# SUMMARY:

This Lab covers S3 fundamentals: Static website hosting, KMS, Encrypting, S3 Replication, S3 Pre-signed URLs

## Contents



## 1. Creating a static website with S3

Create public bucket for animals 4 life:

| Name | AWS Region | Access | Creation date |
|---|---|---|---|
| top.10.animals4lifeivan.click | US East (N. Virginia) us-east-1 | Objects can be public | July 14, 2021, 19:29:07 (UTC-05:00) |

Enable statis hosting:

**Static website hosting**

Use this bucket to host a website or redirect requests. Learn more

Edit

Static website hosting

Enabled

Hosting type

Bucket hosting

Bucket website endpoint

When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. Learn more

http://top.10.animals4lifeivan.click.s3-website-us-east-1.amazonaws.com

Upload website files:

| Name | Folder | Type | Size |
|---|---|---|---|
| annnndmerlin.jpg | img/ | image/jpeg | 53.9 KB |
| anothermerlin.jpg | img/ | image/jpeg | 53.9 KB |
| boris.jpg | img/ | image/jpeg | 14.9 KB |
| differentcat1.jpg | img/ | image/jpeg | 53.9 KB |
| differentcat2.jpg | img/ | image/jpeg | 53.9 KB |
| error.html | - | text/html | 233.0 B |
| index.html | - | text/html | 2.6 KB |
| merlin.jpg | img/ | image/jpeg | 53.9 KB |
| merlinagain.jpg | img/ | image/jpeg | 53.9 KB |
| samson.jpg | img/ | image/jpeg | 47.0 KB |

**Destination**

Destination

s3://top.10.animals4lifeivan.click

Create bucket policy:

Bucket ARN

arn:aws:s3:::top.10.animals4lifeivan.click

Policy

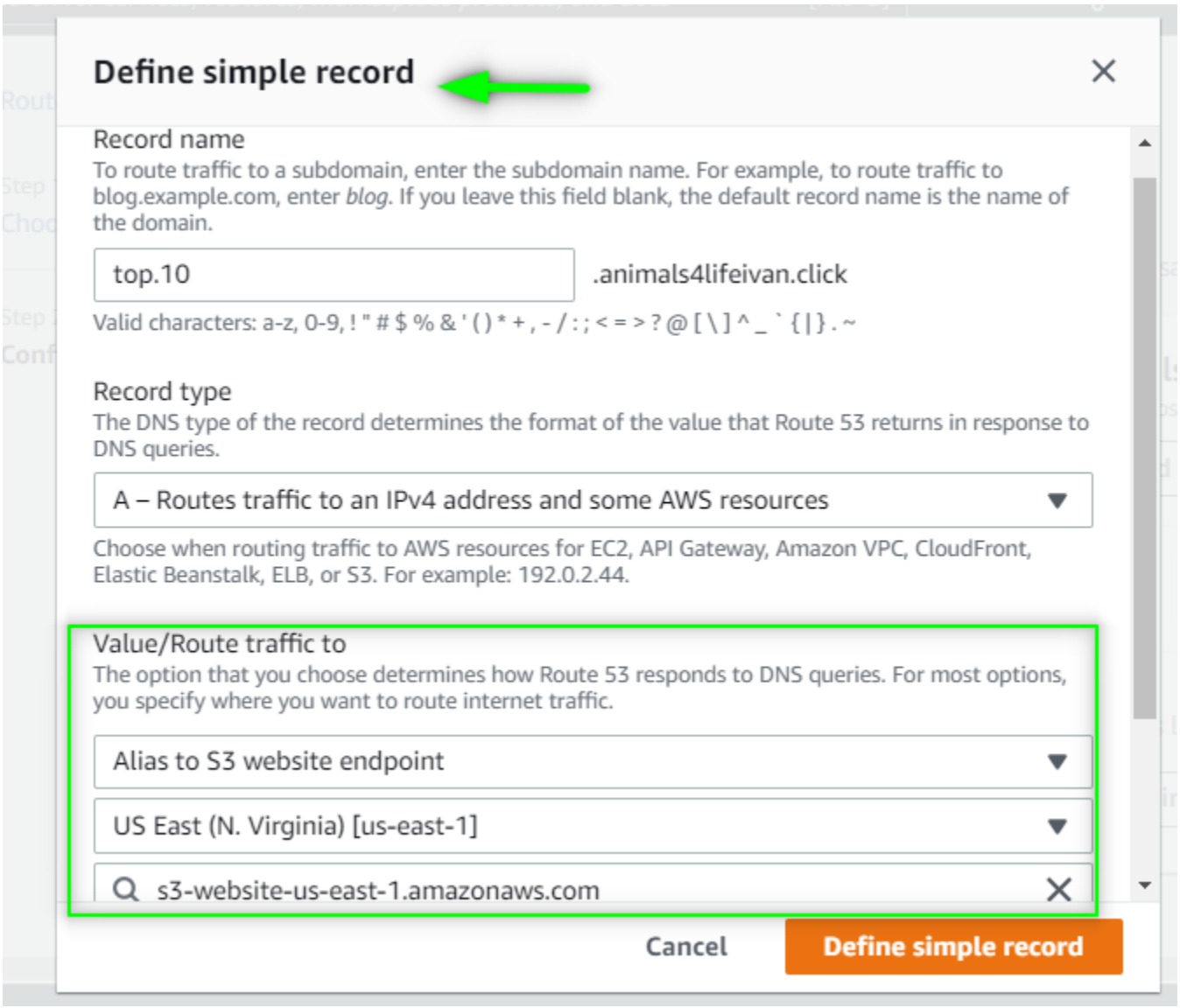
```
1  {
2      "Version":"2012-10-17",
3      "Statement":[
4        {
5          "Sid":"PublicRead",
6          "Effect":"Allow",
7          "Principal": "*",
8          "Action":["s3:GetObject"],
9          "Resource":["arn:aws:s3:::top.10.animals4lifeivan.click/*"]
10       }
11     ]
12   }
13
```

