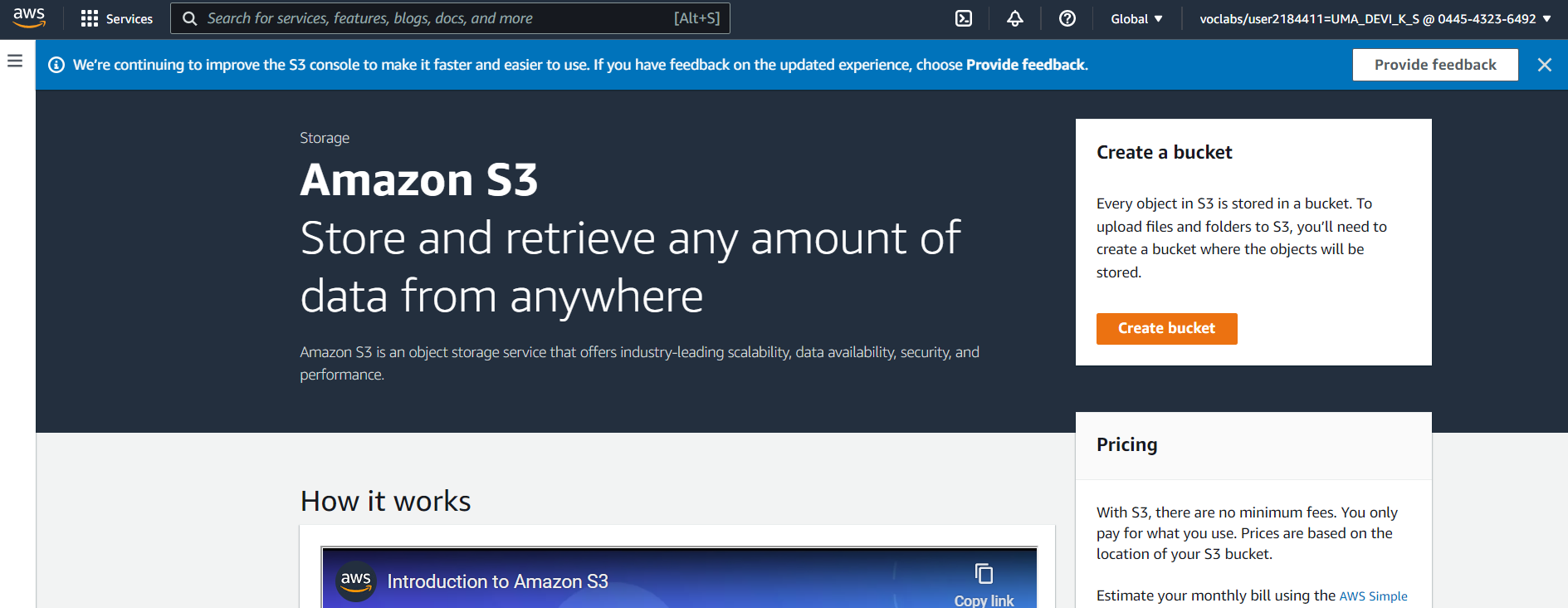
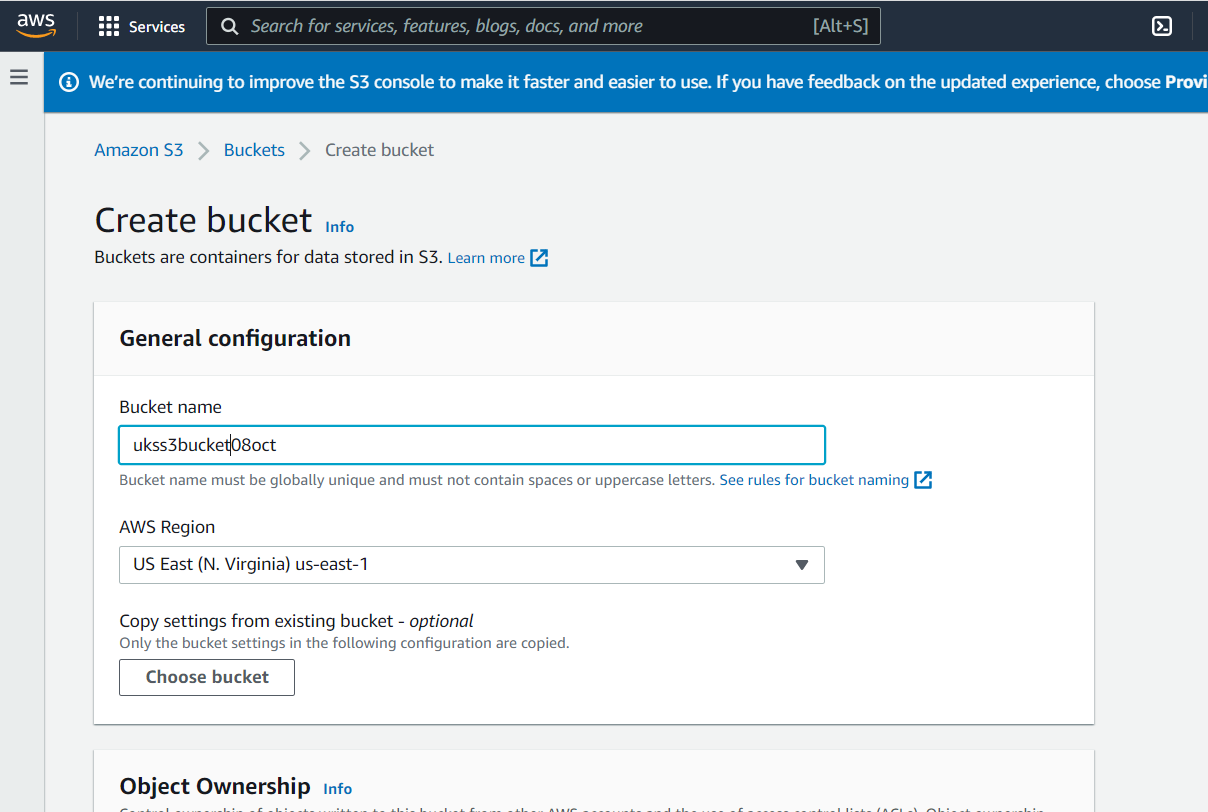
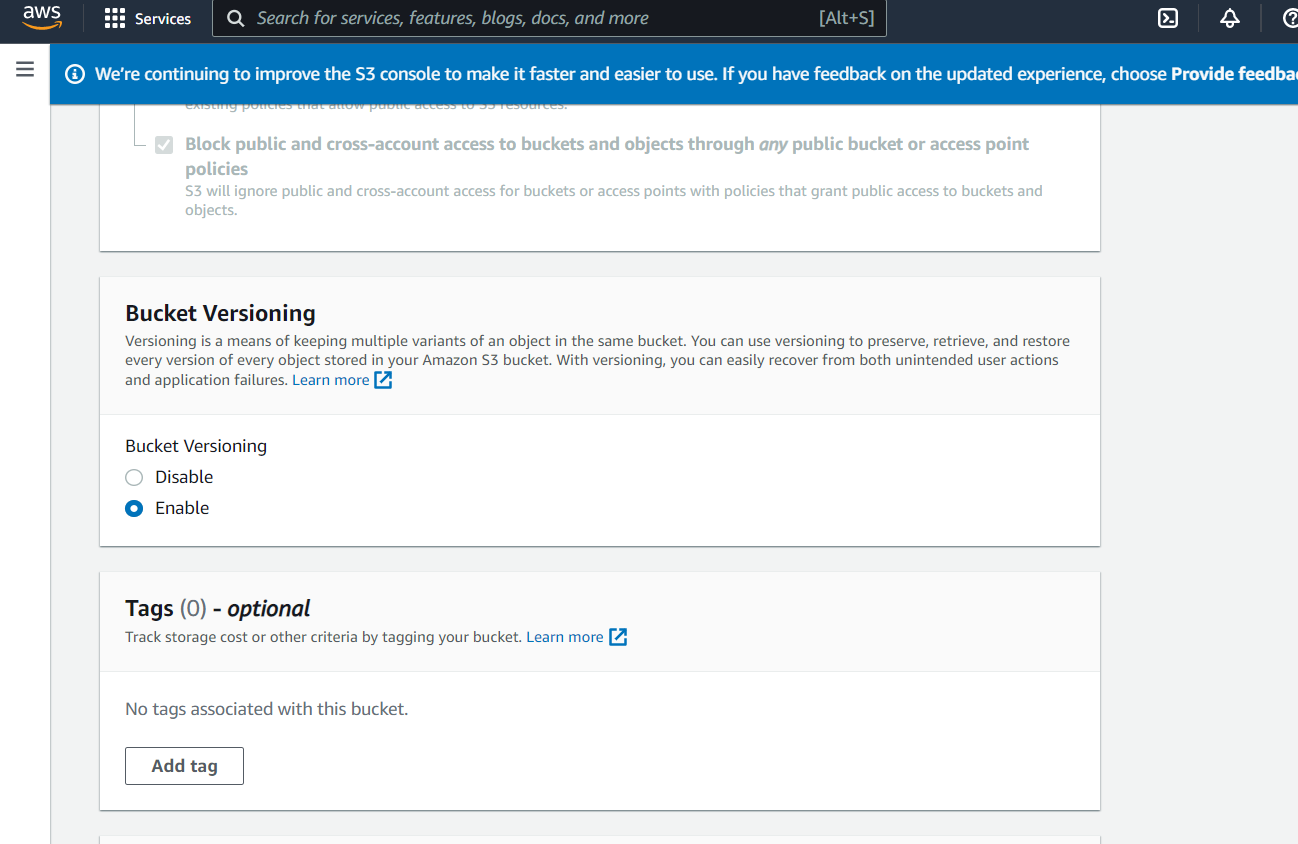
**Assignment 5:**

