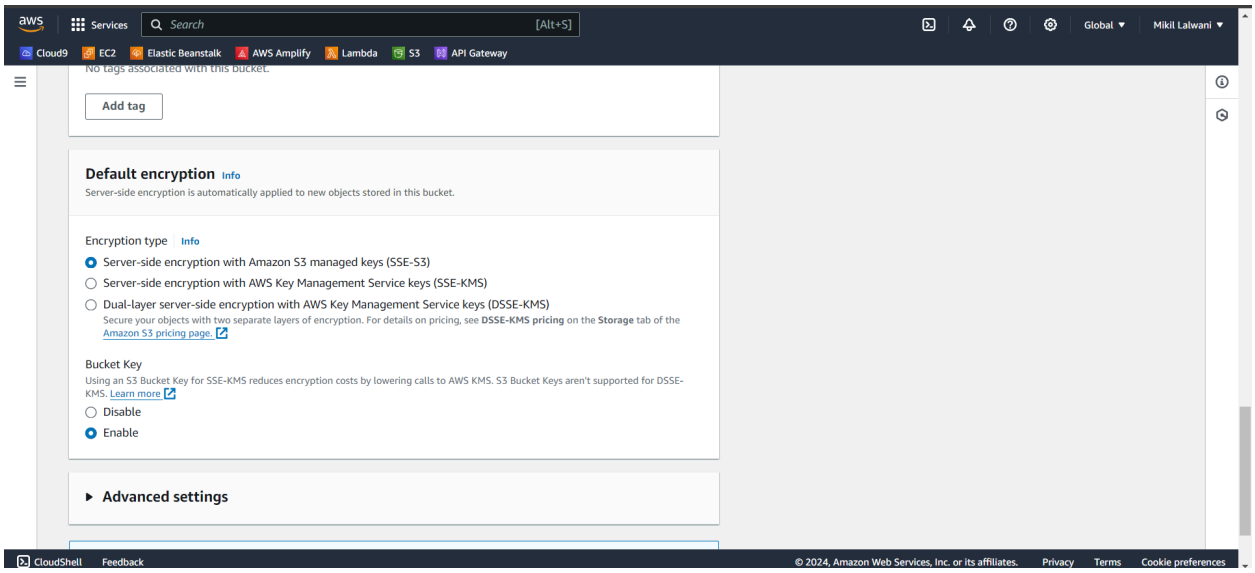


Step-5: Select the checkbox for Turning off block all public access might result in this bucket and the objects within becoming public

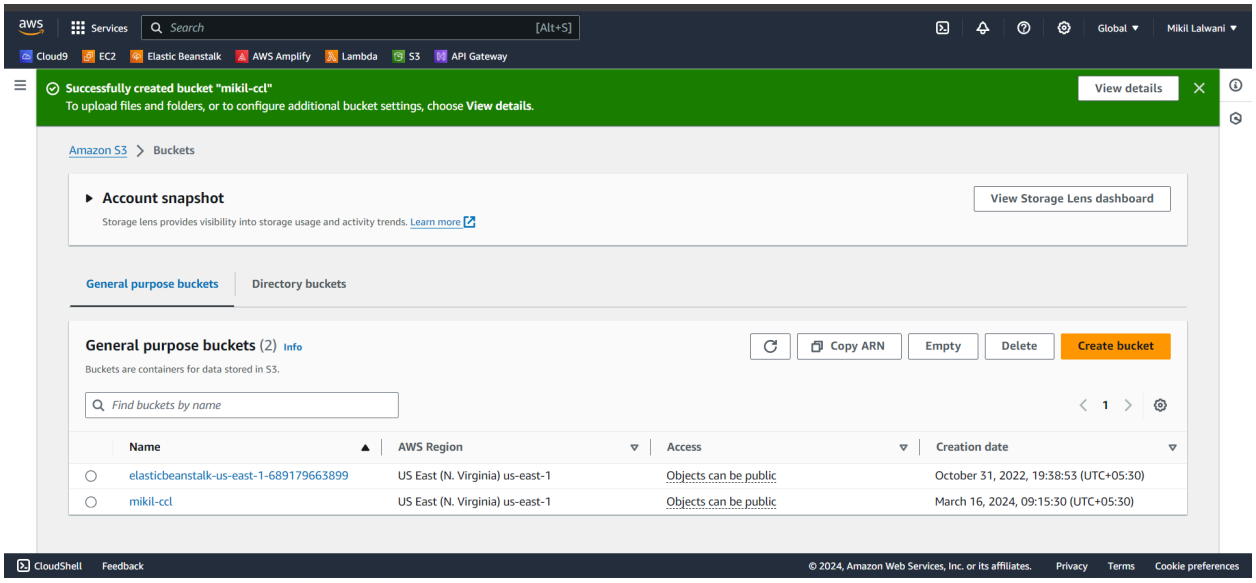
Step-6: Keep bucket versioning as disabled and add tags if required.



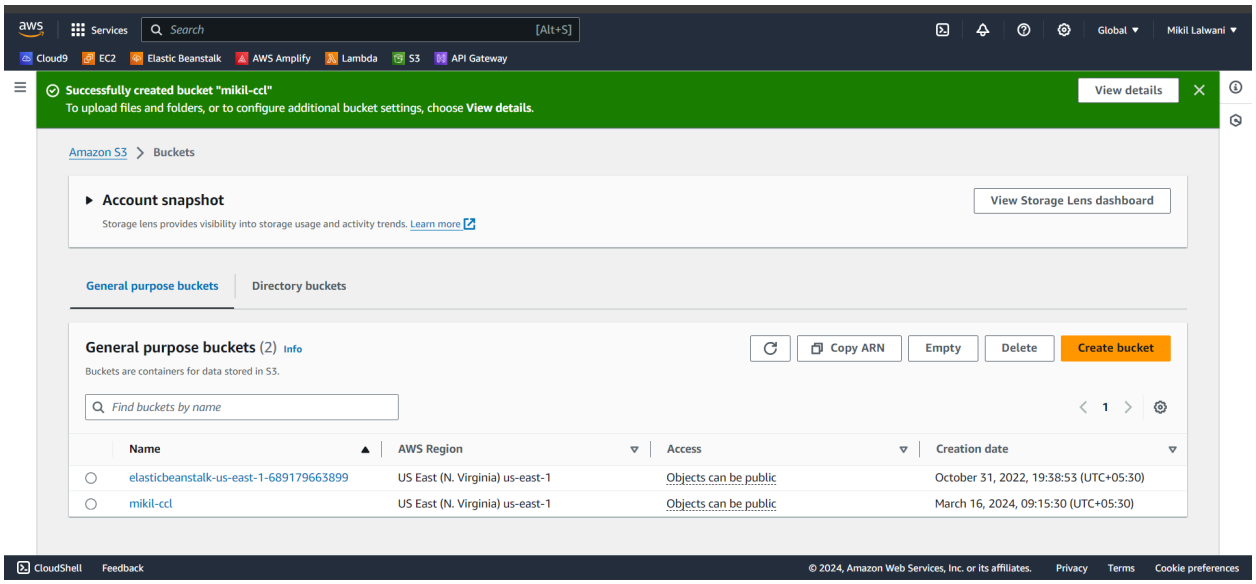
Step-7: Keep default encryption and click on create bucket button