JSON used:

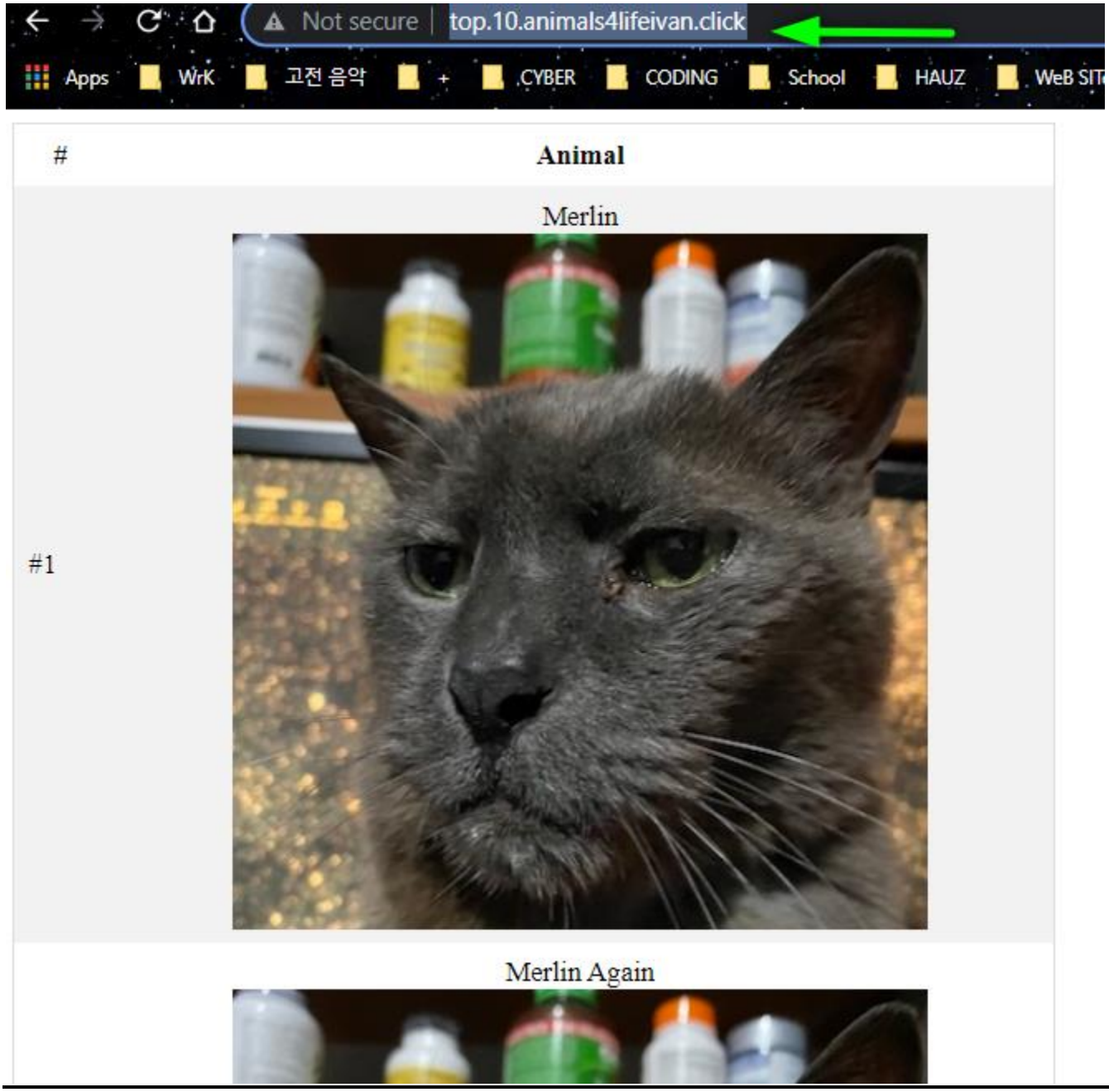
```
{
    "Version":"2012-10-17",
    "Statement":[
```

```
    {
      "Sid":"PublicRead",
      "Effect":"Allow",
      "Principal": "*",
      "Action":["s3:GetObject"],
      "Resource":["arn:aws:s3:::examplebucket/*"]
    }
  ]
}
```

