

Assignment 3 – S3

Make sure you follow **the least privilege principle for the IAM policies** for these 3 tasks and next classes.

1. Create a bucket for assets of the web app hosted on EC2. Put an image into the bucket.
 - a. Create an inline IAM policy in the LabRole that allows the instance to get objects from the bucket.
 - b. Download the image in the EC2. Copy from S3 to EC2.

```
aws s3 cp s3://<bucket>/< image_name> <image_name>
```

- c. Update the index.html and read the image from the /var/www/html folder
2. Send an email to yourself when the object is created in the bucket.
 - a. You need to create an SNS topic. Modify the default SNS policy while creating the SNS.
 - b. Subscribe it with your email.
3. Write a lambda that returns a Signed URL of the object. Make sure the LabRole has an inline policy that allows getting objects from the bucket.

```
const AWS = require("aws-sdk");
```

```
const s3 = new AWS.S3({apiVersion: '2006-03-01'});
```

```
exports.handler = async (event) => {
```

```
  const params = { Bucket: 'myfirstbucketcreatedwithcli2022cs516', Key: 'Capture.PNG' };
```

```
  return s3.getSignedUrl('getObject', params);
```

```
};
```

Refer: <https://docs.aws.amazon.com/AWSJavaScriptSDK/latest/AWS/S3.html#getSignedUrl-property>

Extra:

- Read a file from S3 in EC2 using S3 Gateway Endpoint. After a successful connection, write S3 resource-based policy that allows read access only from the VPC endpoint in the bucket policy.

Refer: <https://www.youtube.com/watch?v=TqApkvJx5hw>

Setting Event Notification

Create SNS Topic & Add Permission for S3 to Publish Message. Subscribe to the SNS Topic

```
{
  "Sid": "__console_pub_0",
  "Effect": "Allow",
  "Principal": {
    "Service": "s3.amazonaws.com"
  },
  "Action": "SNS:Publish",
  "Resource": "<Your SNS ARN>",
  "Condition": {
    "StringEquals": {
      "aws:SourceAccount": "<Your AWS Account Number>"
    },
    "ArnLike": {
      "aws:SourceArn": "arn:aws:s3:*:*:<Your Bucket Name>"
    }
  }
}
```

Create S3 Event Notification for Your Bucket Under Properties Tab Event Notification

Event notifications (0) Edit Delete Create event notification


Send a notification when specific events occur in your bucket. [Learn more](#)

	Name	Event types	Filters	Destination type	Destination
No event notifications					
Choose Create event notification to be notified when a specific event occurs.					
Create event notification					

1) Click Create Event Notification

...

Create event notification [Info](#)

The notification configuration identifies the events you want Amazon S3 to publish and the destinations where you want Amazon S3 to send the notifications. [Learn more](#) 

General configuration

Event name

S3_Put_Notification

Event name can contain up to 255 characters.

Prefix - *optional*

Limit the notifications to objects with key starting with specified characters.

images/

Suffix - *optional*

Limit the notifications to objects with key ending with specified characters.

.jpg


2) Name Event

Event types


Specify at least one type of event for which you want to receive notifications. [Learn more](#) 

- ☐ All object create events
s3:ObjectCreated:*
 - ☒ Put
s3:ObjectCreated:Put
 - ☐ Post
s3:ObjectCreated:Post
 - ☐ Copy
s3:ObjectCreated:Copy
 - ☐ Multipart upload completed
s3:ObjectCreated:CompleteMultipartUpload
- ☐ All object delete events
s3:ObjectRemoved:*
 - ☐ Permanently deleted
s3:ObjectRemoved:Delete

3) Select Your Criteria

Destination
Choose a destination to publish the event. [Learn more](#) 

☐ **Lambda function**
Run a Lambda function script based on S3 events.

☒ **SNS topic** 
Send notifications to email, SMS, or an HTTP endpoint.


☐ **SQS queue**
Send notifications to an SQS queue to be read by a server.


Specify SNS topic

☒ Choose from your SNS topics

☐ Enter SNS topic ARN

SNS topic

myRequestSNS 



Cancel **Save changes**

Test by Uploading a File to Your S3 Bucket