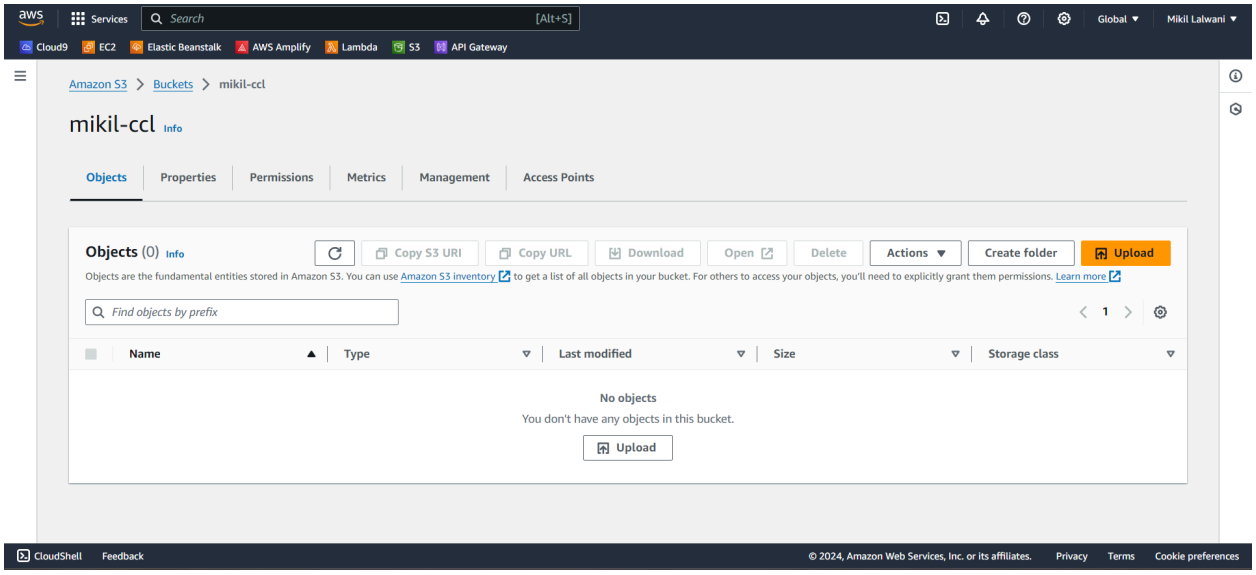
You can now see the successful creation of your bucket



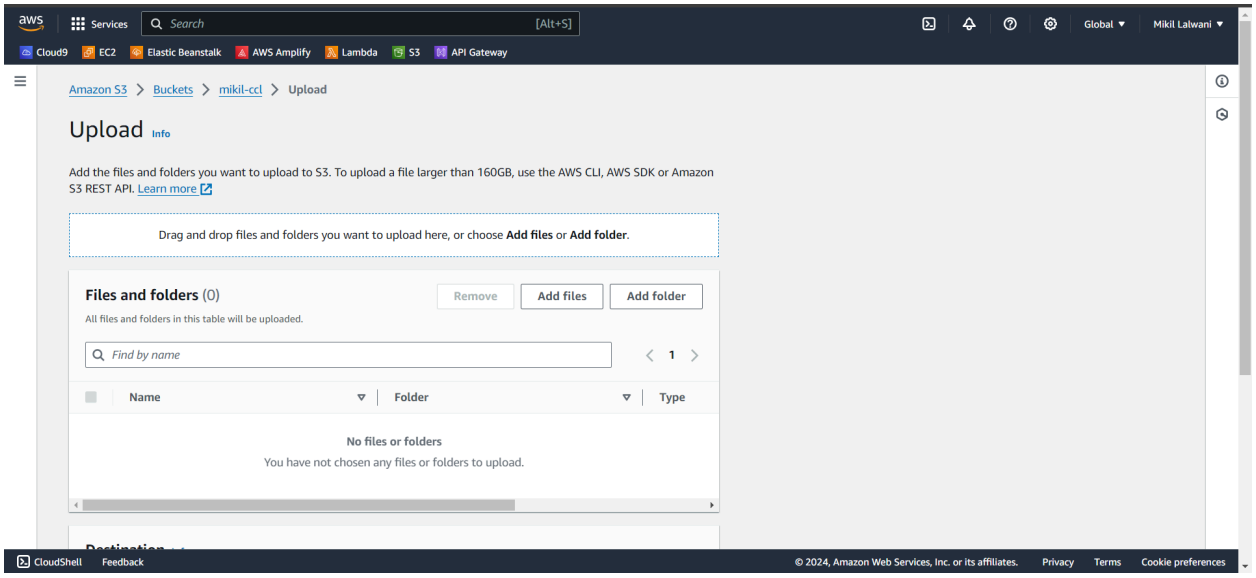
Step-8: Now click on the bucket that you have created.



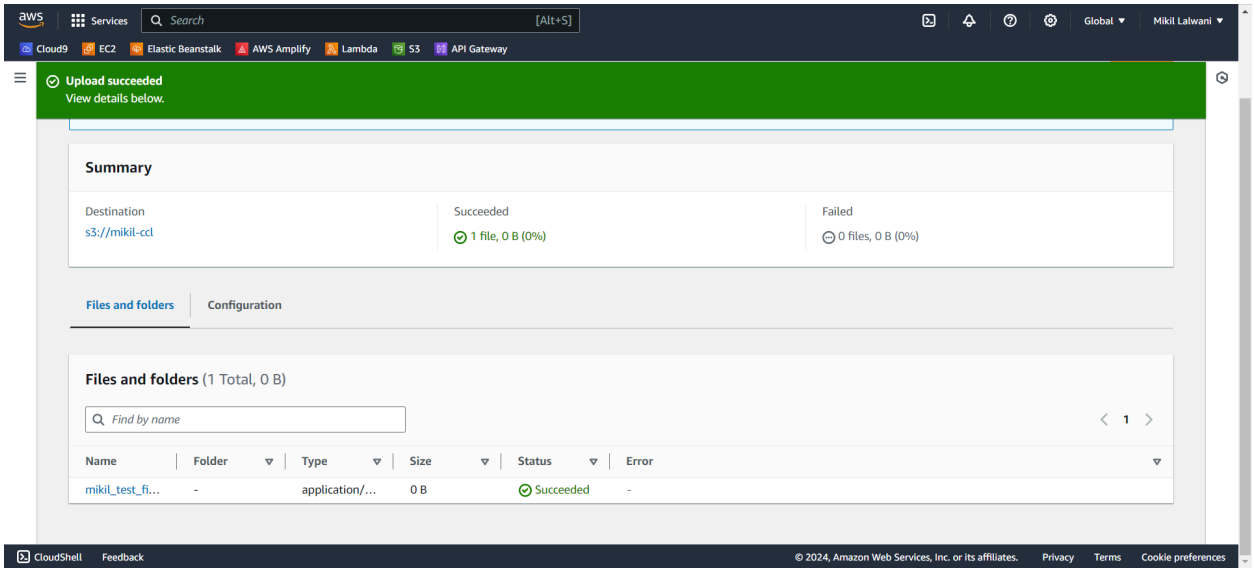
Step-9: You can either create a folder here or upload an existing file in the bucket