Create DNS record in Route53 and point to bucket:




Test FQDN and verify it points to bucket site:




## 2. Encrypting with KMS

In this [DEMO] lesson we run through the practical steps of creating and configuring a Customer Managed CMK, an Alias and we use that CMK and the CLI tools to encrypt and decrypt some data.

Commands used:

```
  # Shared
echo "find all the doggos, distract them with the yumz" > battleplans.txt


Windows Commands
```

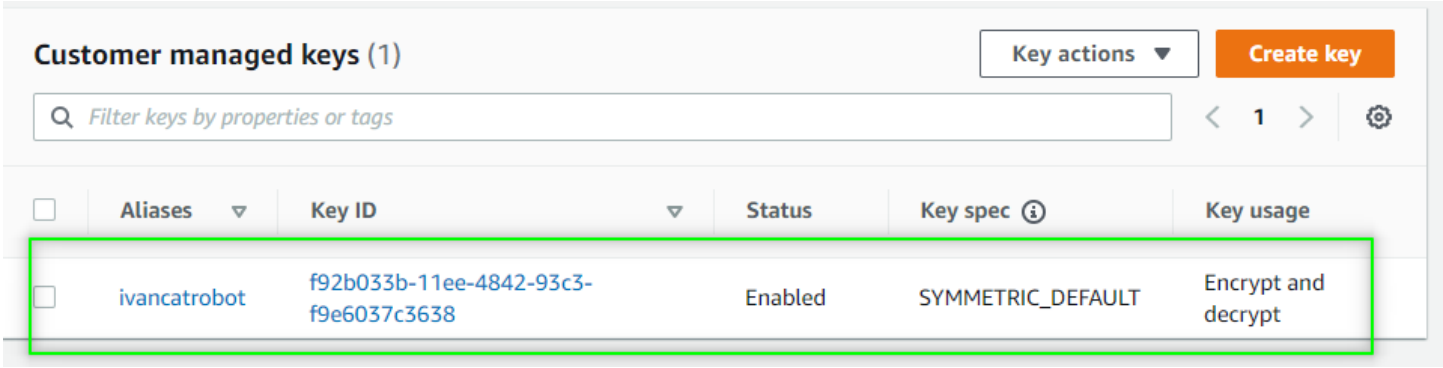
```
aws kms encrypt --key-id alias/catrobot --plaintext fileb://battleplans.txt --output text --
profile iamadmin-general --query CiphertextBlob > battleplans.base64

certutil -decode battleplans.base64 not_battleplans.enc

aws kms decrypt --ciphertext-blob fileb://not_battleplans.enc --output text --
profile iamadmin-general --query Plaintext > decreyptedplans.base64

certutil -decode decreyptedplans.base64 decryptedplans.txt
```