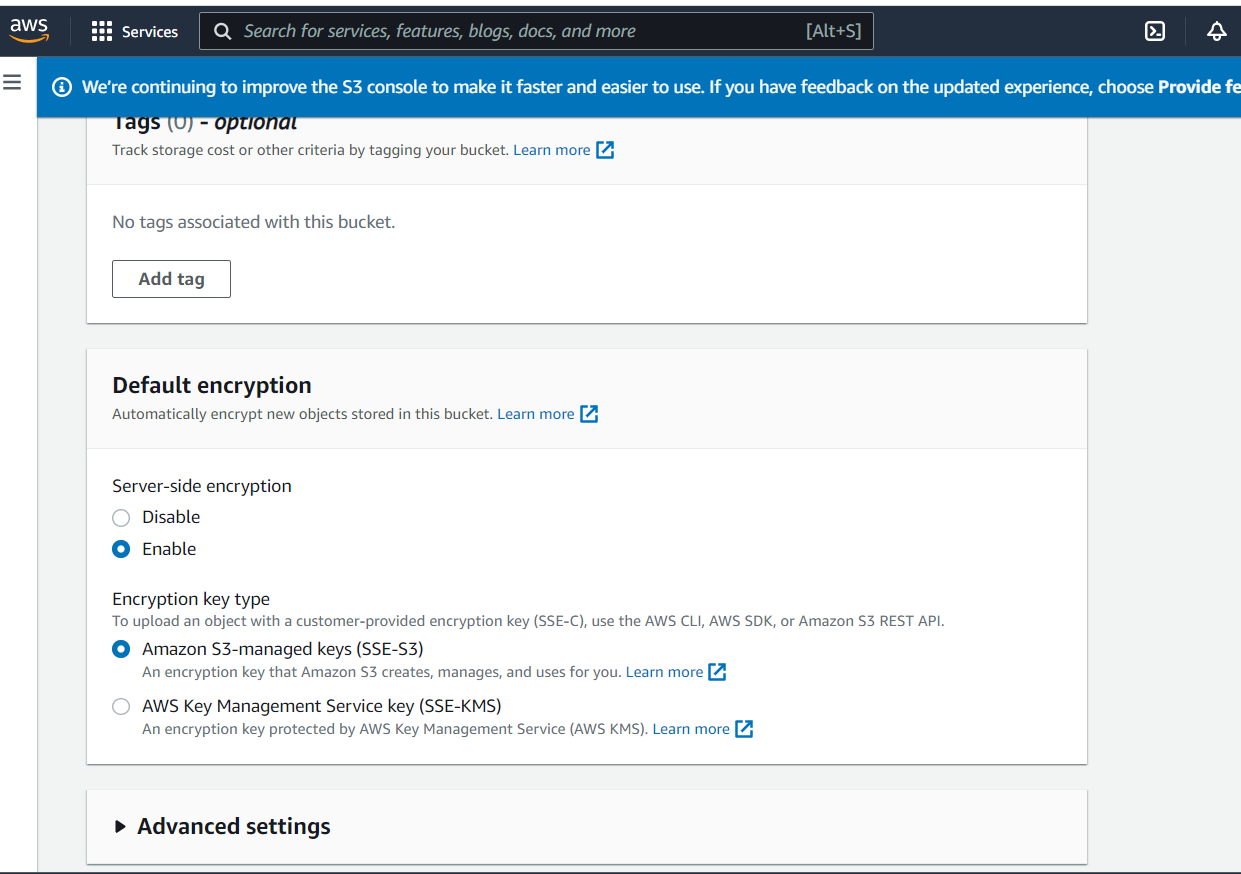
Create S3 bucket with static website hosting property enabled using cloud formation automation template

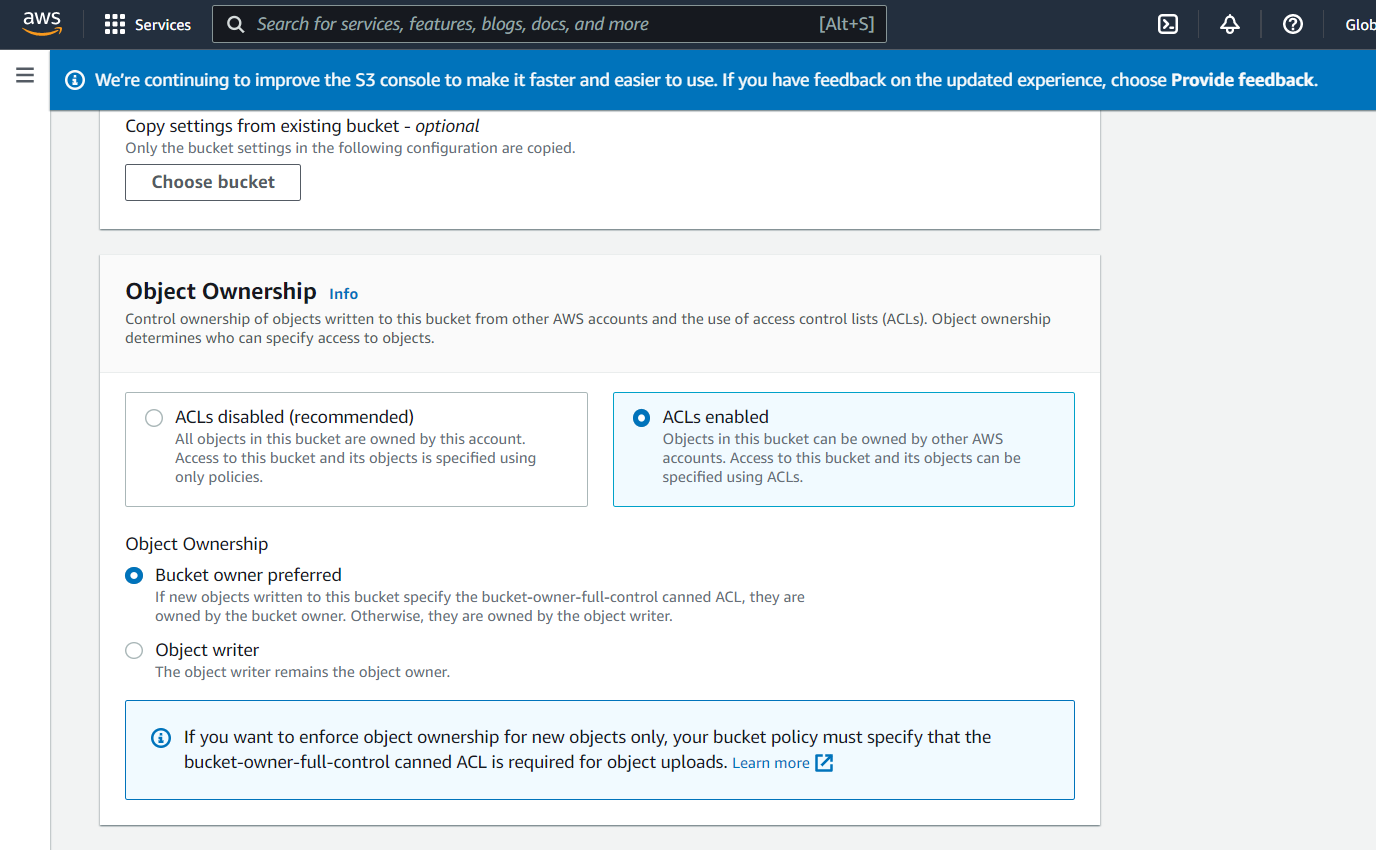
Step 1: Create S3 bucket with static website host. Below are the screenshots for the S3 bucket creation.