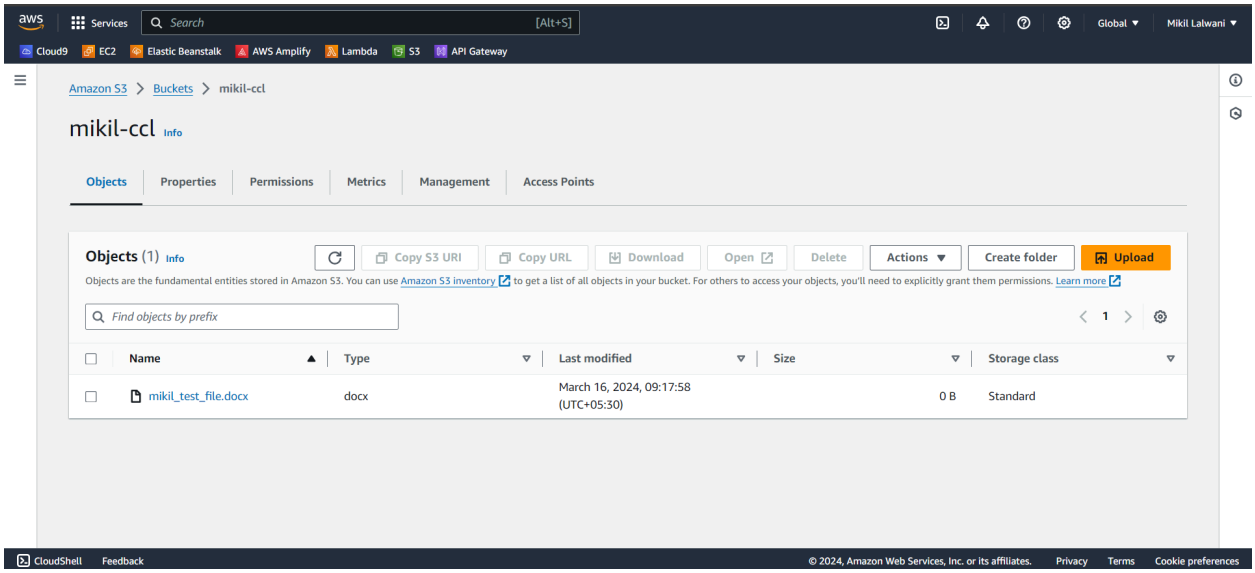
Step-10: now click on upload button and click on add files button browse your local machine and select which file you need to upload on S3 next click on upload button at bottom right end



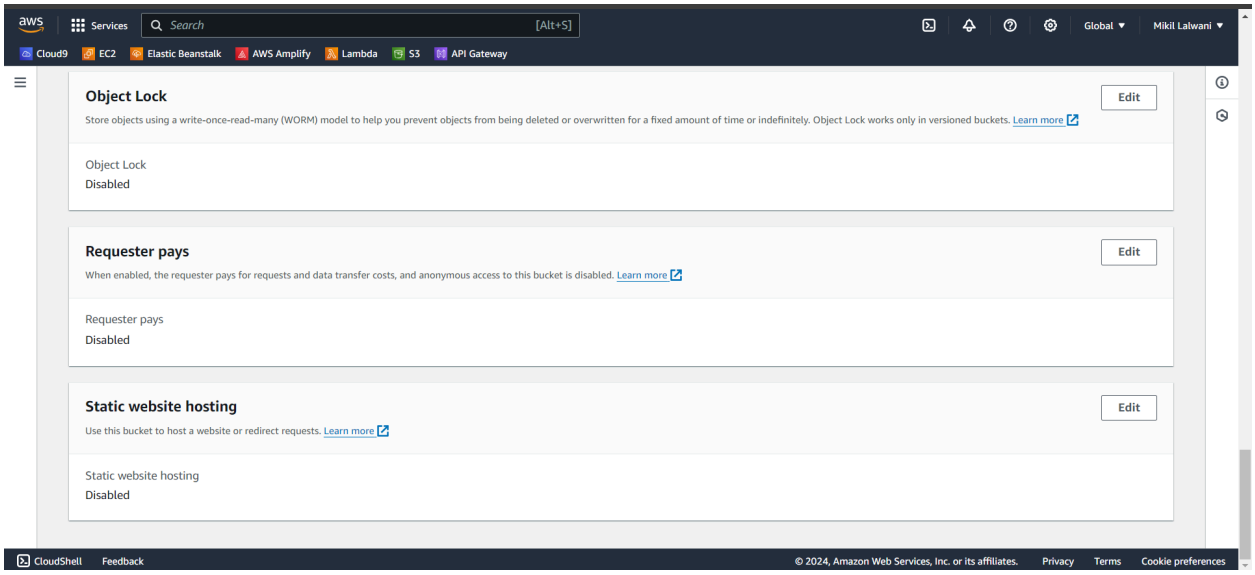
Now you can check the upload status screen