Keypolicy & Key creation:




Key Policy details:

```
{
  "Id": "key-consolepolicy-3",
  "Version": "2012-10-17",
  "Statement": [
      {
          "Sid": "Enable IAM User Permissions",
          "Effect": "Allow",
          "Principal": {
              "AWS": "arn:aws:iam::361618461803:root"
          },
          "Action": "kms:*",
          "Resource": "*"
      },
      {
          "Sid": "Allow access for Key Administrators",
          "Effect": "Allow",
          "Principal": {
              "AWS": "arn:aws:iam::361618461803:user/iamadmin"
          },
          "Action": [
              "kms:Create*",
              "kms:Describe*",
              "kms:Enable*",
              "kms:List*",
              "kms:Put*",
              "kms:Update*",
              "kms:Revoke*",
              "kms:Disable*",
              "kms:Get*",
              "kms:Delete*",
```

```
                "kms:TagResource",
                "kms:UntagResource",
                "kms:ScheduleKeyDeletion",
                "kms:CancelKeyDeletion"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Allow use of the key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::361618461803:user/iamadmin"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Allow attachment of persistent resources",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::361618461803:user/iamadmin"
            },
            "Action": [
                "kms:CreateGrant",
                "kms:ListGrants",
                "kms:RevokeGrant"
            ],
            "Resource": "*",
            "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": "true"
                }
            }
        }
    ]
}
```

Using key to encrypt & decrypt test file in windows 10 cli:

```
Command Prompt
Microsoft Windows [Version 10.0.19042.1083]
(c) Microsoft Corporation. All rights reserved.

C:\Users\IvanVlad>echo "ivan test kms lab 1" > battleplans.txt

C:\Users\IvanVlad>aws kms encrypt --key-id alias/ivancatrobot --plaintext fileb://battleplans.txt --output text --profile iamadmin-general --query CiphertextBlob > batt
leplans.base64

C:\Users\IvanVlad>battleplans.base64

C:\Users\IvanVlad>type battleplans.base64
AQICAHgWNu1rSzEHVb+/LJJwLDdH4DnJvfHyswmb2xWgeTJEKAG+enppnQ2scnXd6Z2FqQgHAAAAdjB0BgkqhkiG9w0BBwagZzBlAgEAMGAGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQMJ9oTyfmislvCkXFVAgEQgDMP
HFYyVmBB4mvyDTlyhOW2qY9zjdLsC9ys0DxGtUit58CTDI+7TbW6+TBpFwXkcPbpfuQ=

C:\Users\IvanVlad>certutil -decode battleplans.base64 not_battleplans.enc
Input Length = 238
Output Length = 176
CertUtil: -decode command completed successfully.

C:\Users\IvanVlad>
C:\Users\IvanVlad>aws kms decrypt --ciphertext-blob fileb://not_battleplans.enc --output text --profile iamadmin-general --query Plaintext > decreyptedplans.base64

C:\Users\IvanVlad>
C:\Users\IvanVlad>certutil -decode decreyptedplans.base64 decryptedplans.txt
Input Length = 34
Output Length = 24
CertUtil: -decode command completed successfully.

C:\Users\IvanVlad>type decryptedplans.txt
"ivan test kms lab 1"

C:\Users\IvanVlad>_
```

