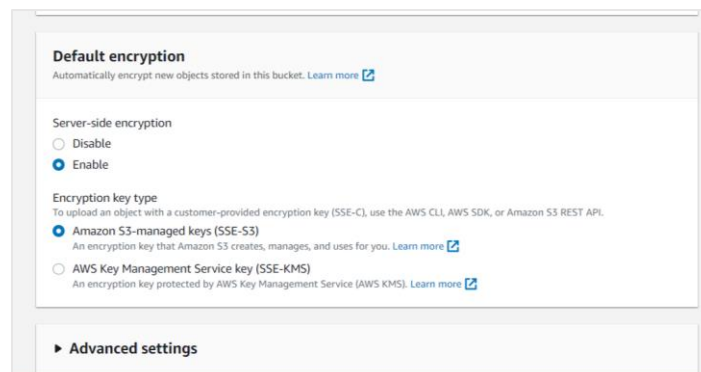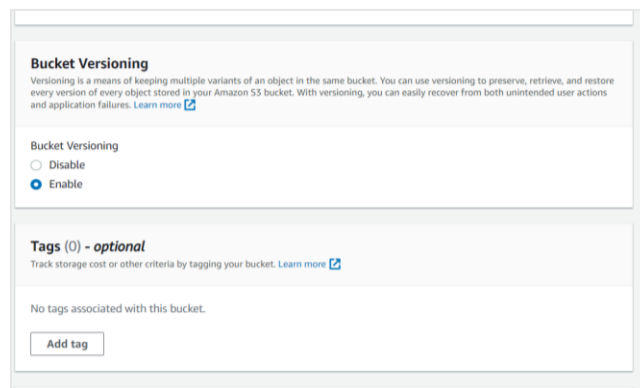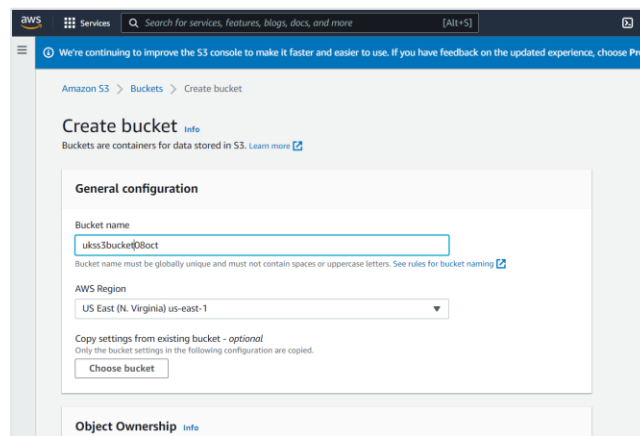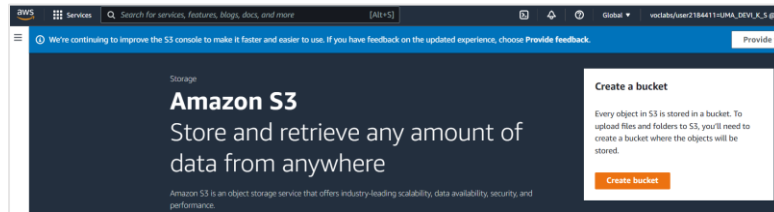**Assignment 5:**

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.
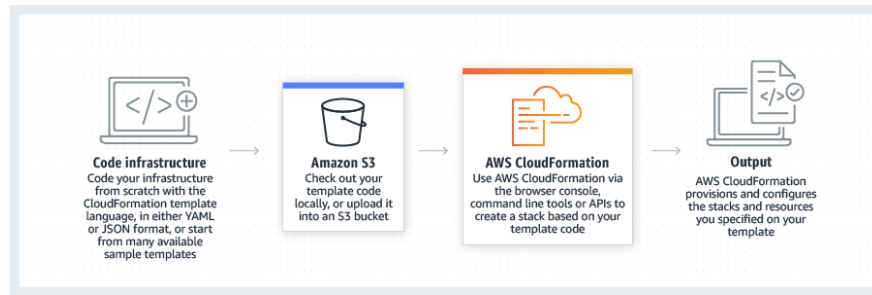
Step1 : We will use cloudFormation service from Management & Governance and create S3 bucket.