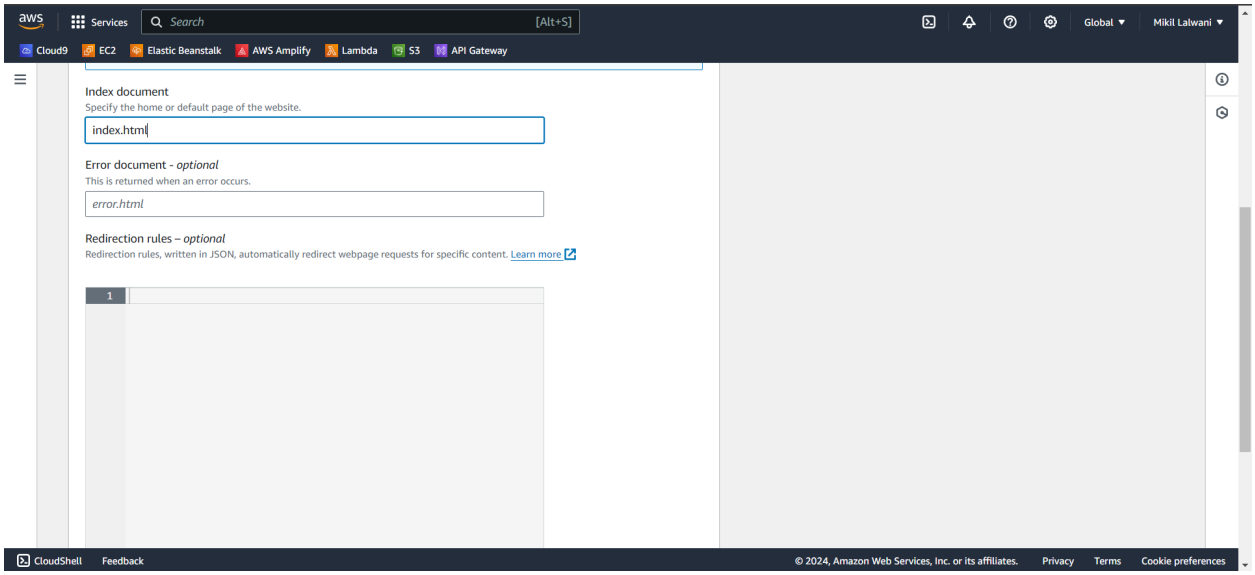
Now click on close button
The screen will appear as below

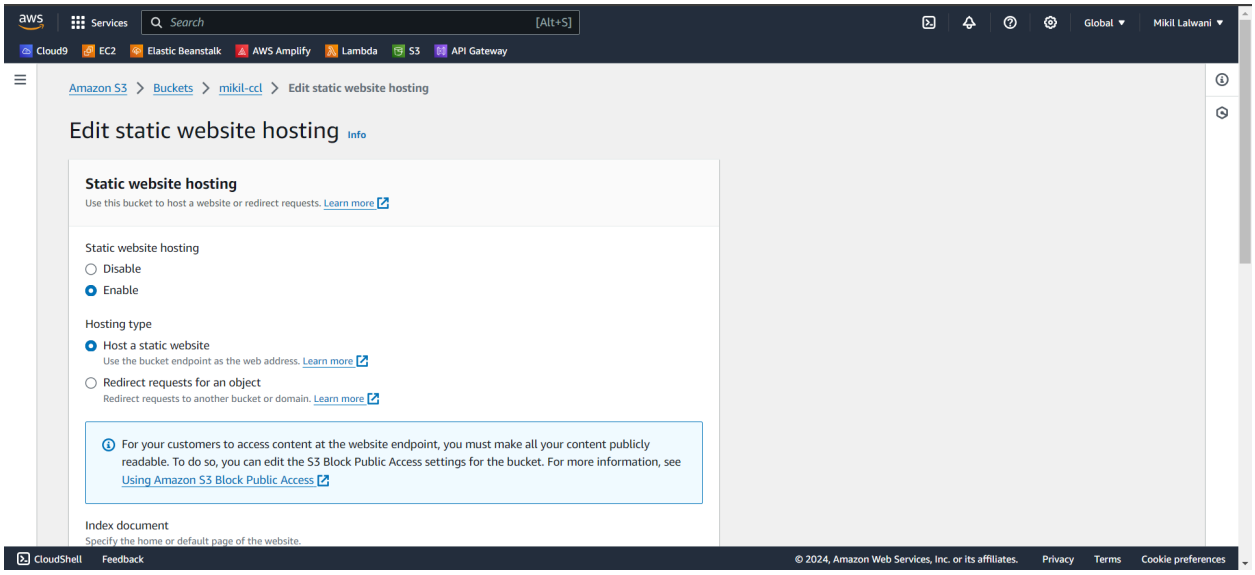


Step-11: Select properties and scroll down to Static website hosting option which is disabled now click on Edit option on right side

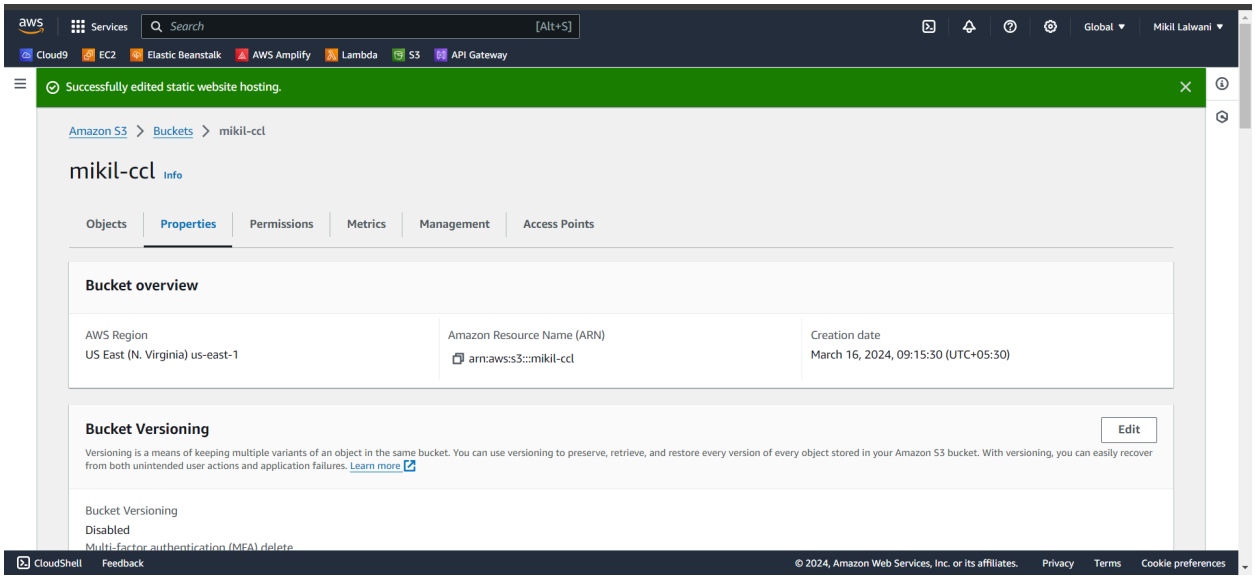


Step-12: Enable the radio button and specify the file name in Index document which you have added in S3