Encryption command

File now unreadable

Decrypting

File now readable

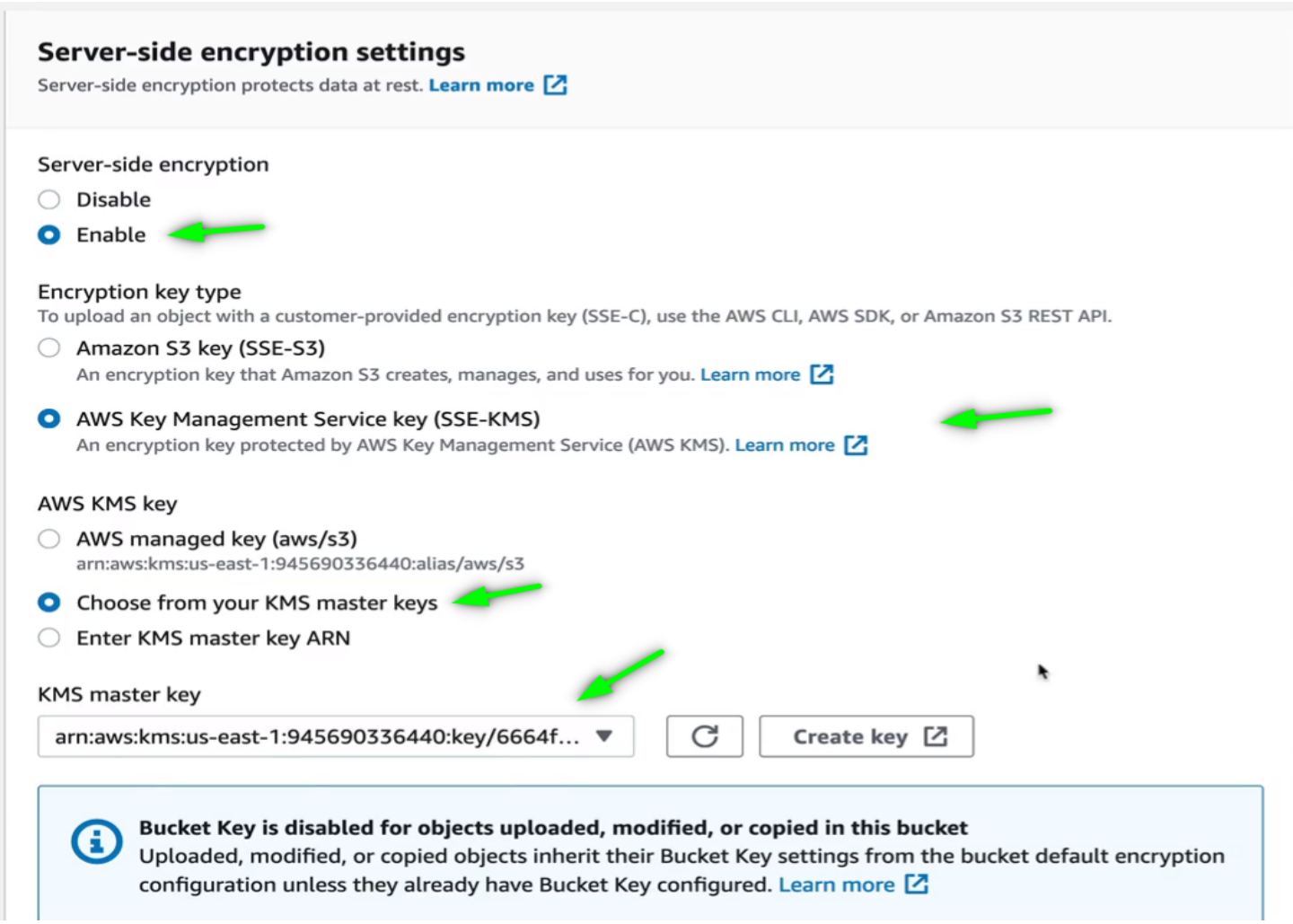
## 3. Encrypting S3 Objects

In this [DEMO] lesson we create an S3 bucket, and upload 4 images to the bucket using different encryption methods.

After adjusting the permissions of the IAMADMIN user we review what access changes occur, and why.

This DEMO focusses on the role separation aspect of S3 encryption using KMS.