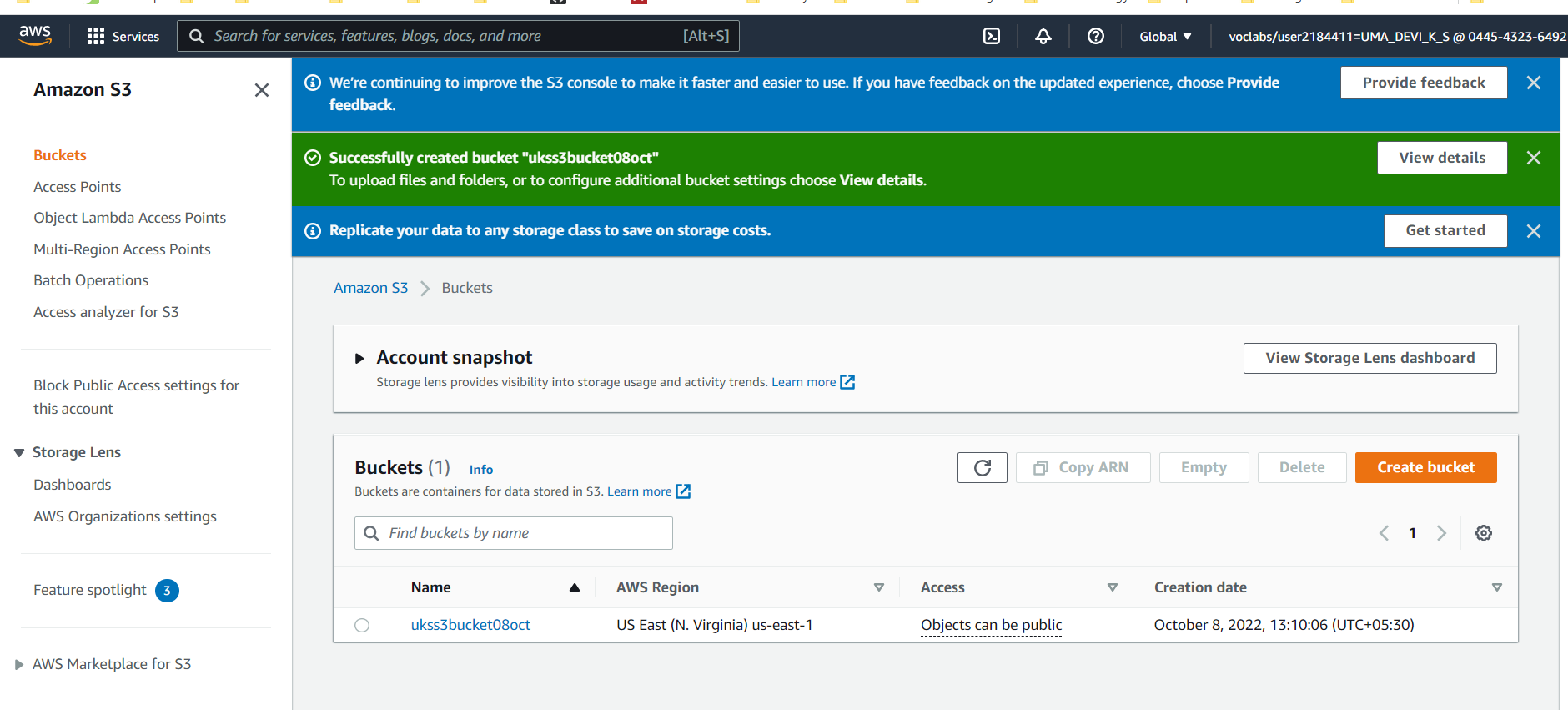


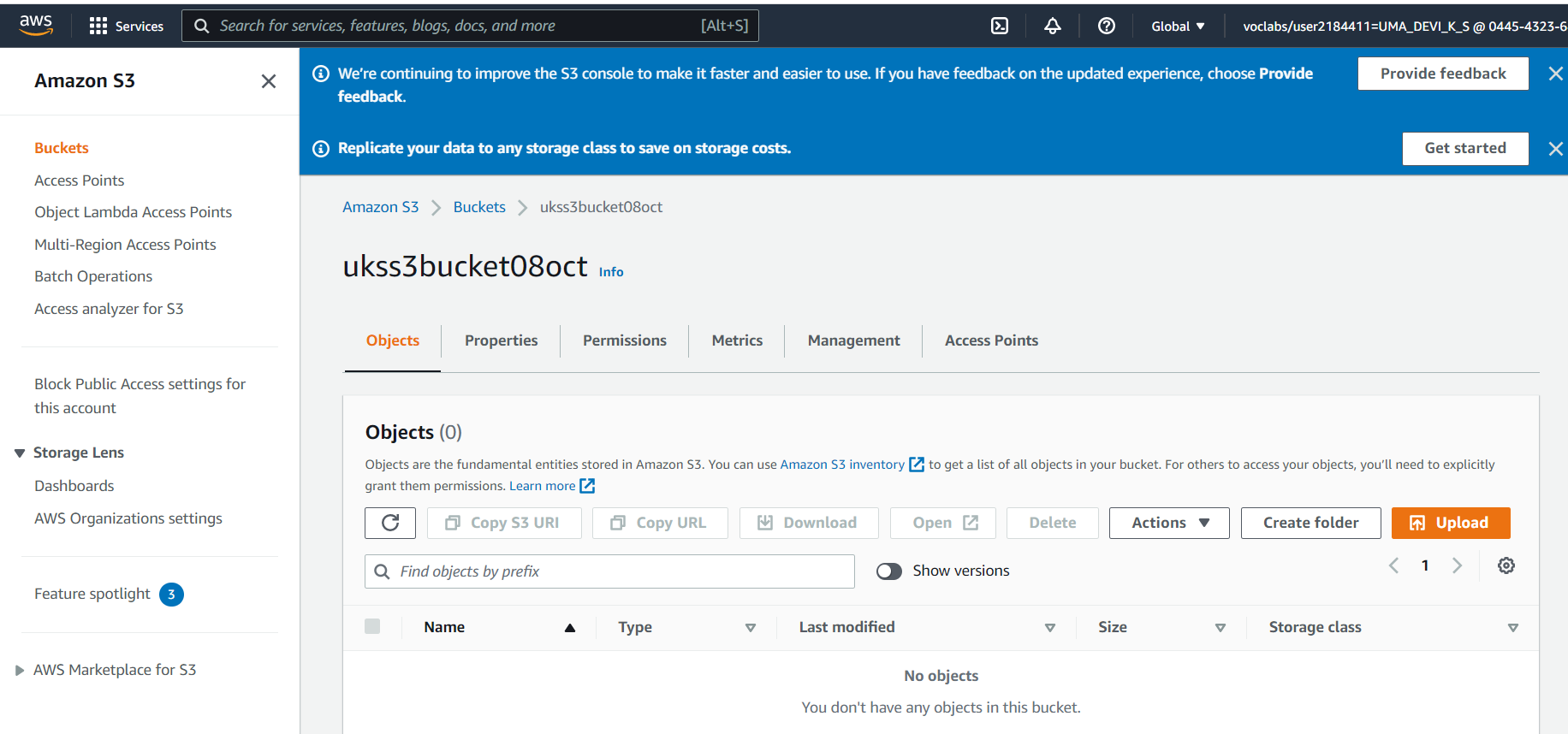






**S3 bucket create successfully.**

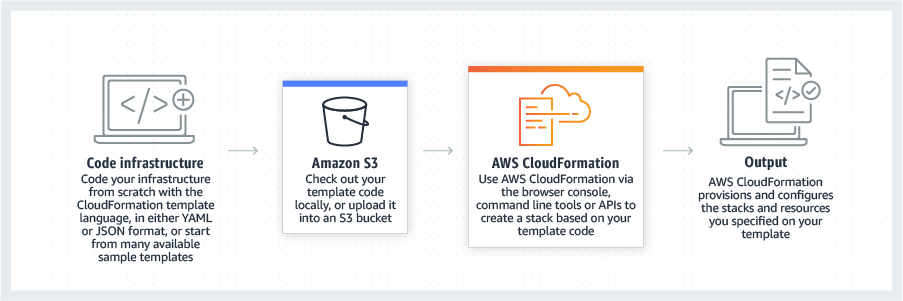




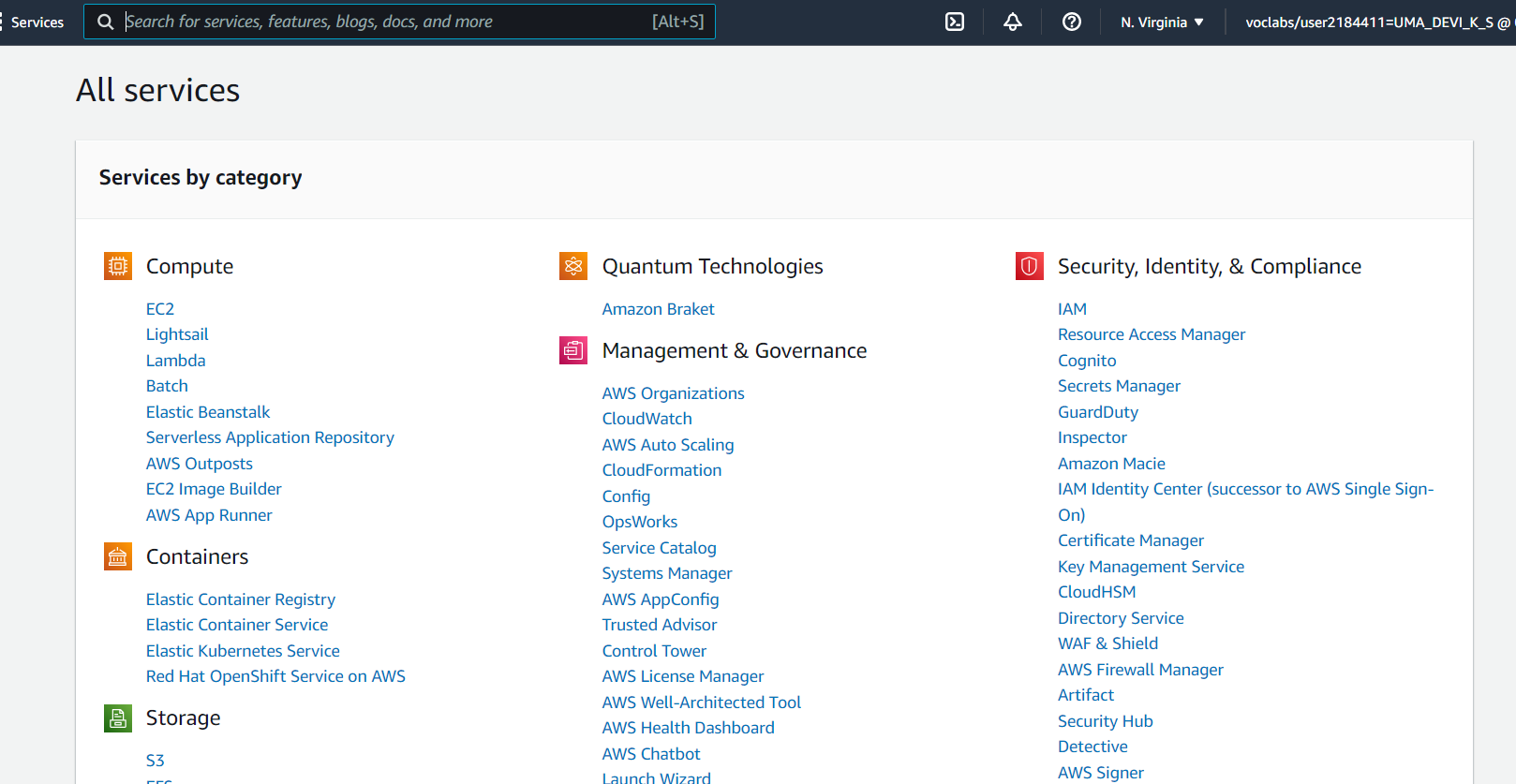
**We can enable server access logs for the reference. I have not attached screenshot for the enabling logs.**