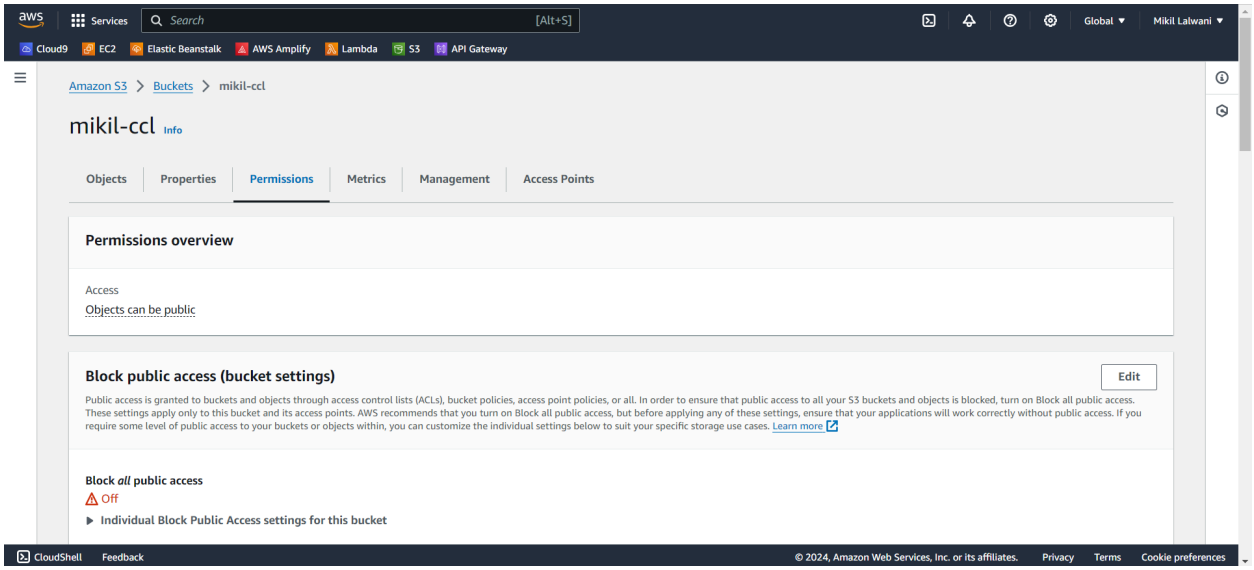




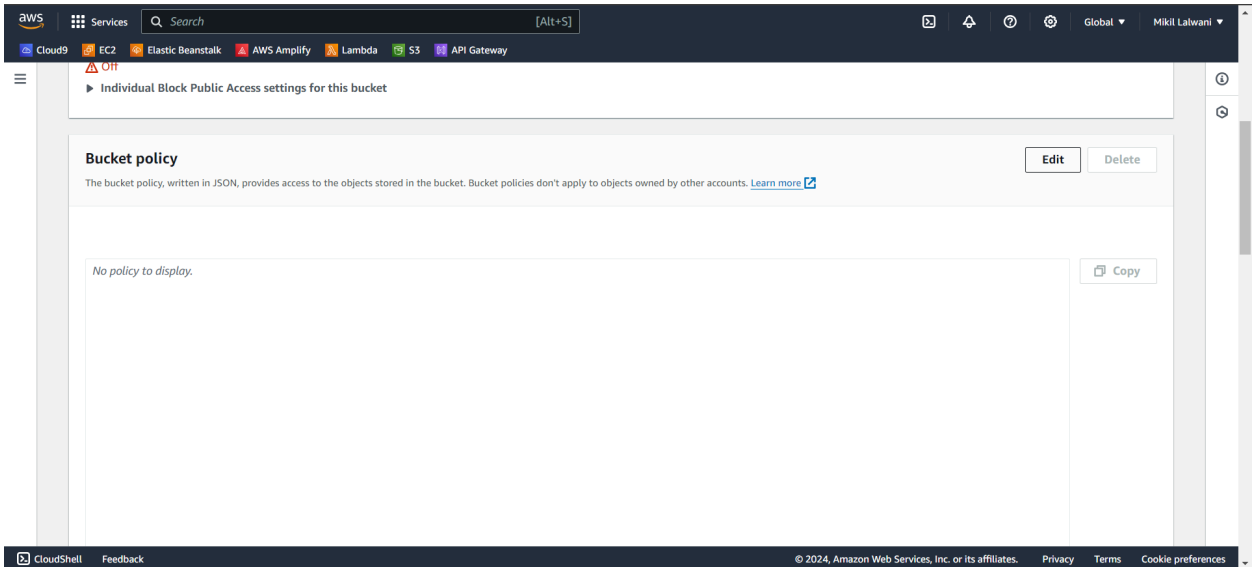
Scroll down and save the changes at bottom right
Following screen will appear