Create:




JSON for deny KMS policy for IAM admin:

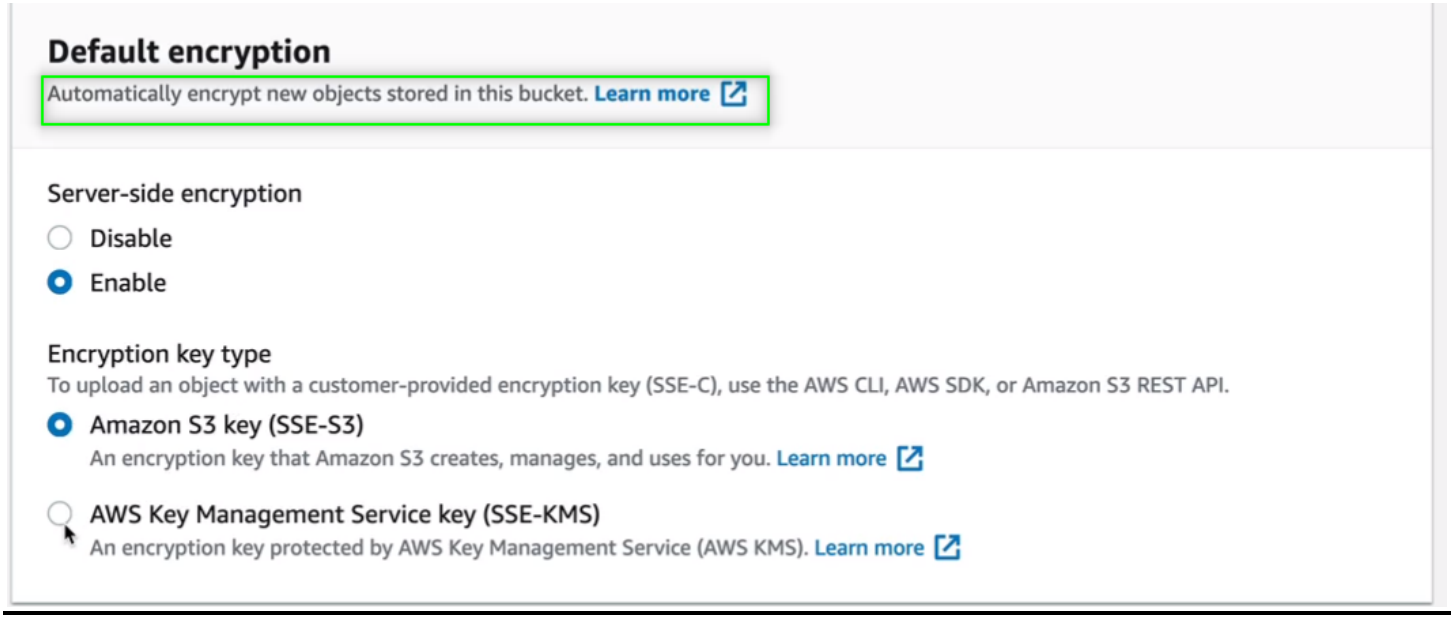
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Deny",
            "Action": "kms:*",
            "Resource": "*"
        }
    ]
}
```

Access Denied with new policy:

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
-<Error>
    <Code>AccessDenied</Code>
    <Message>Access Denied</Message>
    <RequestId>37F0CEDD6099029C</RequestId>
  -<HostId>
      HYLYqWdUu96nmys1M8byU48xDfdNh1Az/215MBOdLdMIdRWrLk1+BLfa+odRS1KFCPH+HA/MZ2U=
    </HostId>
  </Error>
```


Can select default encryption for auto enc for new objects:




## 4. S3 Replication for Disaster Recovery

In this [DEMO] We create 2 S3 buckets - one in N. Virginia, the other in N. California and configure Cross-Region Replication (CRR) between the two.

Replication policy:

```
{
    "Version":"2012-10-17",
    "Statement":[
      {
        "Sid":"PublicRead",
        "Effect":"Allow",
        "Principal": "*",
        "Action":["s3:GetObject"],
        "Resource":["arn:aws:s3:::examplebucket/*"]
      }
```

```
    ]
  }
```

Source Bucket:

Bucket ARN

arn:aws:s3:::sourcebucketivan1234

Policy

```
1  {
2      "Version":"2012-10-17",
3      "Statement":[
4        {
5          "Sid":"PublicRead",
6          "Effect":"Allow",
7          "Principal": "*",
8          "Action":["s3:GetObject"],
9          "Resource":["arn:aws:s3:::arn:aws:s3:::sourcebucketivan1234/*"]
10       }
11     ]
12   }
13
```