**AWS CloudFormation:** Speed up cloud provisioning with **infrastructure as code**

* Scale your infrastructure worldwide and manage resources across all AWS accounts and regions through a single operation.
* Extend and manage your infrastructure to include cloud resources published in the CloudFormation Registry, the developer community, and your library.
* Automate resource management across your organization with AWS service integrations offering turnkey application distribution and governance controls.



**Step1 : We will use cloudFormation service from Management & Governance and create S3 bucket.**





We will prepare template to perform S3 creation using cloud formation. This requires us to prepare a template. This template can be prepared using JSON or YAML formats.

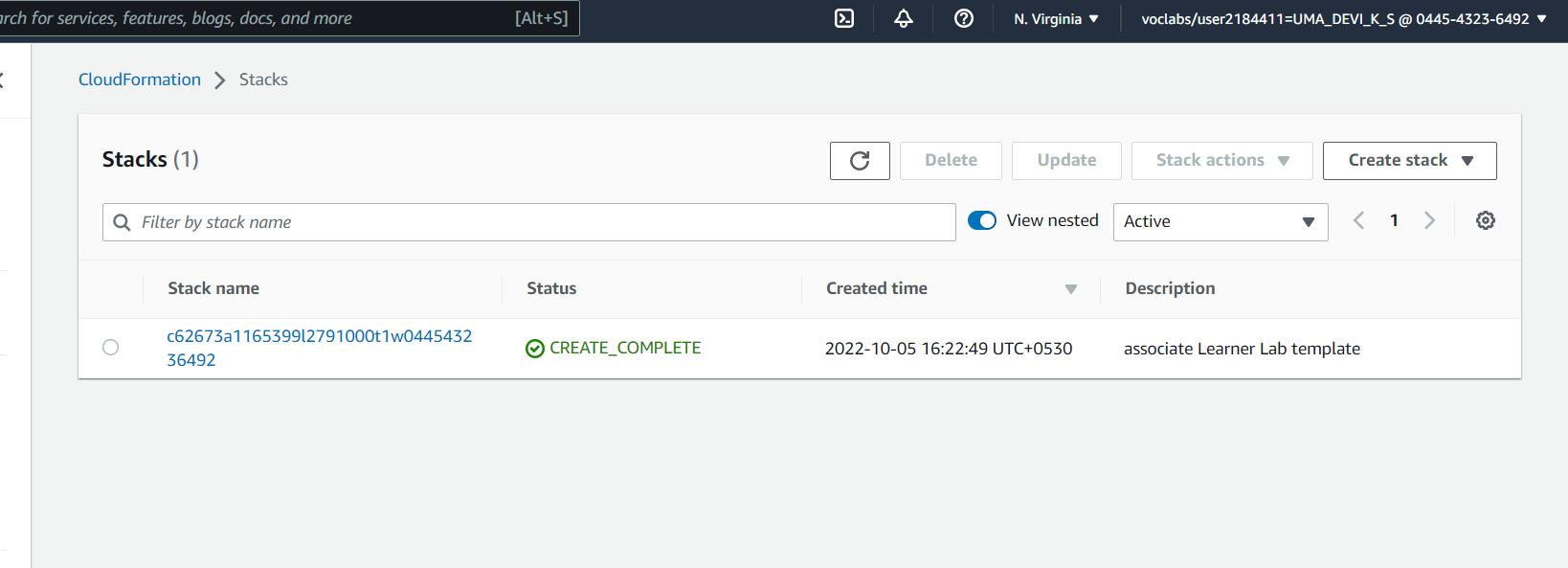
Example:

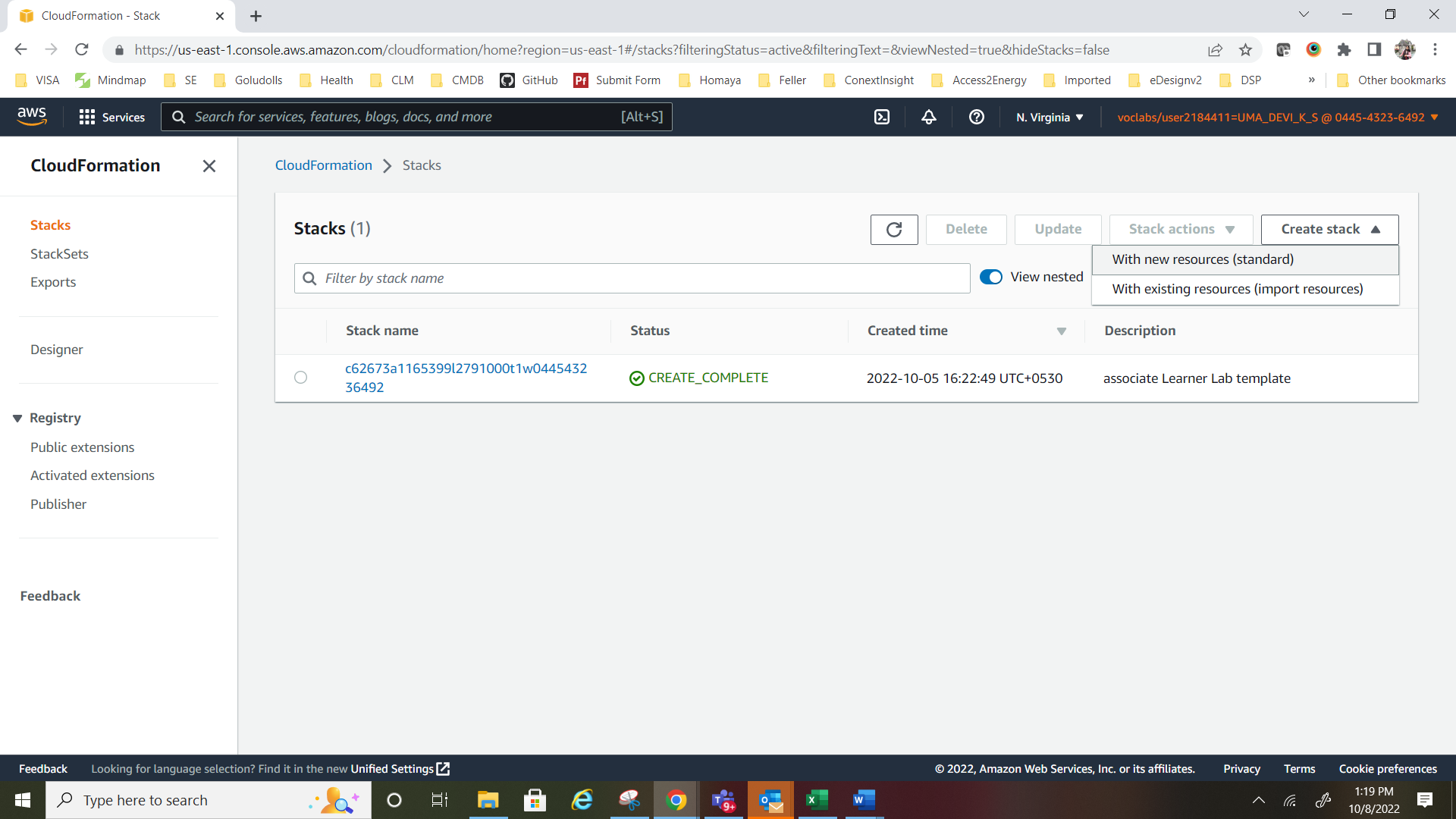
|  |  |
| --- | --- |
| YAML format | JSON format |
| AWSTemplateFormatVersion: '2022-10-09'  Description: Template to create s3 bucket and policy  Parameters:  BucketName:  Type: String  Description: Bucket Name  Default: sandeepm-bucket  Resources:  MyS3Bucket:  Type: 'AWS::S3::Bucket'  Description: Bucket on which we will attach and test bucket policy  Properties:  BucketName: !Ref BucketName    MyS3BucketPolicy:  Type: AWS::S3::BucketPolicy  Properties:  Bucket: !Ref MyS3Bucket  PolicyDocument:  Statement:  -  Action:  - s3:\*  Effect: Allow  Resource:  - !Sub arn:aws:s3:::${MyS3Bucket}  - !Sub arn:aws:s3:::${MyS3Bucket}/\*  Principal:  AWS:  - '\*' | {  "AWSTemplateFormatVersion": "2010-09-09",  "Description": "Template to create s3 bucket and policy",  "Parameters": {  "BucketName": {  "Type": "String",  "Description": "Bucket Name",  "Default": " sandeepm-bucket"  }  },  "Resources": {  "MyS3Bucket": {  "Type": "AWS::S3::Bucket",  "Description": "Bucket on which we will attach and test bucket policy",  "Properties": {  "BucketName": {  "Ref": "BucketName"  }  }  },  "MyS3BucketPolicy": {  "Type": "AWS::S3::BucketPolicy",  "Properties": {  "Bucket": {  "Ref": "MyS3Bucket"  },  "PolicyDocument": {  "Statement": [  {  "Action": [  "s3:\*"  ],  "Effect": "Allow",  "Resource": [  {  "Fn::Sub": "arn:aws:s3:::${MyS3Bucket}"  },  {  "Fn::Sub": "arn:aws:s3:::${MyS3Bucket}/\*"  }  ],  "Principal": {  "AWS": [  "\*"  ]  }  }  ]  }  }  }  }  } |