Step-13: Click on Permissions Tab



Step-14: In bucket policy click on Edit option

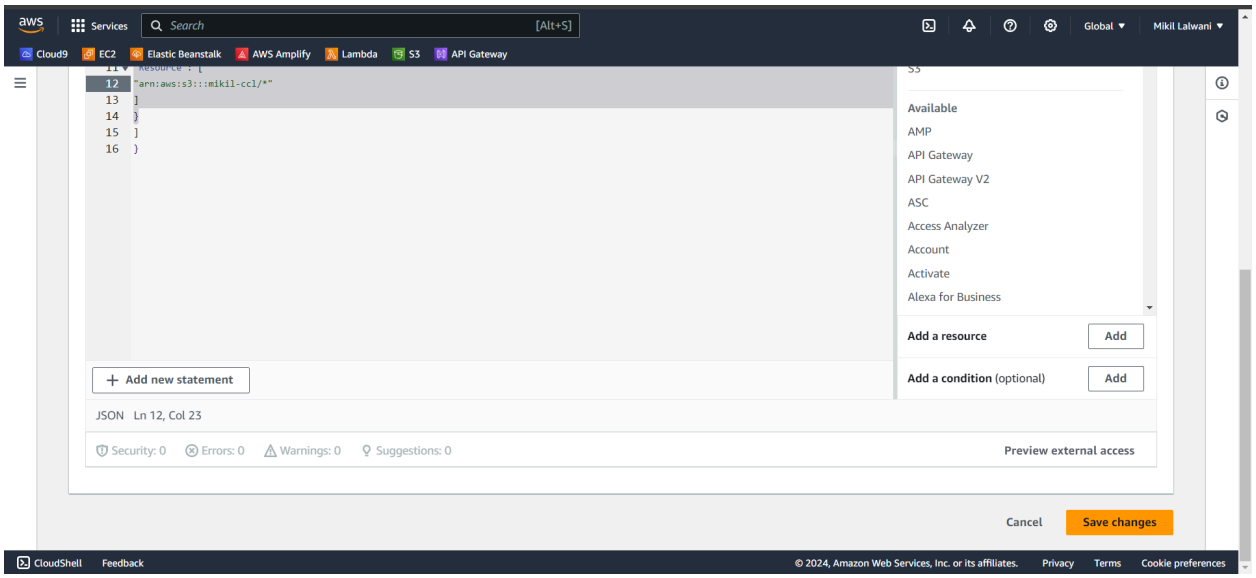


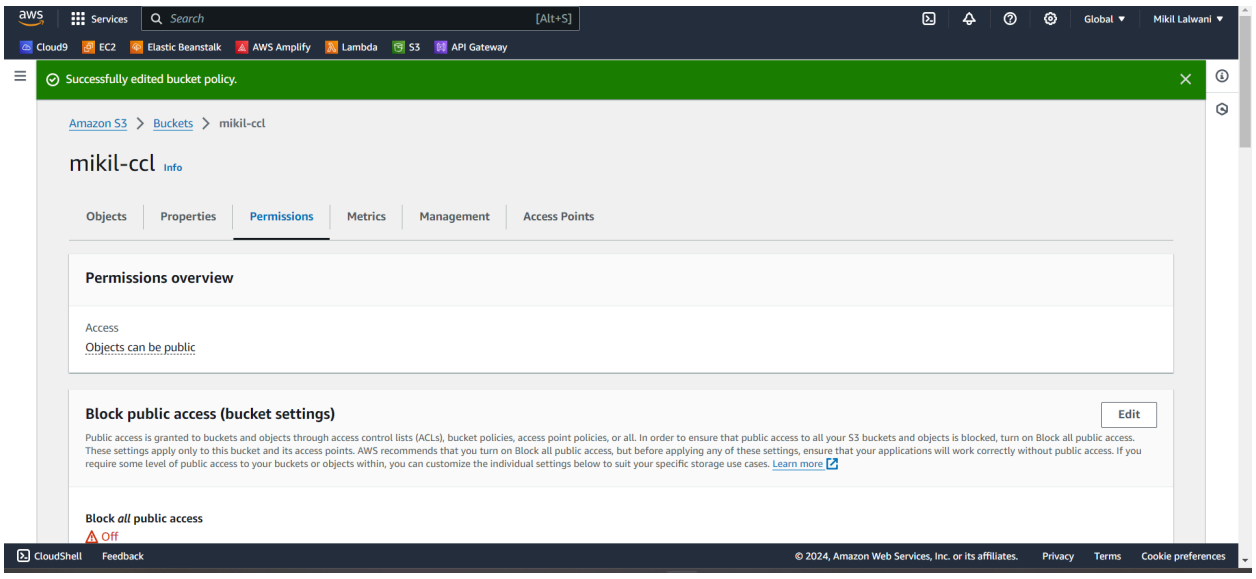
Step 15- after clicking on edit button paste the following code in bucket policy

```
{
	"Version": "2012-10-17",
	"Statement": [
	{
		"Sid": "PublicReadGetObject",
		"Effect": "Allow",
		"Principal": "*",
		"Action": [
			"s3:GetObject"
```

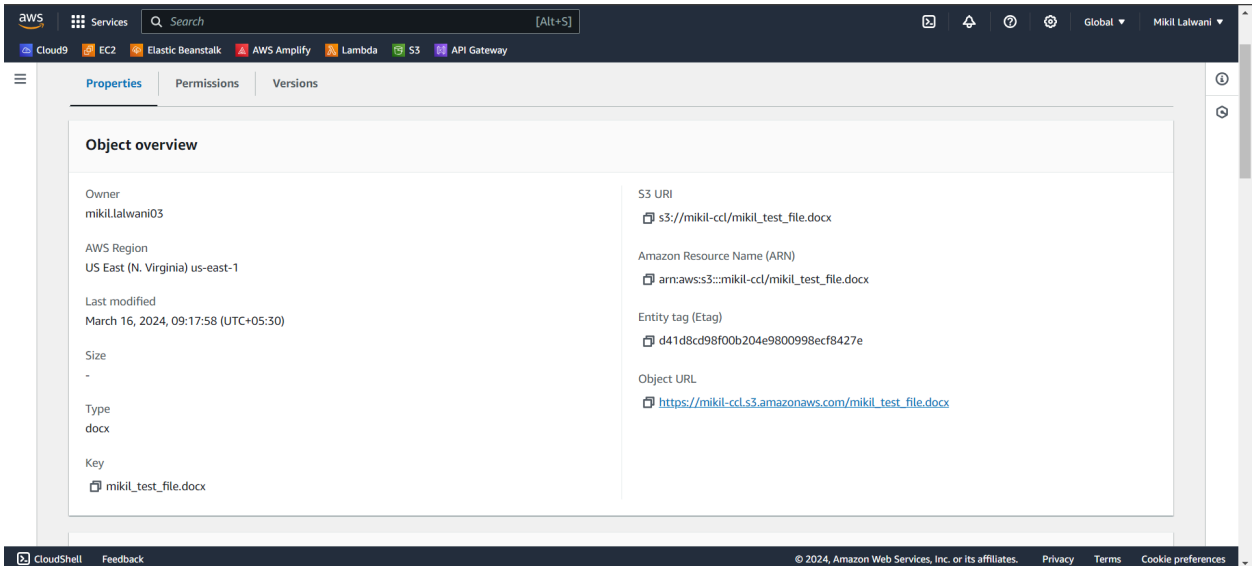
```
            ],
            "Resource": [
                "arn:aws:s3:::Bucket-Name/*"
            ]
        }
        ]
}
```

Note-Make sure that you add your bucket name in the code above
Scroll down and click on Save Changes button