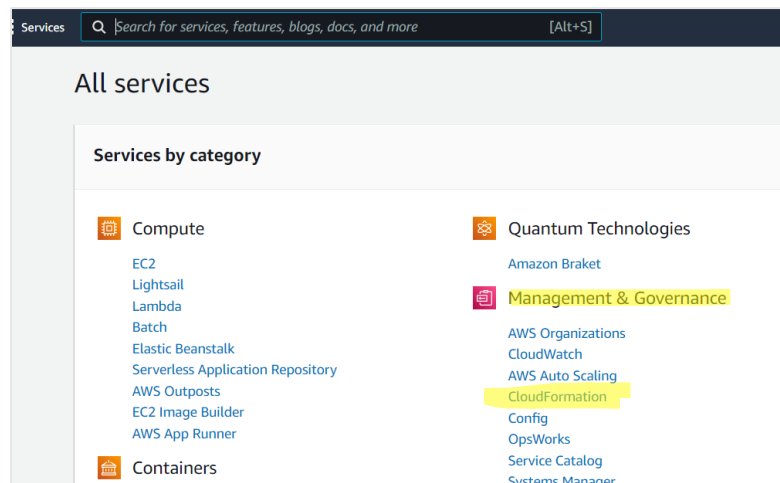
We will prepare template to perform S3 creation using cloud formation. This requires us to prepare a template. This template can be prepared using JSON or YAML formats.

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
	{
	  "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"

```yaml
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:
              - '*'
```

```json
          }
        }
      },
      "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.
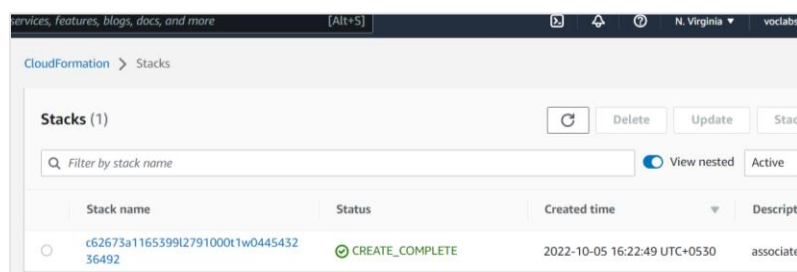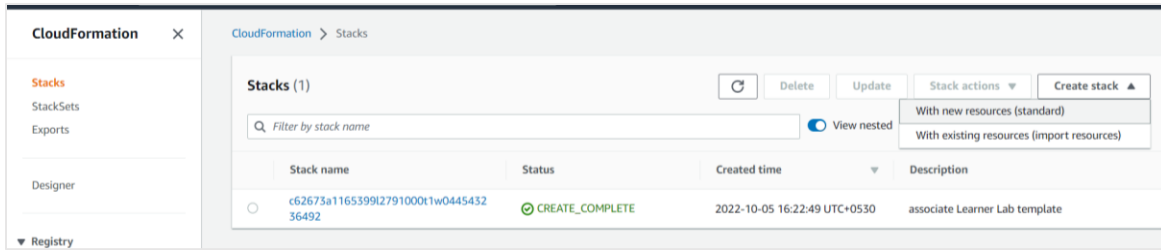
CloudFormation

Stacks
StackSets
Exports

Designer

▼ Registry

CloudFormation > Stacks

Stacks (1)

[⟳] [Delete] [Update] [Stack actions ▾] [Create stack ▲]

With new resources (standard)

With existing resources (import resources)

🔍 Filter by stack name   ● View nested

	Stack name	Status	Created time ▾	Description
○	c62673a1165399l2791000t1w0445432 36492	⊘ CREATE_COMPLETE	2022-10-05 16:22:49 UTC+0530	associate Learner Lab template

---

Prerequisite - Prepare template

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

(● Template is ready)   (○ Use a sample template)   (○ Create template in Designer)

Specify template
A template is a JSON or YAML file that describes your stack's resources and properties.

Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

(○ Amazon S3 URL)   (● Upload a template file)

Upload a template file

[Choose file ⊞] No file chosen
JSON or YAML formatted file

S3 URL: Will be generated when template file is uploaded   [View in Designer]

[Cancel] [**Next**]

Prerequisite - Prepare template

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

(● Template is ready)   (○ Use a sample template)   (○ Create template in Designer)

Specify template
A template is a JSON or YAML file that describes your stack's resources and properties.

Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

(○ Amazon S3 URL)   (● Upload a template file)

Upload a template file

[Choose file ⊞] cloudformation_template.yaml
JSON or YAML formatted file

S3 URL: https://s3-external-1.amazonaws.com/cf-templates-1v8ve1fm6olxu-us-east-1/2022281rOK-cloudformation_template.yaml   [View in Designer]

[Cancel] [**Next**]

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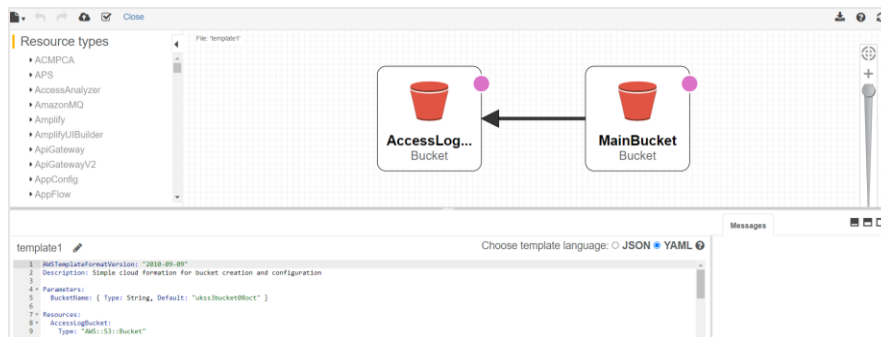
Specify stack details

Stack name

Stack name

uksassignment5

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

BucketName

ukss3bucket08oct

Cancel   Previous   Next

Advanced options

You can set additional options for your stack, like notification options and a stack policy. Learn more ⧉

▶ **Stack policy**
Defines the resources that you want to protect from unintentional updates during a stack update.

▶ **Rollback configuration**
Specify alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back. Learn more ⧉

▶ **Notification options**

▶ **Stack creation options**

Cancel   Previous   Next

Monitoring time
-

CloudWatch alarm ARN
-

Notification options

No notification options
There are no notification options defined

Stack creation options

Timeout
-

Termination protection
Disabled

▶ Quick-create link

Cancel   Previous   Create change set   **Create stack**

---

mystack

Delete   Update   Stack actions ▾   Create stack ▾

Stack info   **Events**   Resources   Outputs   Parameters   Template   Change sets

Events (1)

🔍 Search events

Timestamp ▾	Logical ID	Status	Status reason
2022-10-08 13:53:32 UTC+0530	mystack	ⓘ CREATE_IN_PROGRESS	User Initiated

---

mystack

Delete   Update   Stack action

Stack info   Events   **Resources**   Outputs   Parameters   Template   Change sets

Resources (2)

🔍 Search resources

Logical ID ▲	Physical ID ▽	Type ▽	Status ▽
AccessLogBucket	mystack-accesslogbucket-iwxd5aqalhyh 🔗	AWS::S3::Bucket	✓ CREATE_COMPLETE
MainBucket	-	AWS::S3::Bucket	ⓘ CREATE_IN_PROGRESS

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