Destination Bucket:

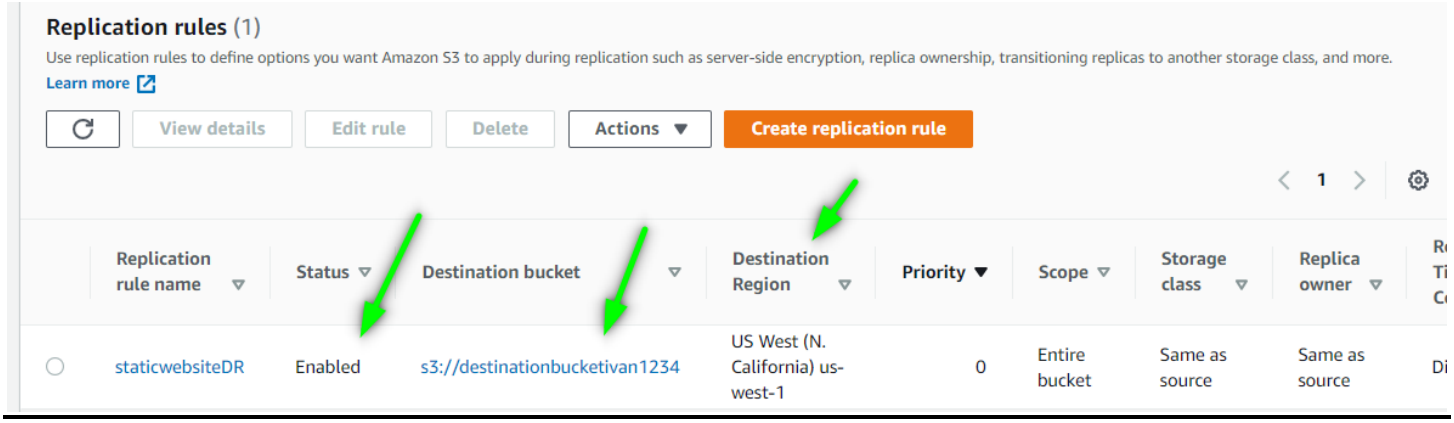
Bucket ARN

arn:aws:s3:::destinationbucketivan1234

Policy

```
1  {
2      "Version":"2012-10-17",
3      "Statement":[
4        {
5          "Sid":"PublicRead",
6          "Effect":"Allow",
7          "Principal": "*",
8          "Action":["s3:GetObject"],
9          "Resource":["arn:aws:s3:::destinationbucketivan1234*"]
10       }
11     ]
12   }
13
```

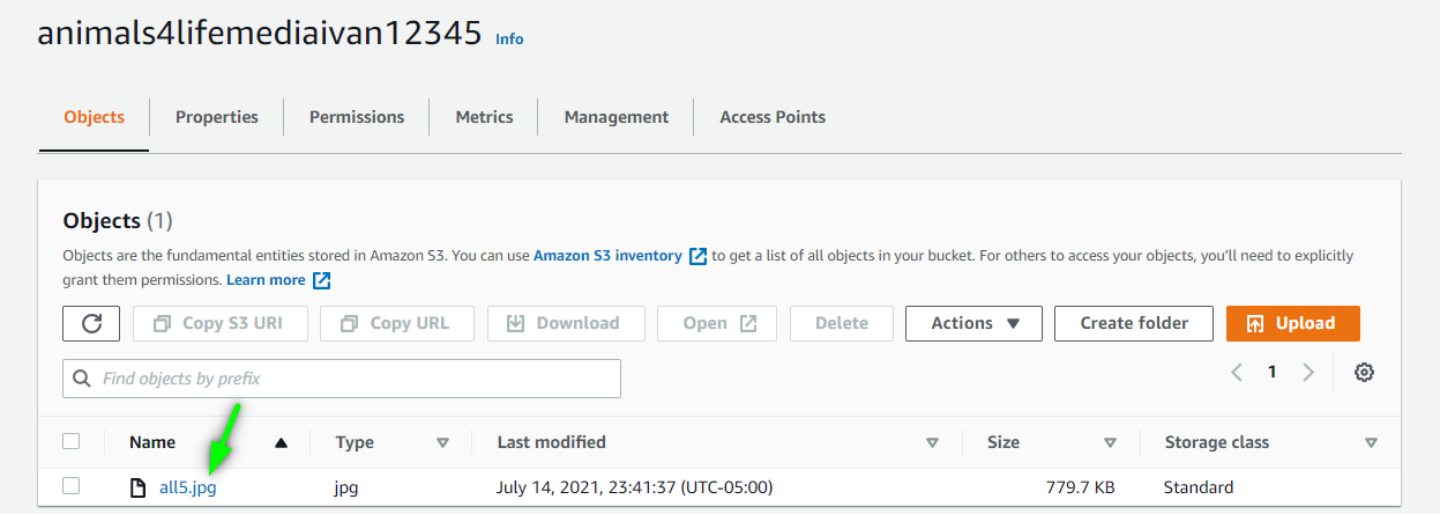
Replication rule for another region (DR):