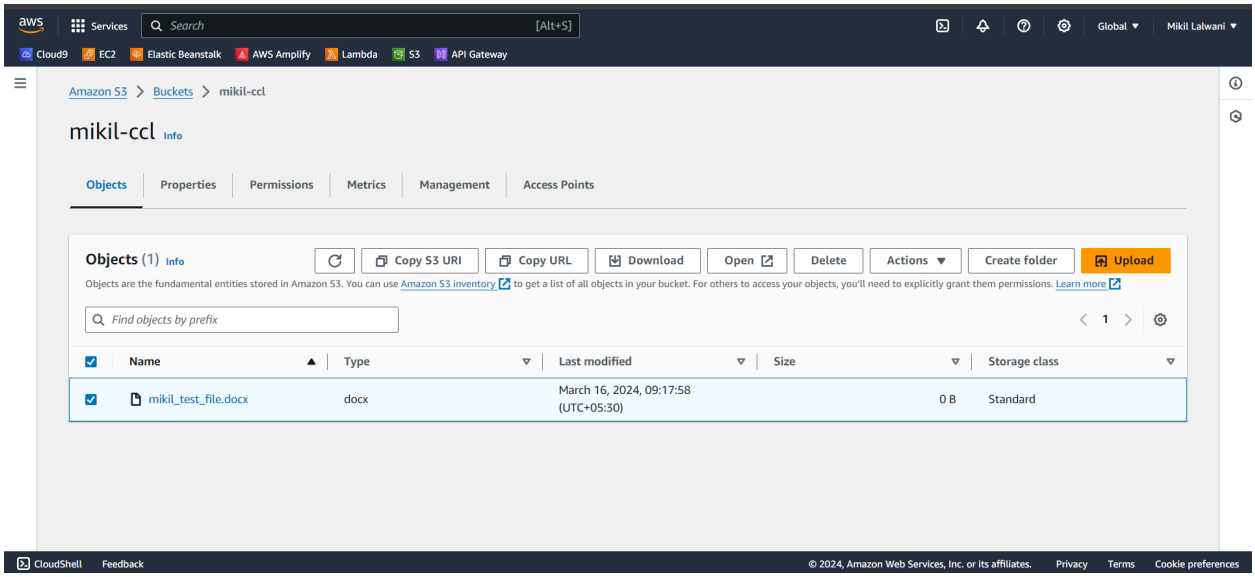




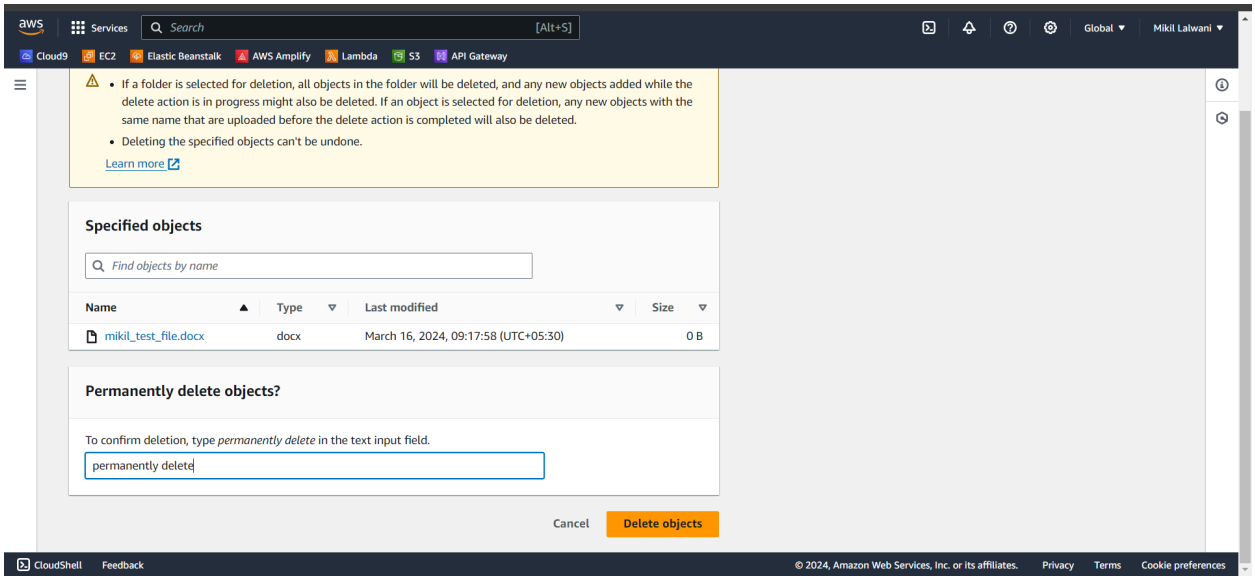
Step-16: open your html file and click on Object URL



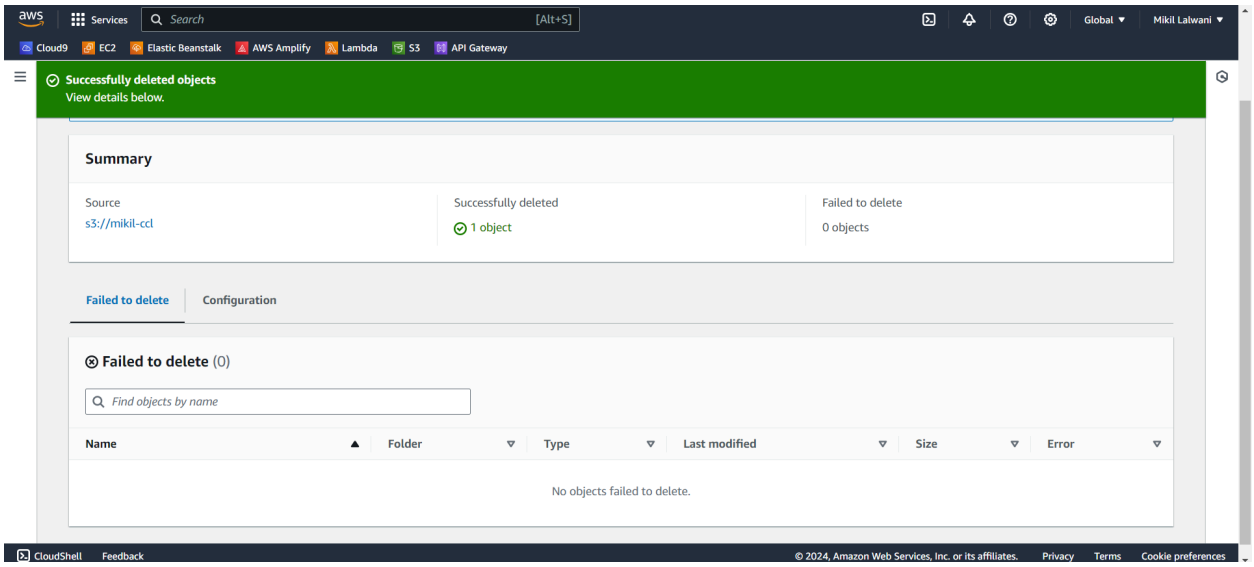
Step-17: Now for delete files click on checkbox of your file and then click on Delete Button



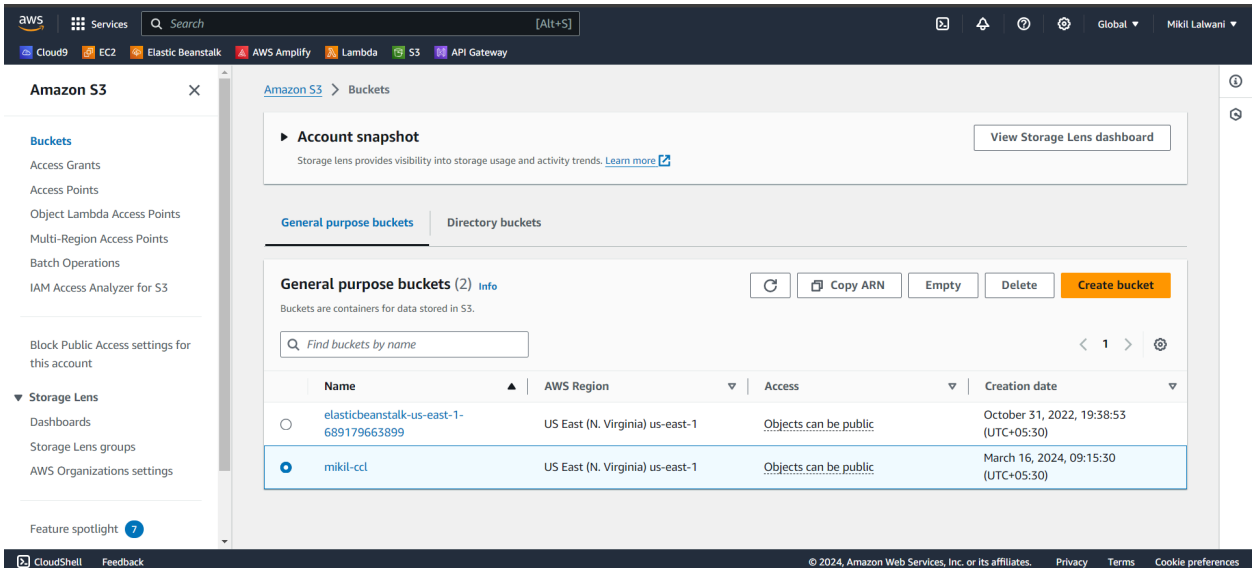
Write permanently delete and click on delete object button



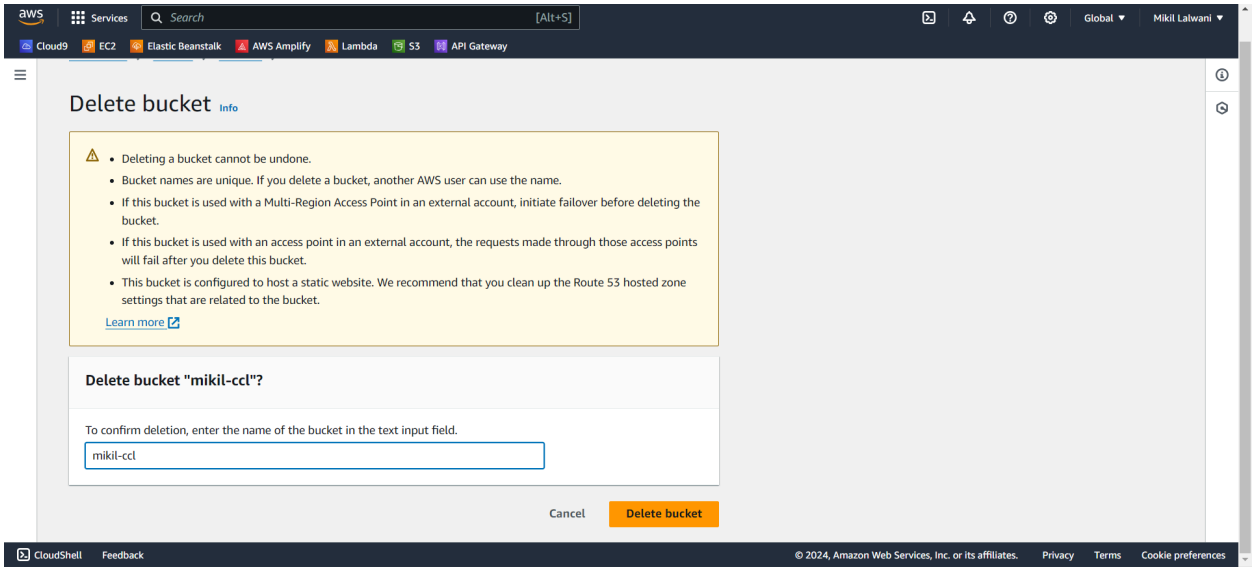
Now click on close button



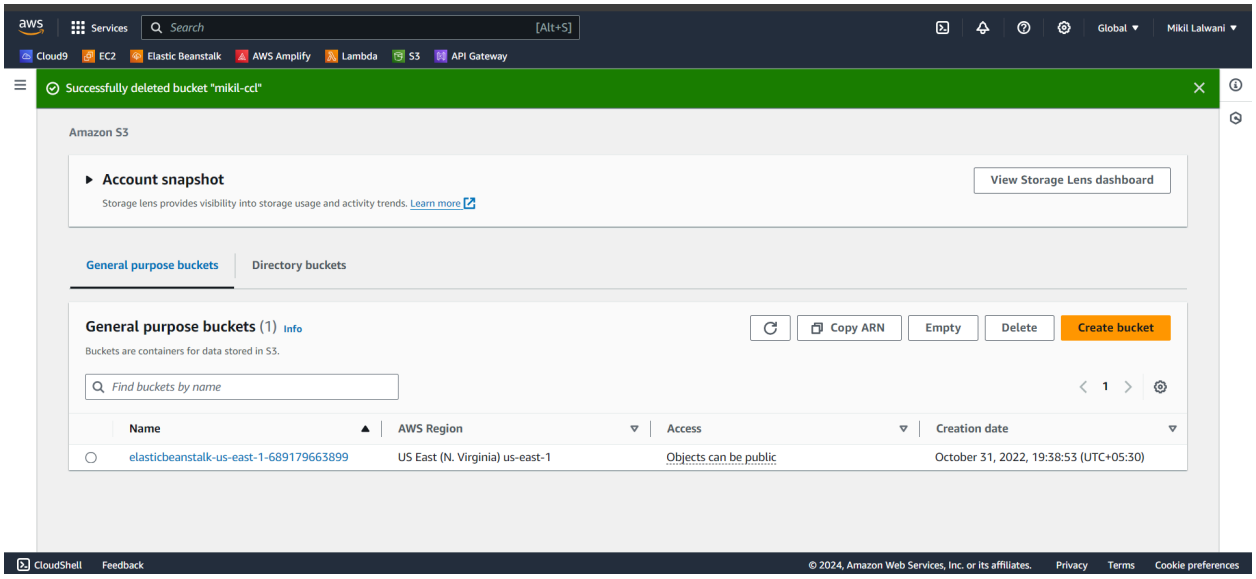
Step-18: now come to Amazon S3 tab and select your bucket and then click on delete button



Write down your bucket name in delete bucket tab and click on delete button at bottom right



You can see that the bucket is deleted



Conclusion -

Thus we have successfully studied and implemented Storage as a Service using AWS.