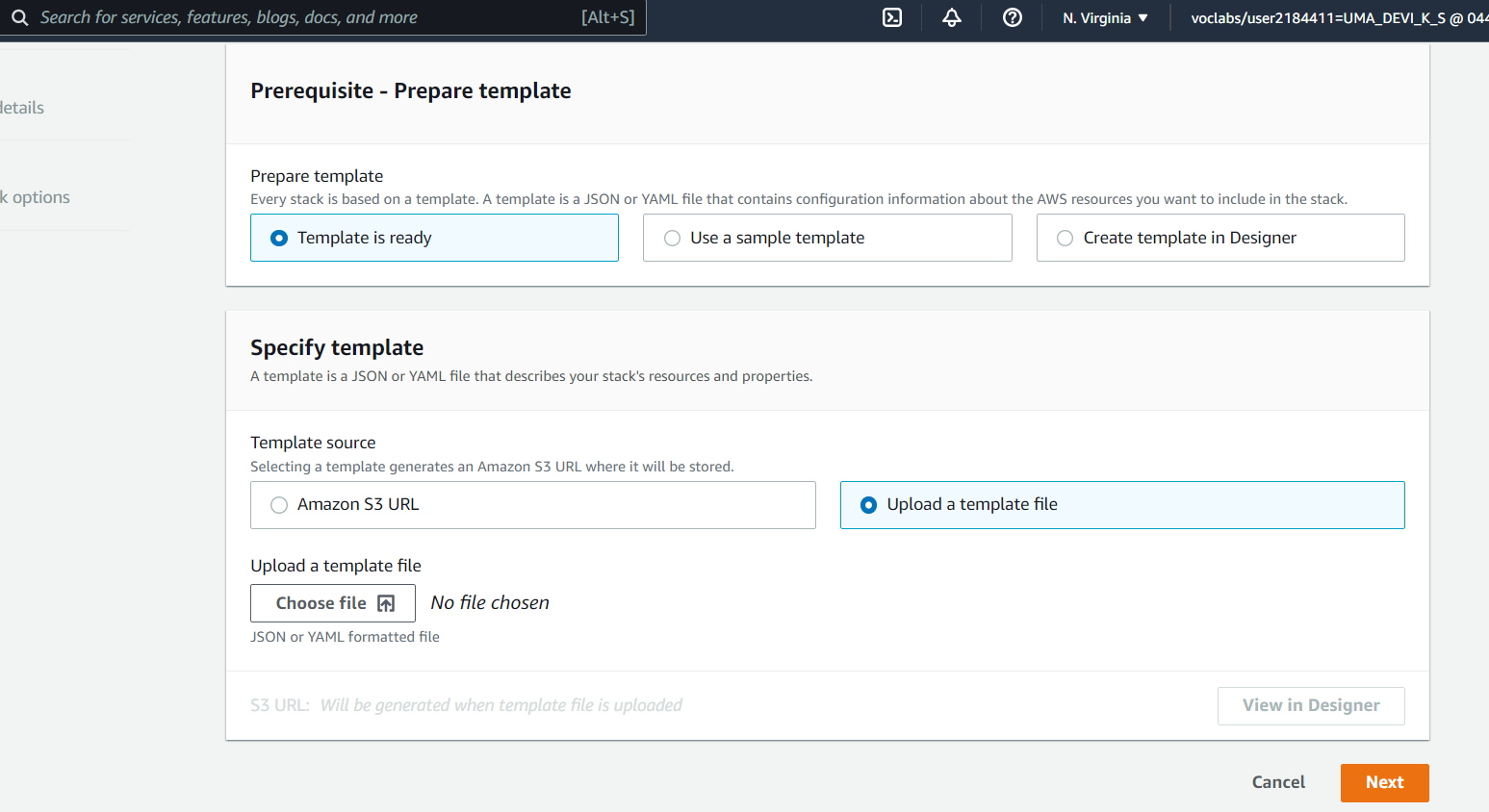
Step : create Stack.

* Click on “Upload a template file”, upload **sandeepm\_bucket\_CF.yml** and click Next
* Enter the stack name and click on Next.
* In the configuration, keep everything as default and click on Next.
* In the events tab of the stack, you can view the status.
* Once the stack is successfully created, head to your S3 bucket and you will see your bucket.

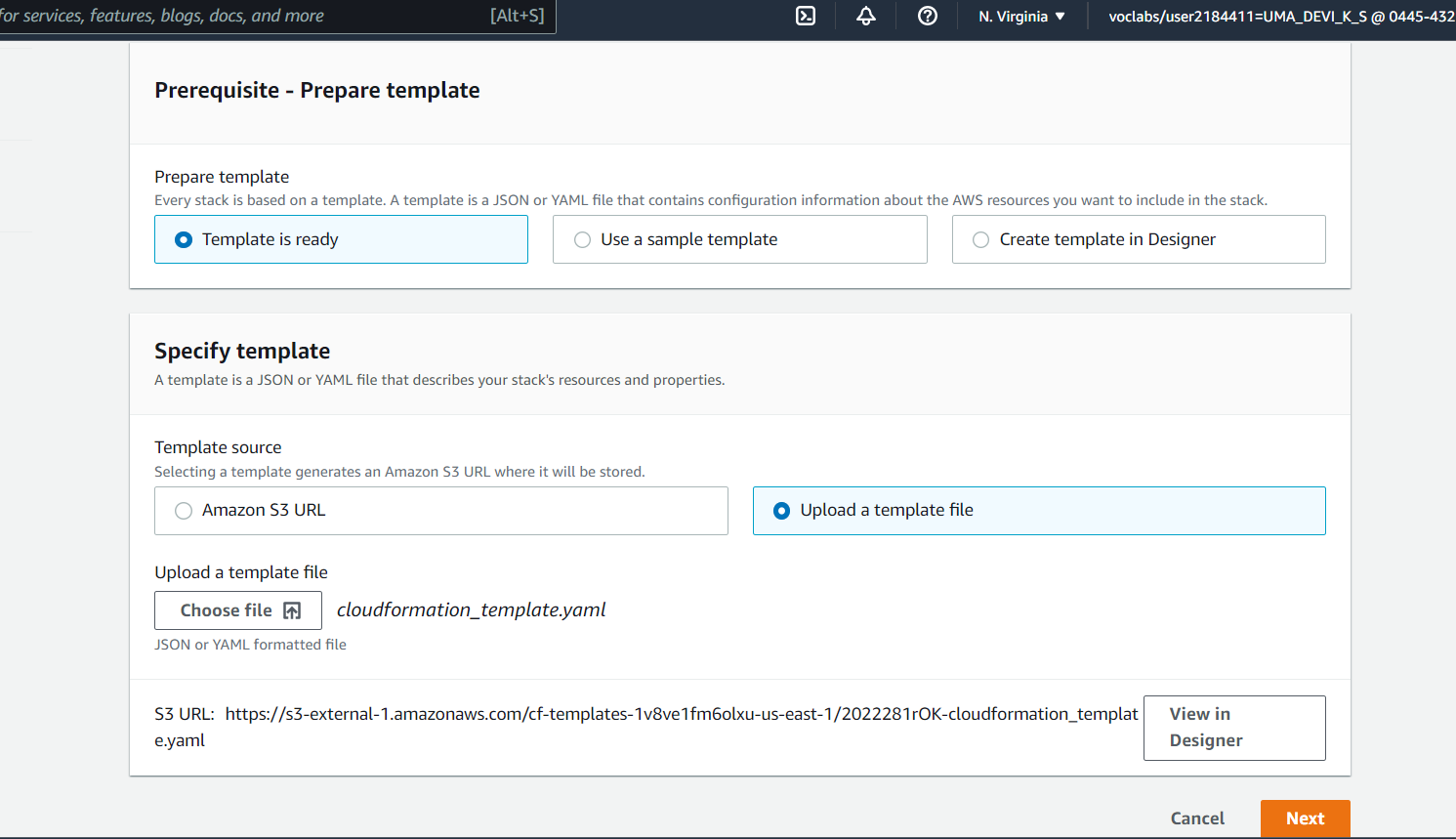
Below are the screenshots for the steps. Once all the steps complete, we will upload objects to S3 bucket. This bucket can be used to host static web or it can used as S3 store.





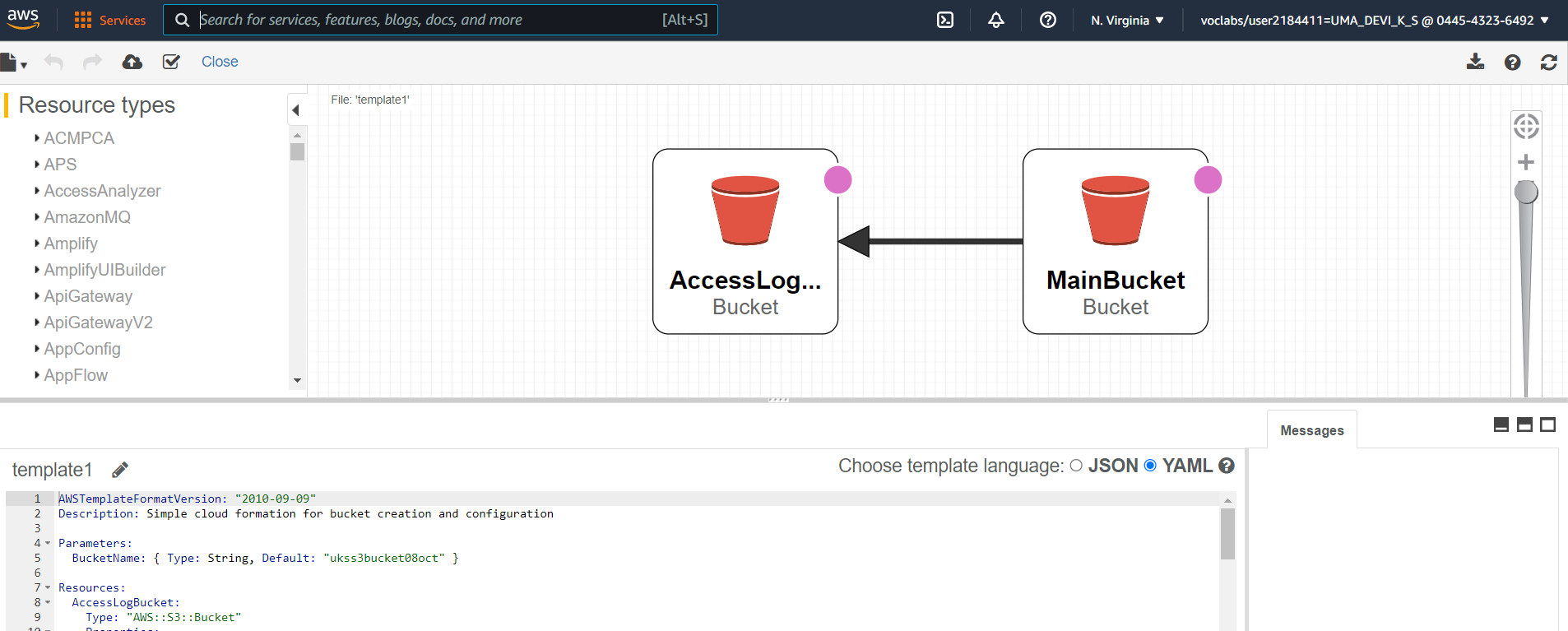


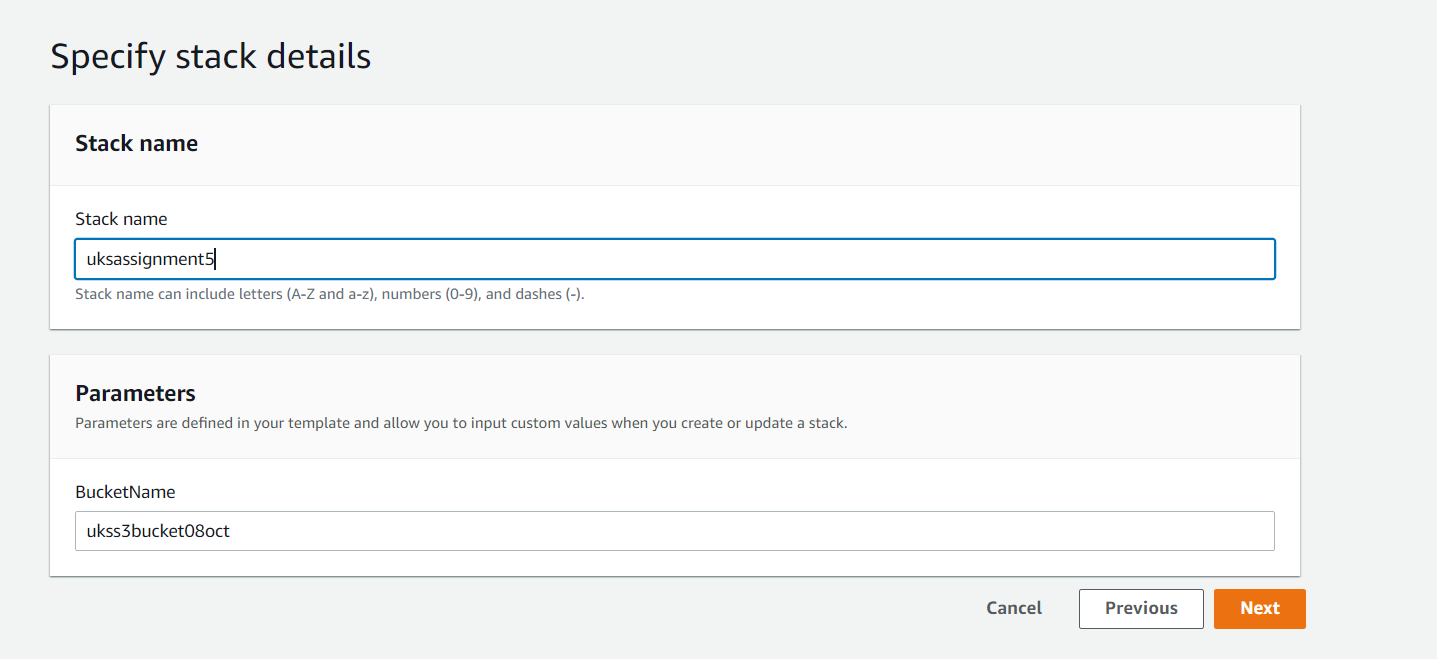


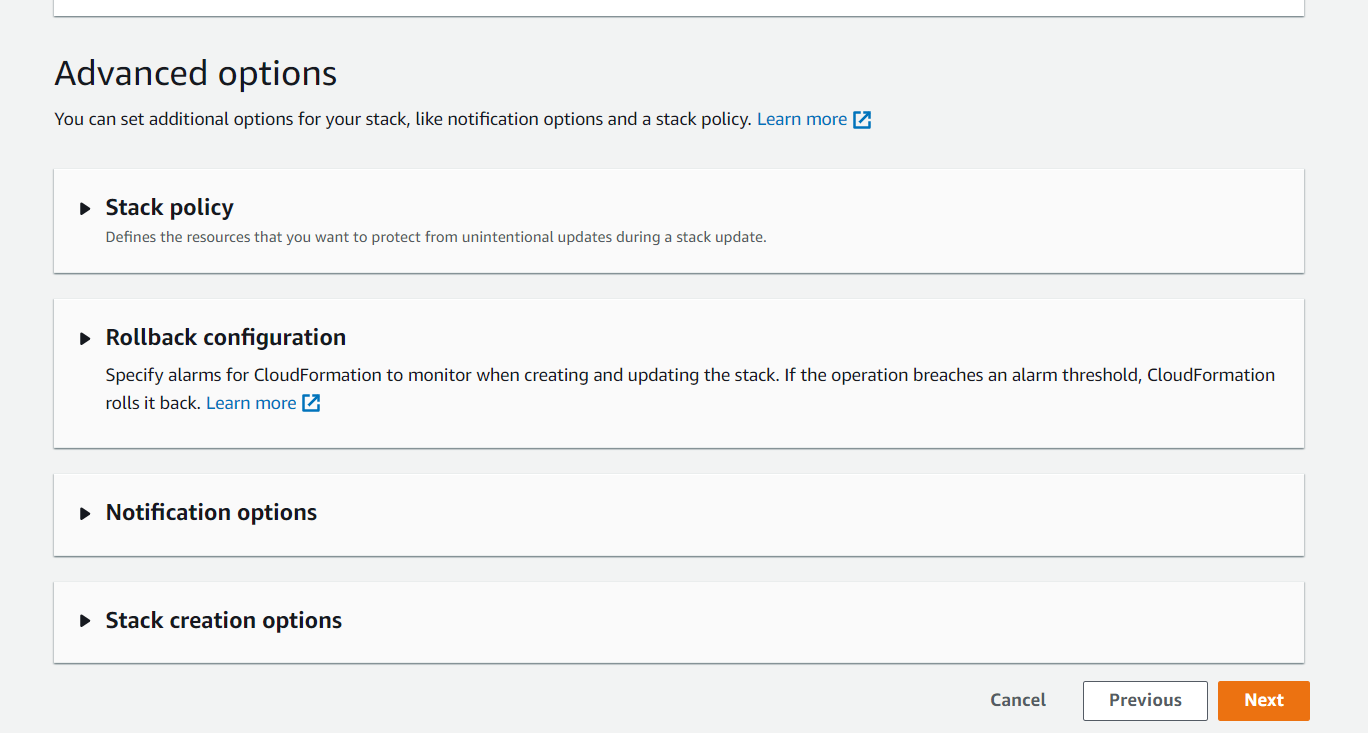


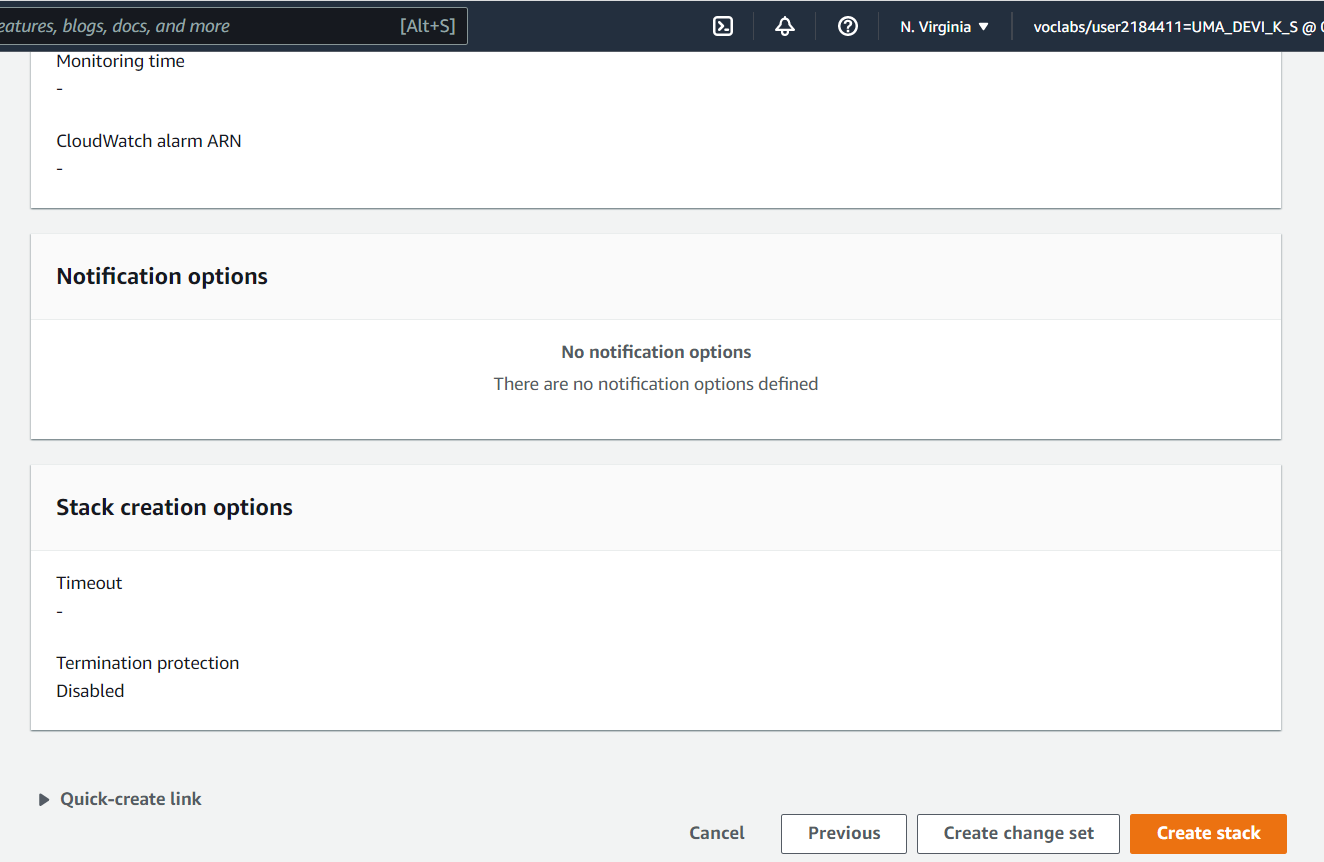


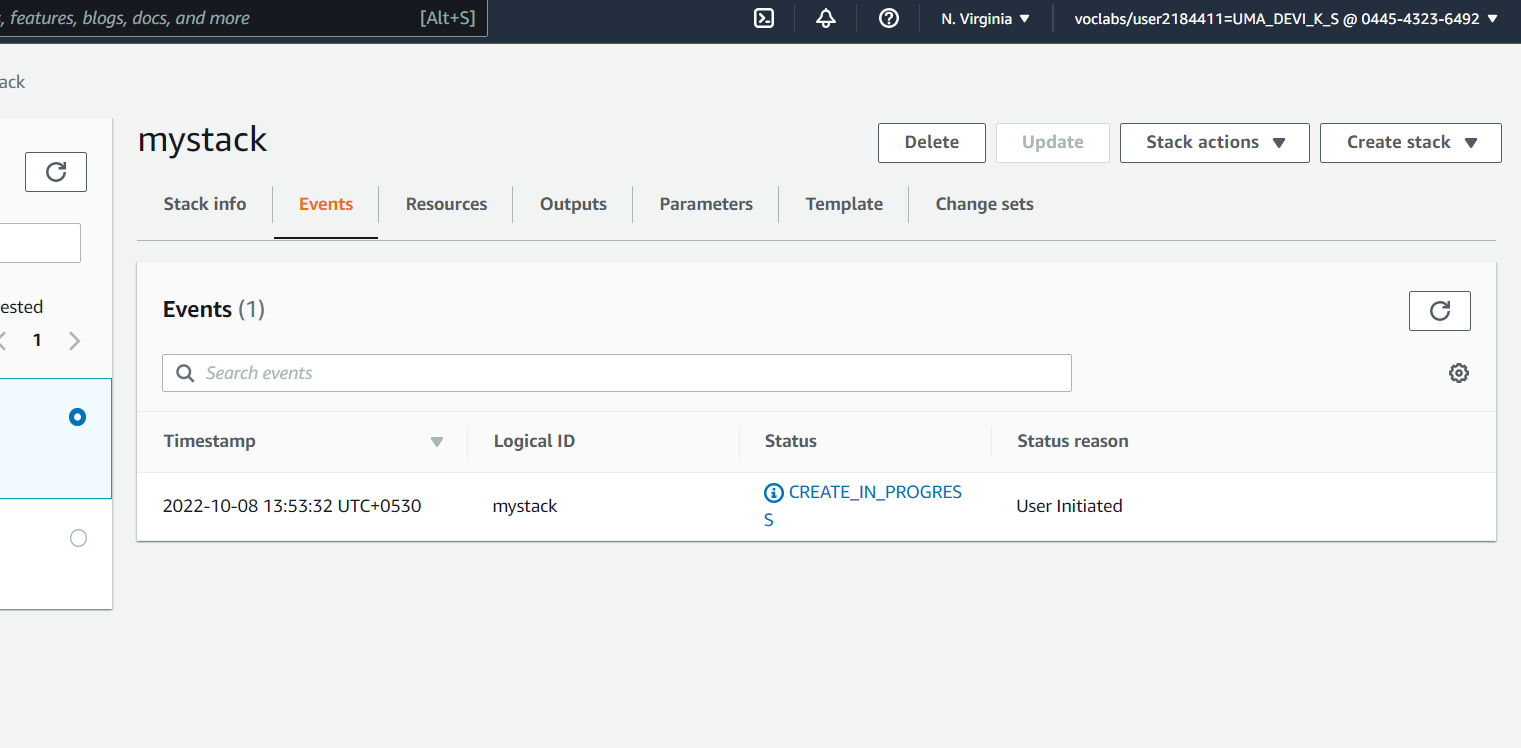
Below is the designer view of the bucket policy created using CF template.

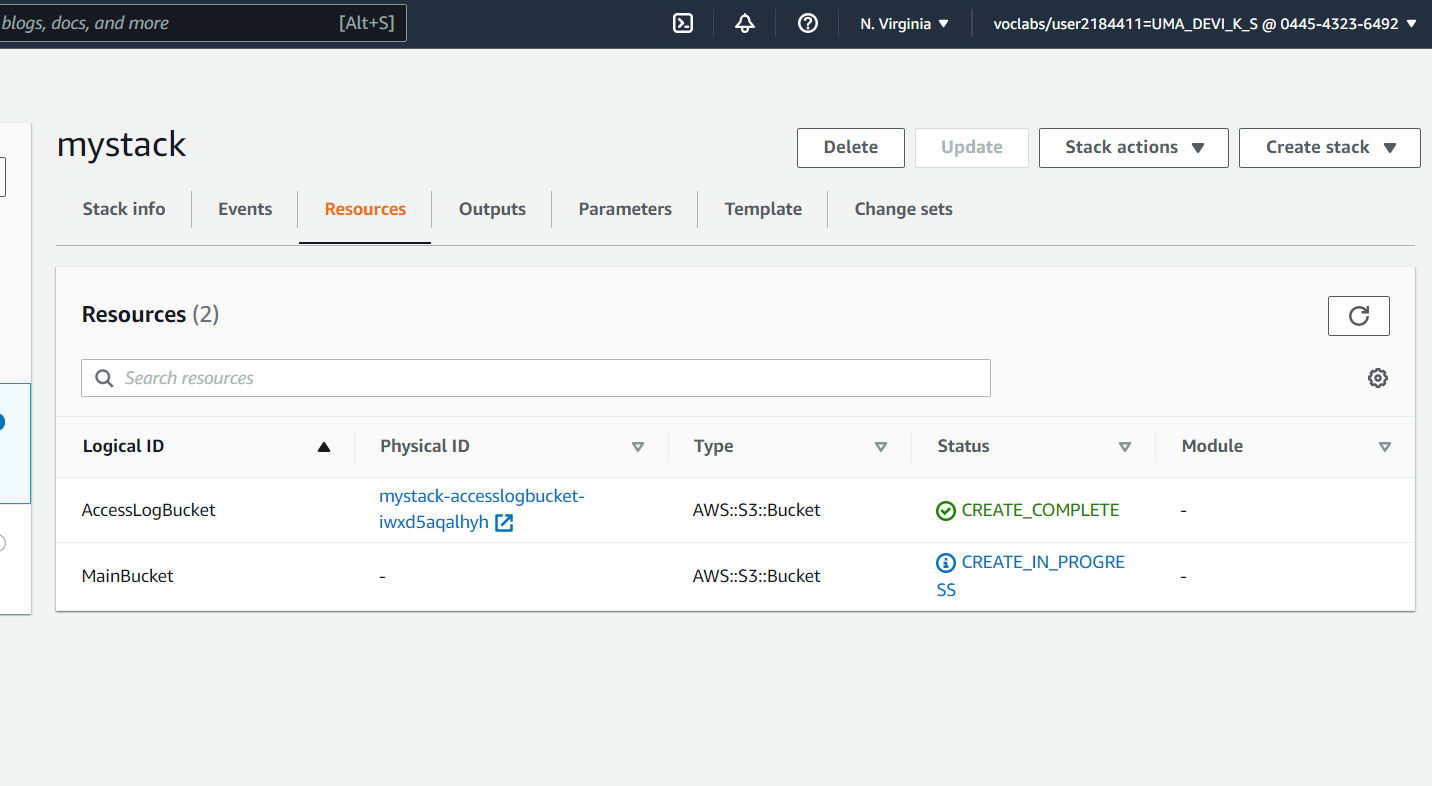


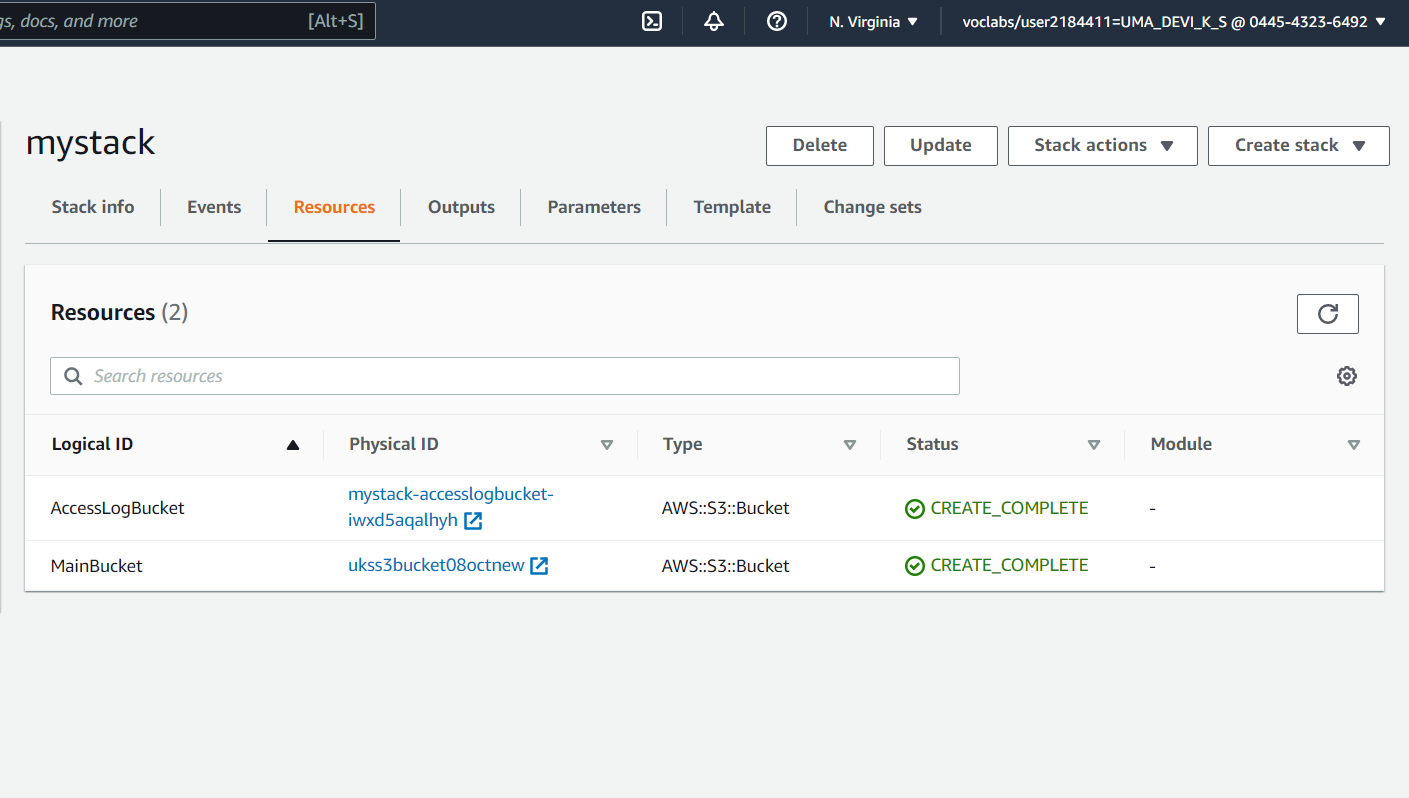


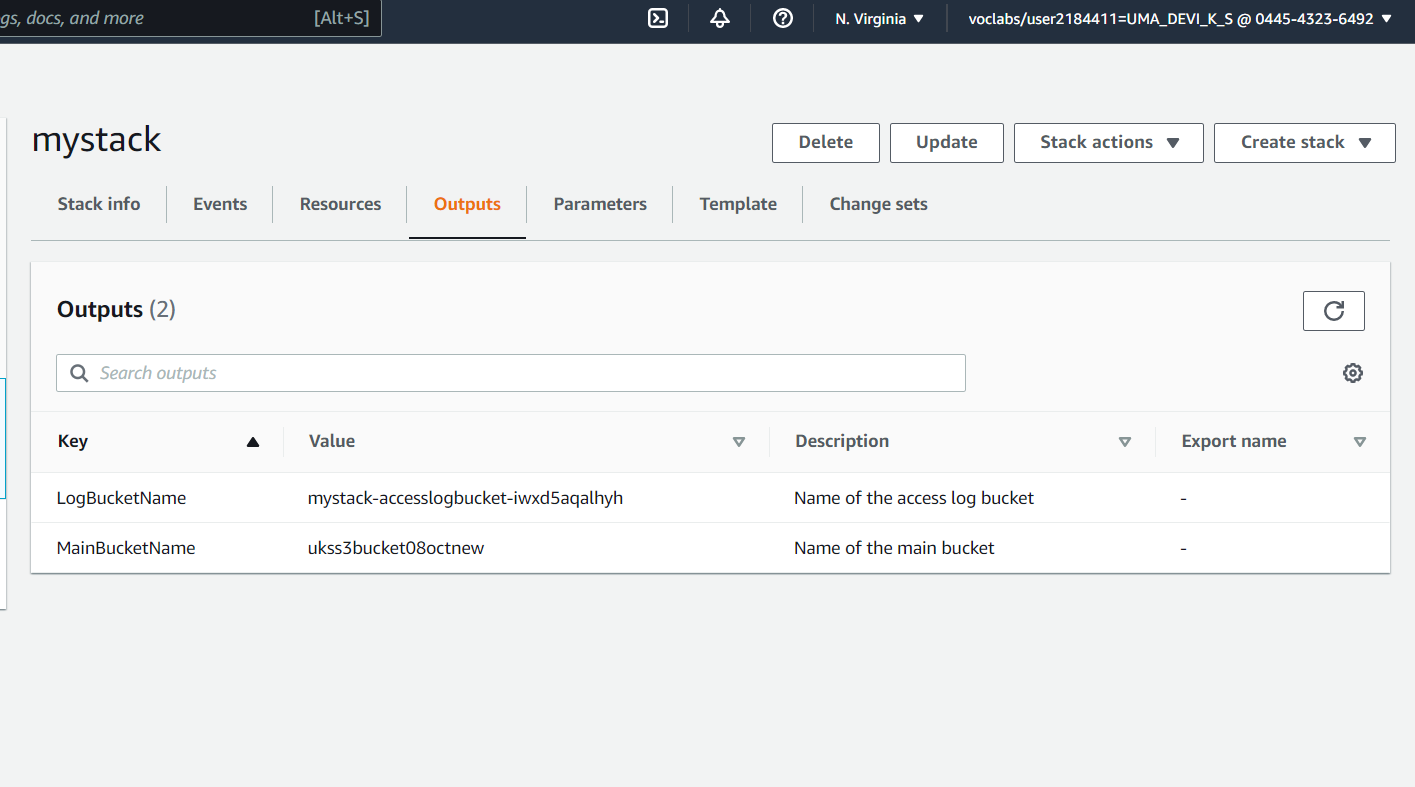












We can see the logs for the complete steps performed during CF execution.