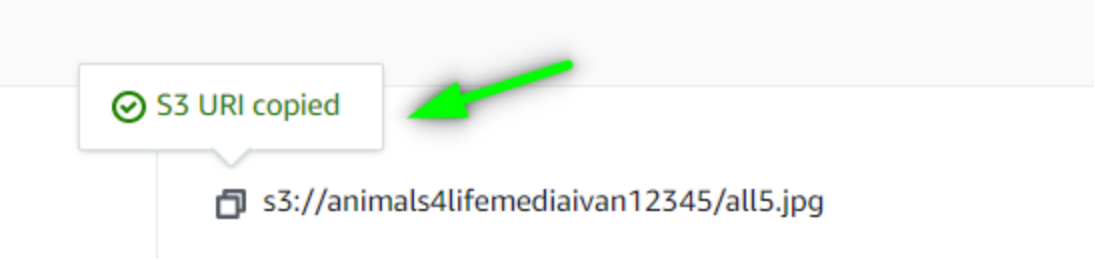

## 5. Pre-Signed URLs

In this [DEMO] lesson you will create a bucket, upload an object and generate a presignedURL allowing access for any unauthenticated identities.

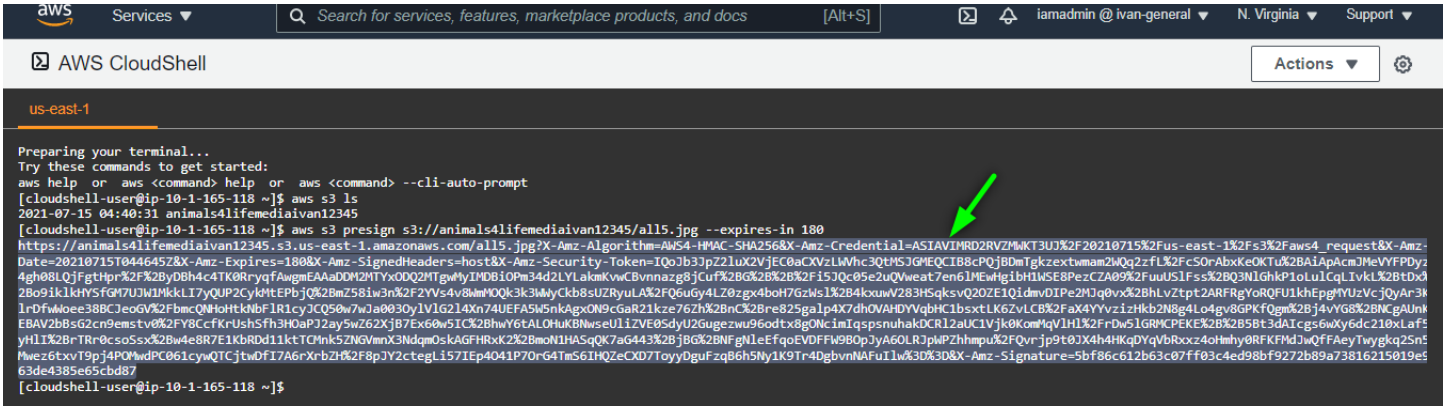
Bucket object target for presignedURL:




URI for cloudshell presignedURL:




Cloudshell presignedURL creation:




Take presignedURL and check if it works: