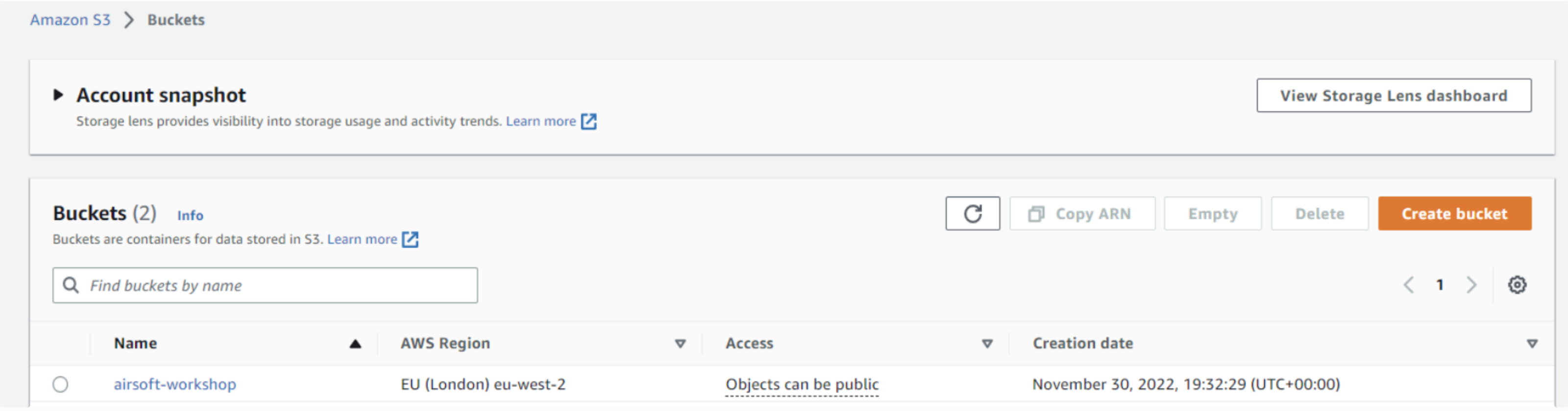
Setting Up AWS to host my static files

Sign up to aws.amazon.com:

search S3 and click create a bucket:

General configurations:

in account, you can view your buckets, next we need to make some changes for it to be allowed for public use.



Account snapshot

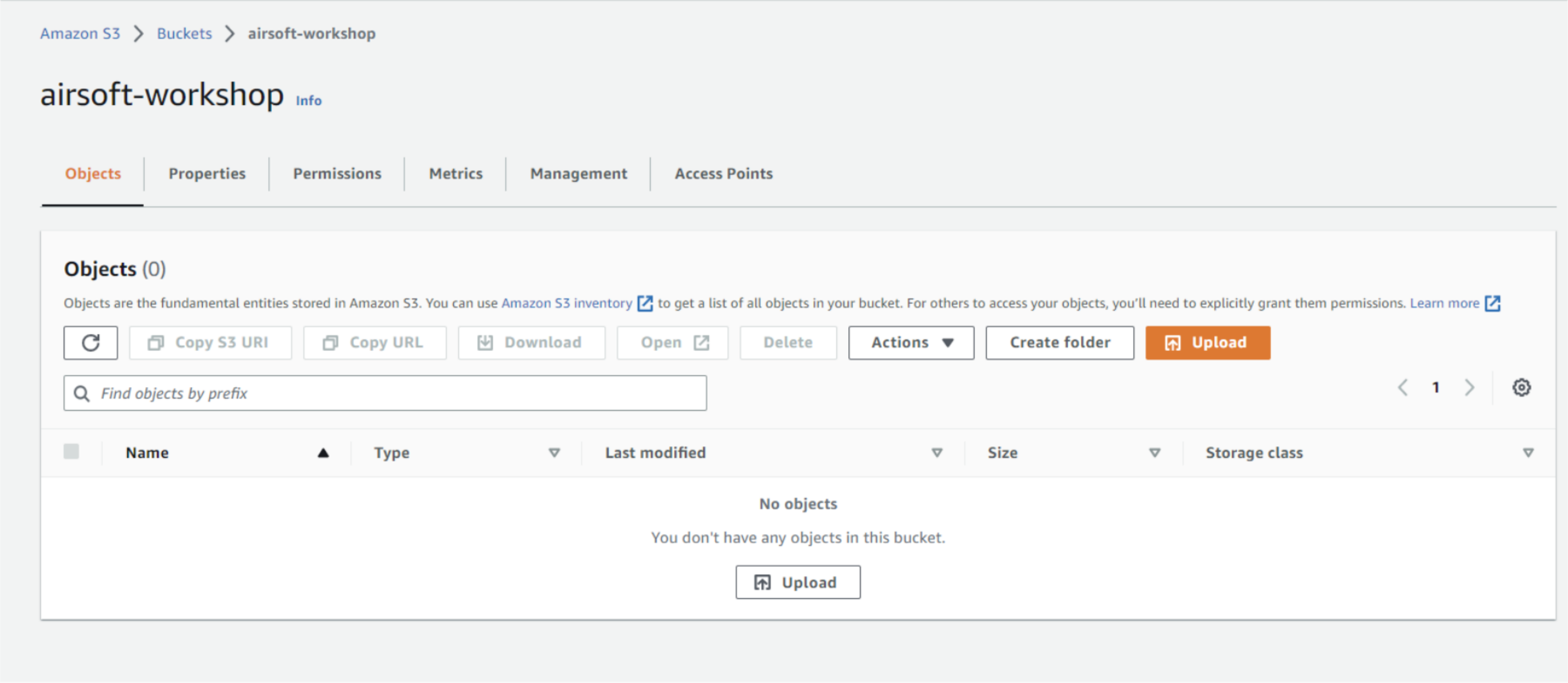
Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets (2)

Buckets are containers for data stored in S3. Learn more

	Name	AWS Region	Access	Creation date
○	airsoft-workshop	EU (London) eu-west-2	Objects can be public	November 30, 2022, 19:32:29 (UTC+00:00)

click on the bucket



airsoft-workshop

Objects (0)

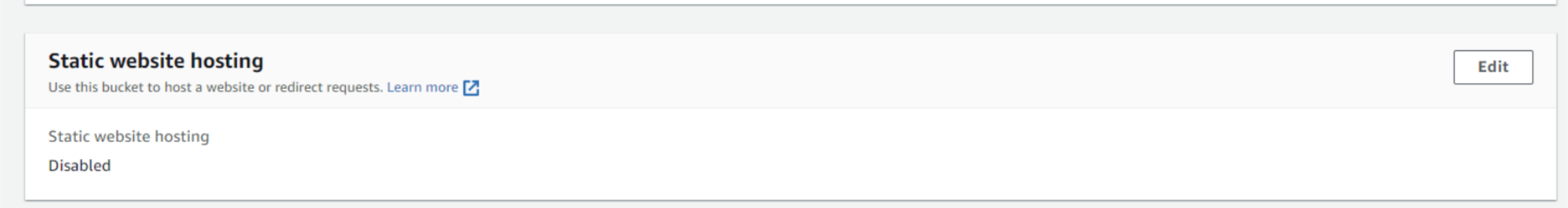
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class

No objects

You don't have any objects in this bucket.

next click on properties and go to static website hosting enable static website hosting...



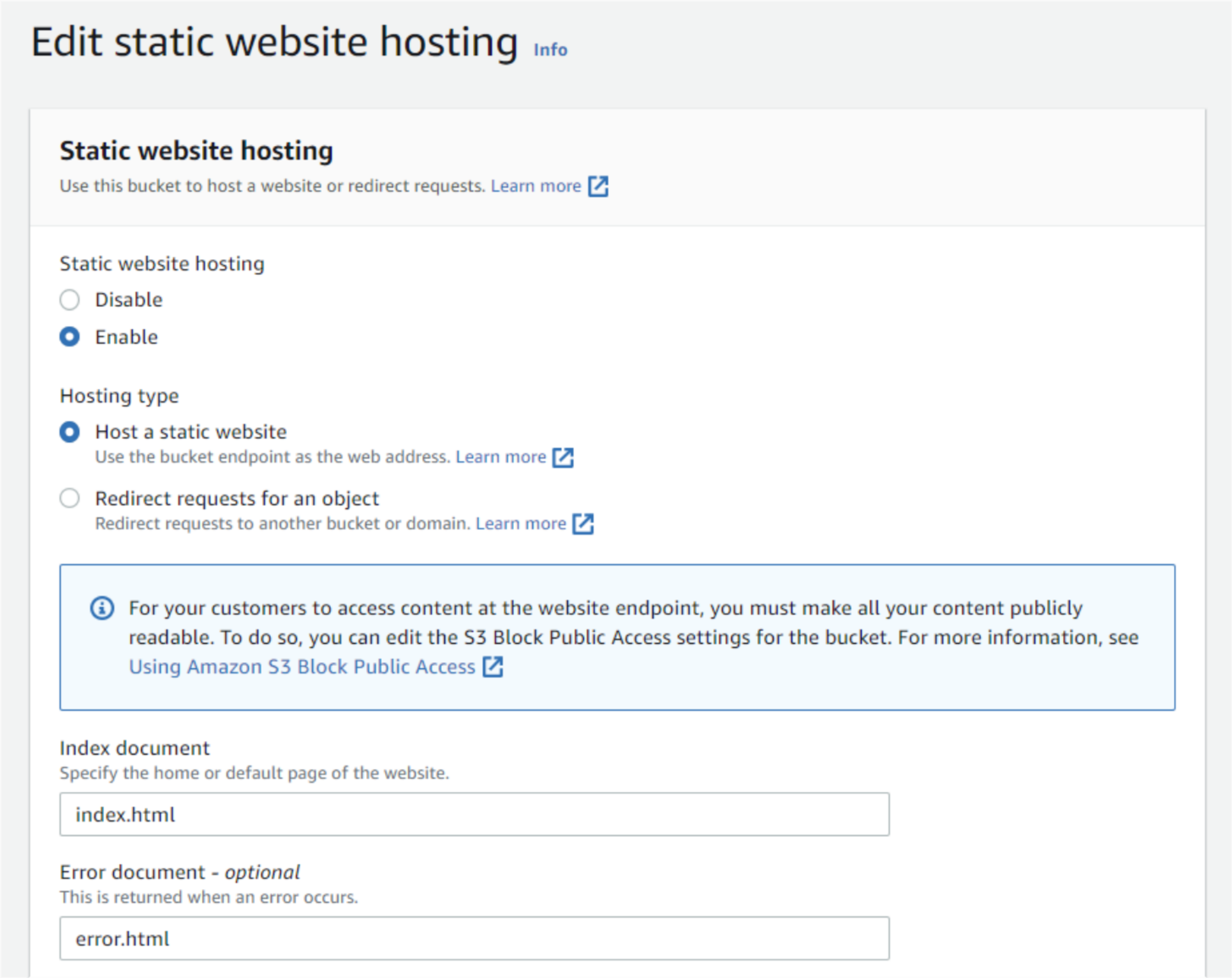
Static website hosting

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

Static website hosting

Disabled

[input default index and error htmls, but these will not be used] and then save next head to the permissions tab...



Edit static website hosting Info

Static website hosting

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

Static website hosting

○ Disable

● Enable

Hosting type

● Host a static website
Use the bucket endpoint as the web address. Learn more

○ Redirect requests for an object
Redirect requests to another bucket or domain. Learn more

For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access

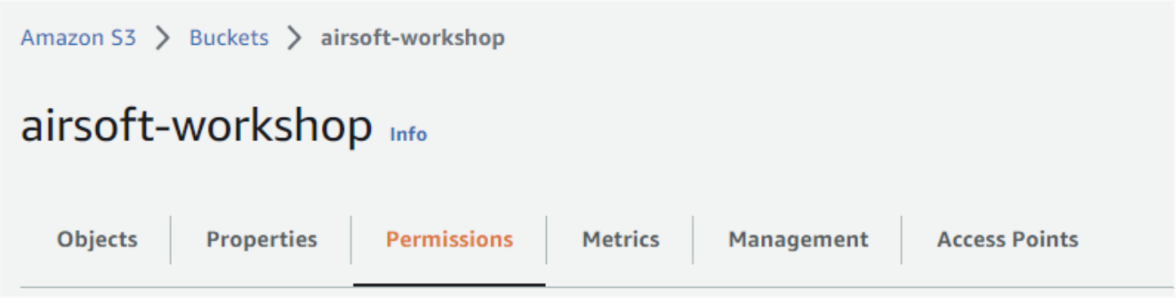
Index document
Specify the home or default page of the website.

index.html

Error document - *optional*
This is returned when an error occurs.

error.html

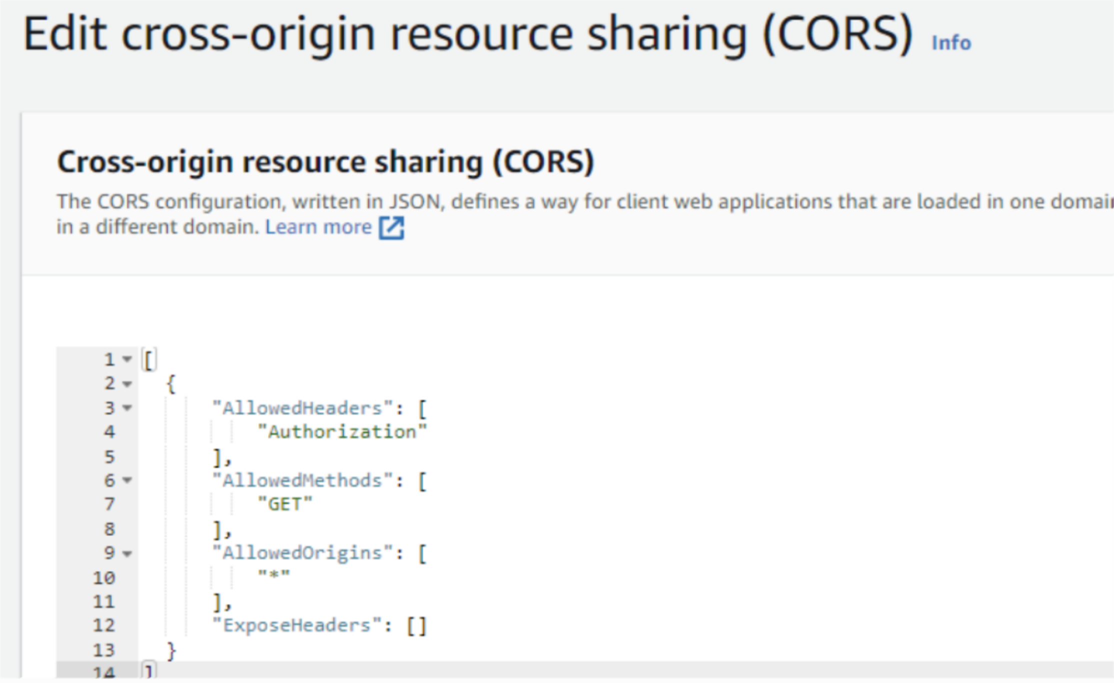
Next there are 3 changes to make...



Amazon S3 > Buckets > airsoft-workshop

airsoft-workshop Info

first CORS config, add allowed hosts



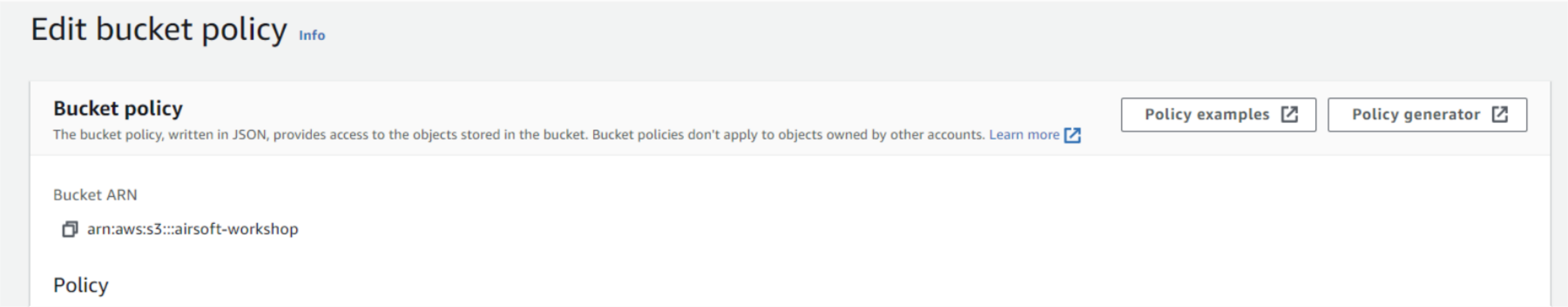
Edit cross-origin resource sharing (CORS) Info

Cross-origin resource sharing (CORS)

The CORS configuration, written in JSON, defines a way for client web applications that are loaded in one domai
in a different domain. Learn more

```
[
  {
      "AllowedHeaders": [
          "Authorization"
      ],
      "AllowedMethods": [
          "GET"
      ],
      "AllowedOrigins": [
          "*"
      ],
      "ExposeHeaders": []
  }
]
```

2nd) open bucket policy, edit and use the policy generator in the top right



Edit bucket policy Info

Bucket policy

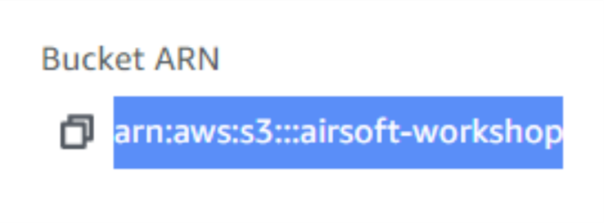
The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more

Bucket ARN

arn:aws:s3:::airsoft-workshop

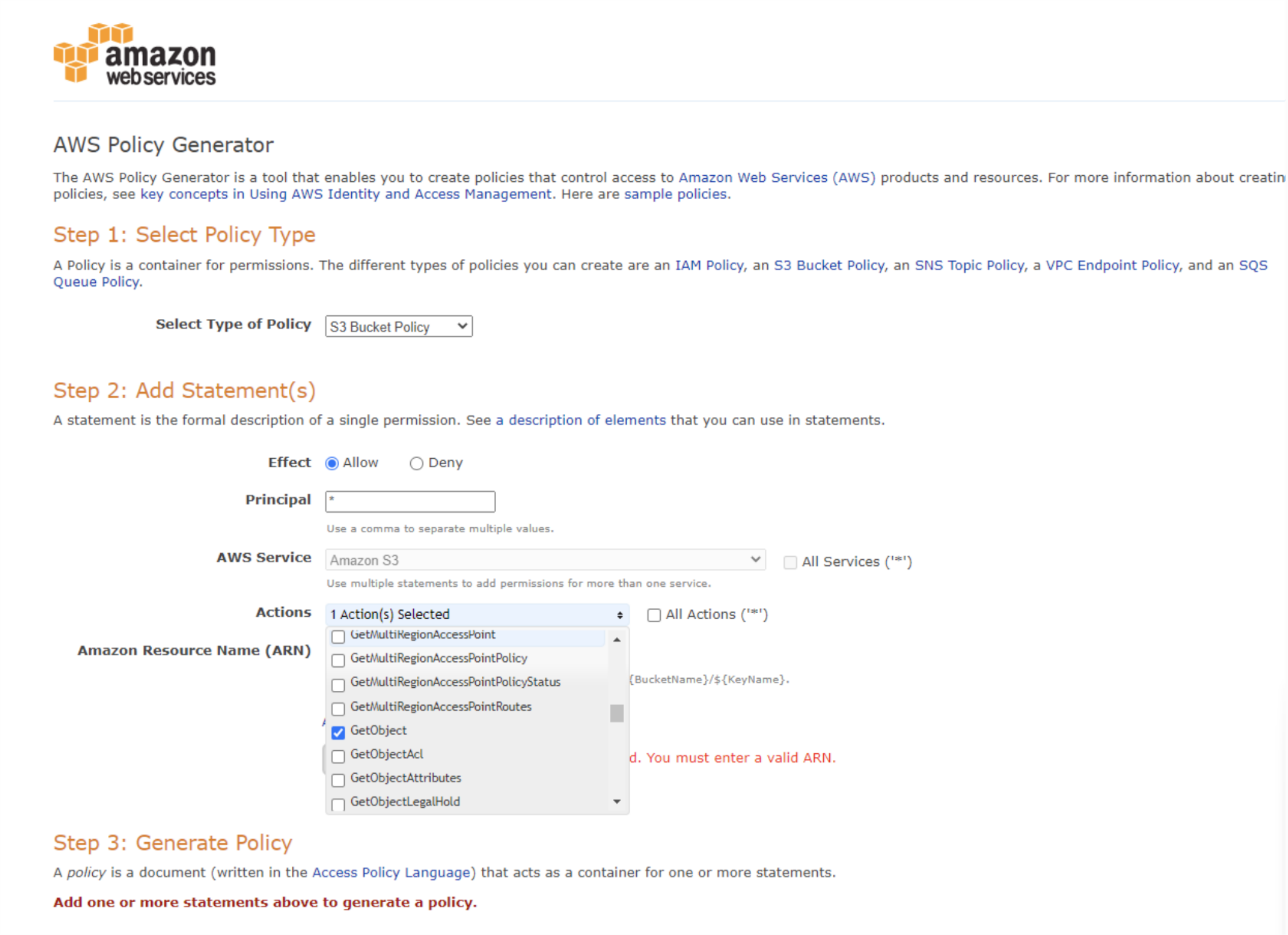
Policy

select type of policy = S3 bucket policy
Effect = Allow
Principle = * (indicates all)
AWS service = fixed
Actions = GetObject
Amazon Resource Name ARN = get from bucket policy rab (above in blue)



Bucket ARN

arn:aws:s3:::airsoft-workshop



AWS Policy Generator

The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creatin policies, see key concepts in Using AWS Identity and Access Management. Here are sample policies.

Step 1: Select Policy Type

A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy, a VPC Endpoint Policy, and an SQS Queue Policy.

Select Type of Policy: S3 Bucket Policy

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a description of elements that you can use in statements.

Effect: ● Allow ○ Deny

Principal: *

Use a comma to separate multiple values.

AWS Service: Amazon S3 ☐ All Services ('*')

Use multiple statements to add permissions for more than one service.

Actions: 1 Action(s) Selected ☐ All Actions ('*')

- ☐ GetMultiRegionAccessPoint
- ☐ GetMultiRegionAccessPointPolicy
- ☐ GetMultiRegionAccessPointPolicyStatus
- ☐ GetMultiRegionAccessPointRoutes
- ☑ GetObject
- ☐ GetObjectAcl
- ☐ GetObjectAttributes
- ☐ GetObjectLegalHold

Amazon Resource Name (ARN)

{BucketName}/${KeyName}.

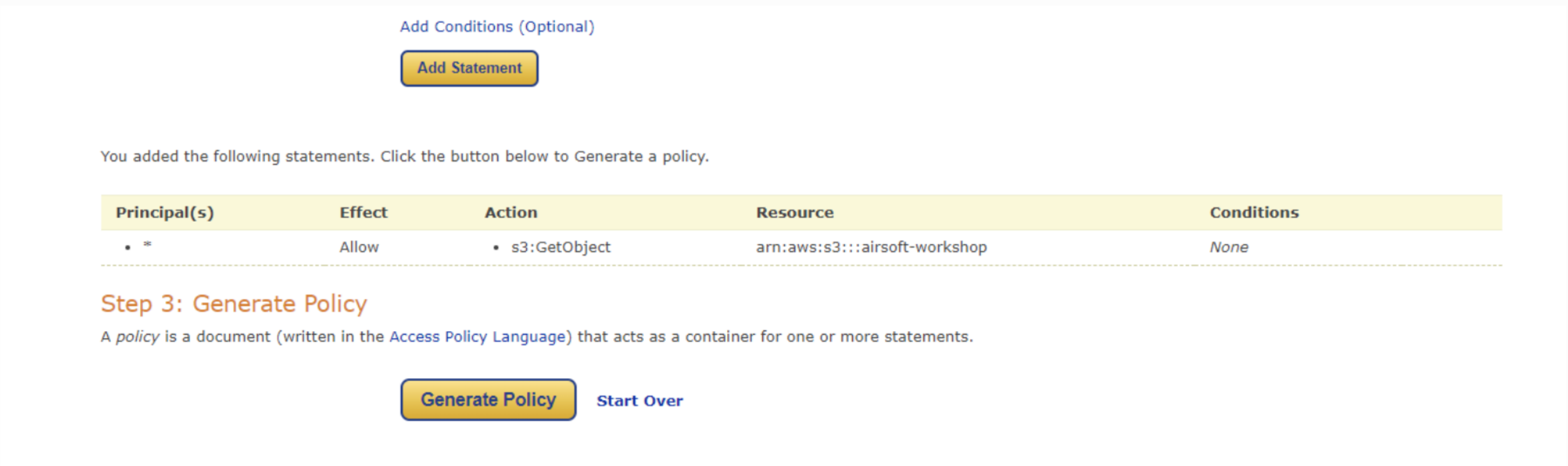
d. You must enter a valid ARN.

Step 3: Generate Policy

A *policy* is a document (written in the Access Policy Language) that acts as a container for one or more statements.

Add one or more statements above to generate a policy.

click create statement then generate policy...



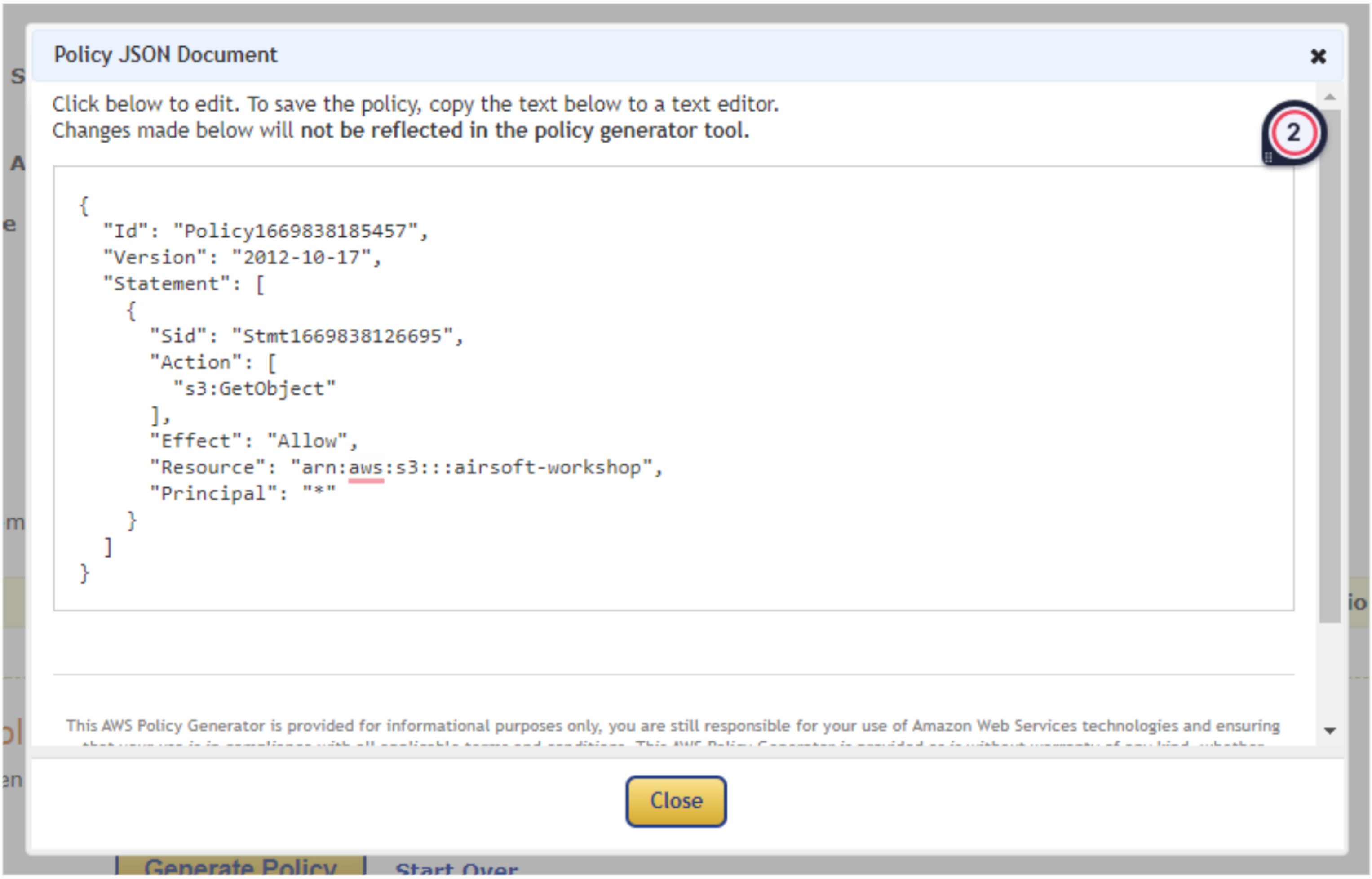
You added the following statements. Click the button below to Generate a policy.

Principal(s)	Effect	Action	Resource	Conditions
- *	Allow	- s3:GetObject	arn:aws:s3:::airsoft-workshop	*None*

Step 3: Generate Policy

A *policy* is a document (written in the Access Policy Language) that acts as a container for one or more statements.

now add this policy in the bucket policy, but do not save yet
add a /* onto the end of the resource policy



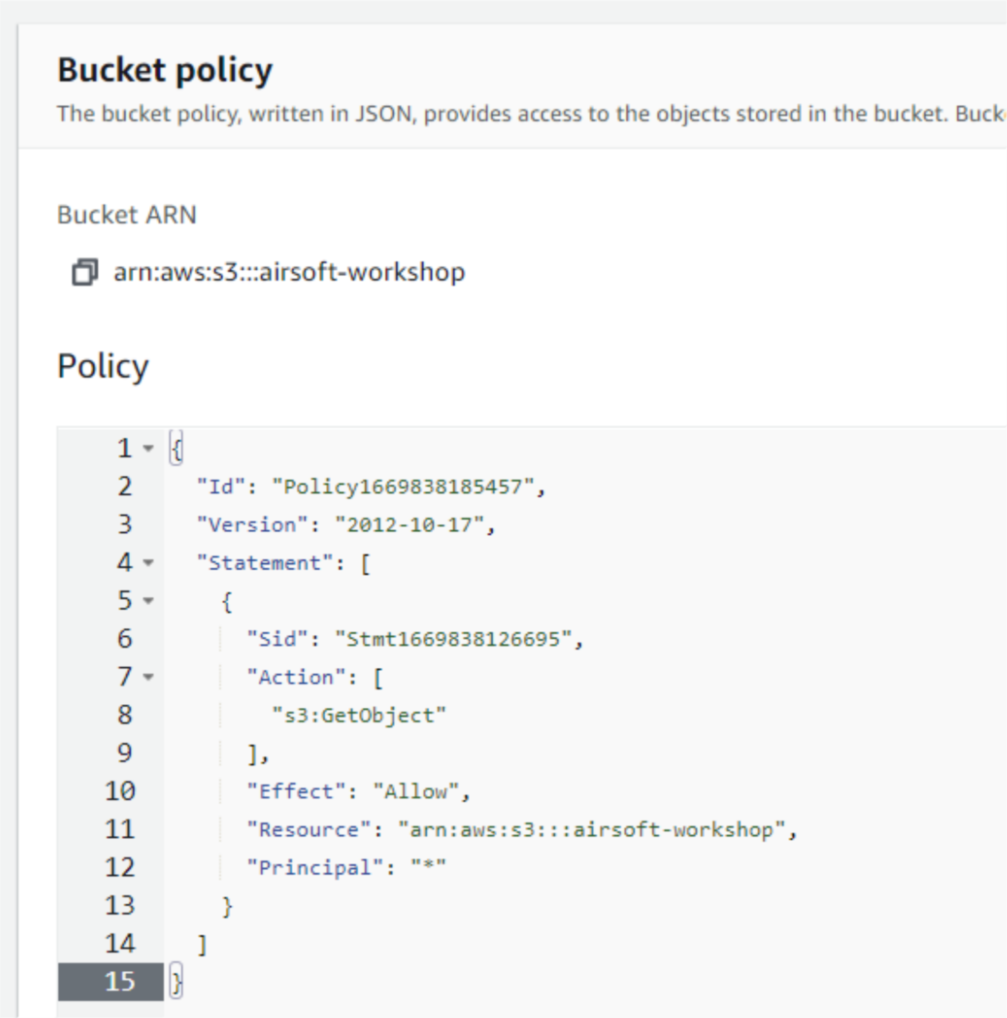
Click below to edit. To save the policy, copy the text below to a text editor.
Changes made below will **not be reflected in the policy generator tool.**

```
{
  "Id": "Policy1669838185457",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1669838126695",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::airsoft-workshop",
      "Principal": "*"
    }
  ]
}
```

This AWS Policy Generator is provided for informational purposes only, you are still responsible for your use of Amazon Web Services technologies and ensuring that your use is in compliance with all applicable terms and conditions. This AWS Policy Generator is provided as is without warranty of any kind, whether

now save changes

Also we need to change the object ownership... set the settings like so on the image to the right and then save.



Bucket policy

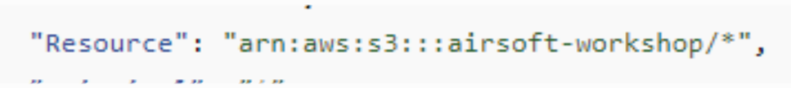
The bucket policy, written in JSON, provides access to the objects stored in the bucket. Buck

Bucket ARN

arn:aws:s3:::airsoft-workshop

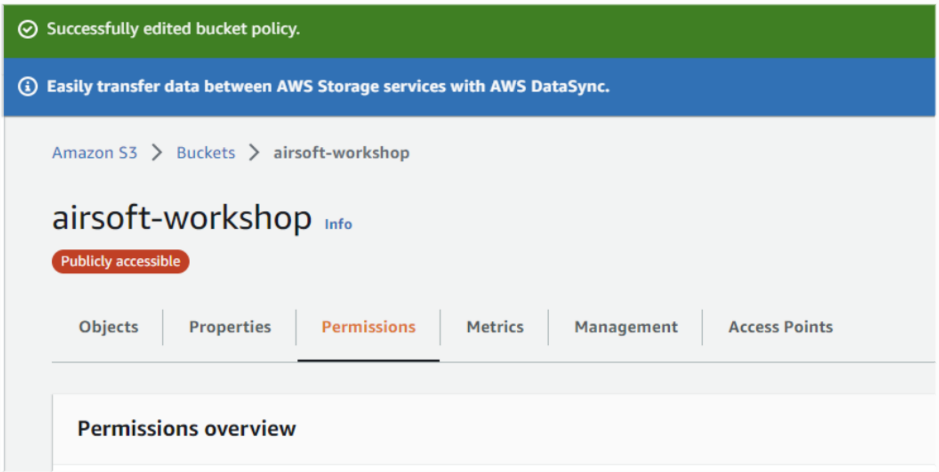
Policy

```
{
  "Id": "Policy1669838185457",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1669838126695",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::airsoft-workshop",
      "Principal": "*"
    }
  ]
}
```



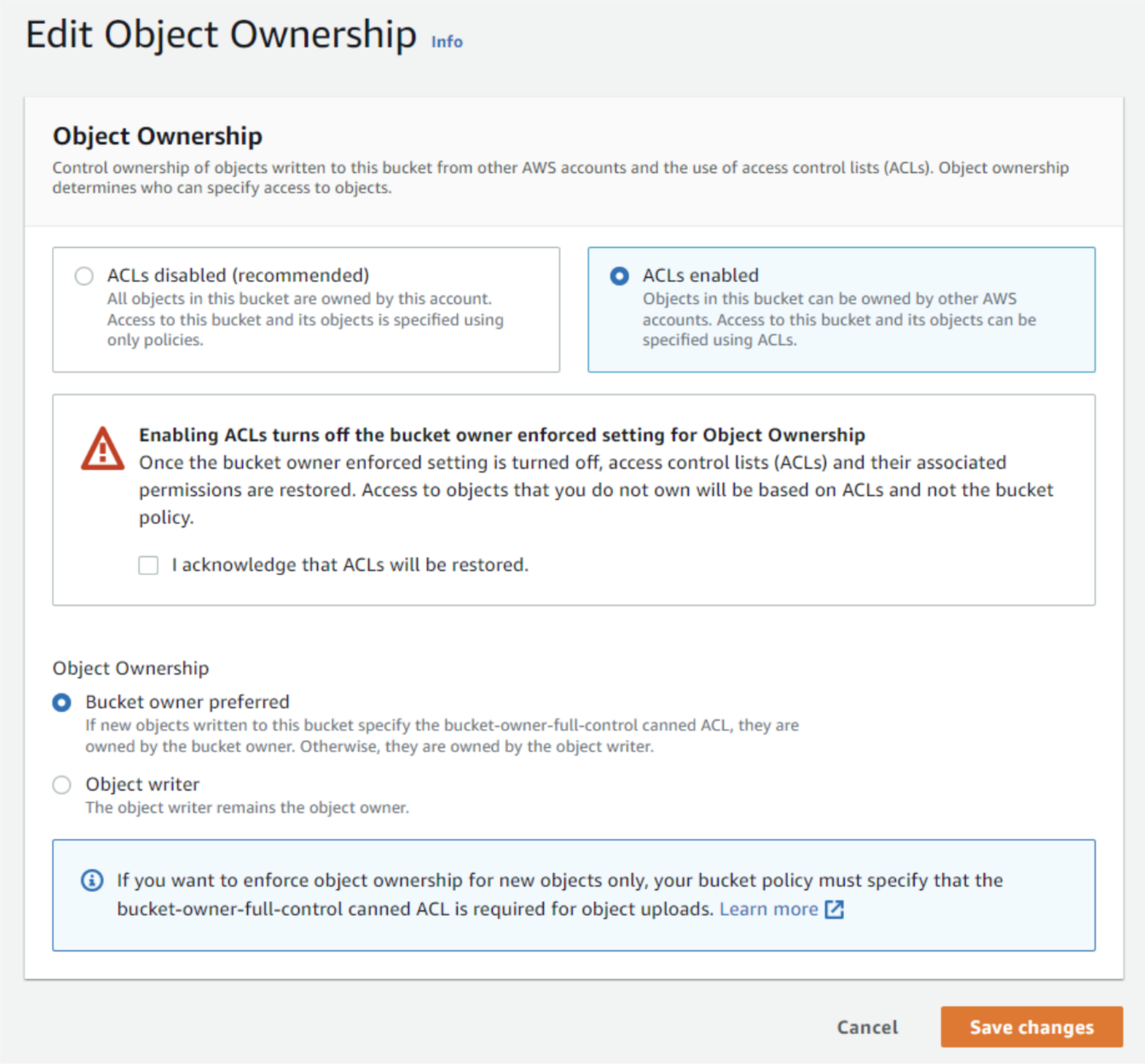
```
"Resource": "arn:aws:s3:::airsoft-workshop/*",
```

now inside permissions, edit access control list (ACL)
set the Everyone (Public Access)
Objects List to tick



Successfully edited bucket policy.

Easily transfer data between AWS Storage services with AWS DataSync.



Edit Object Ownership Info

Object Ownership

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

○ ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

● ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

⚠ **Enabling ACLs turns off the bucket owner enforced setting for Object Ownership**
Once the bucket owner enforced setting is turned off, access control lists (ACLs) and their associated permissions are restored. Access to objects that you do not own will be based on ACLs and not the bucket policy.

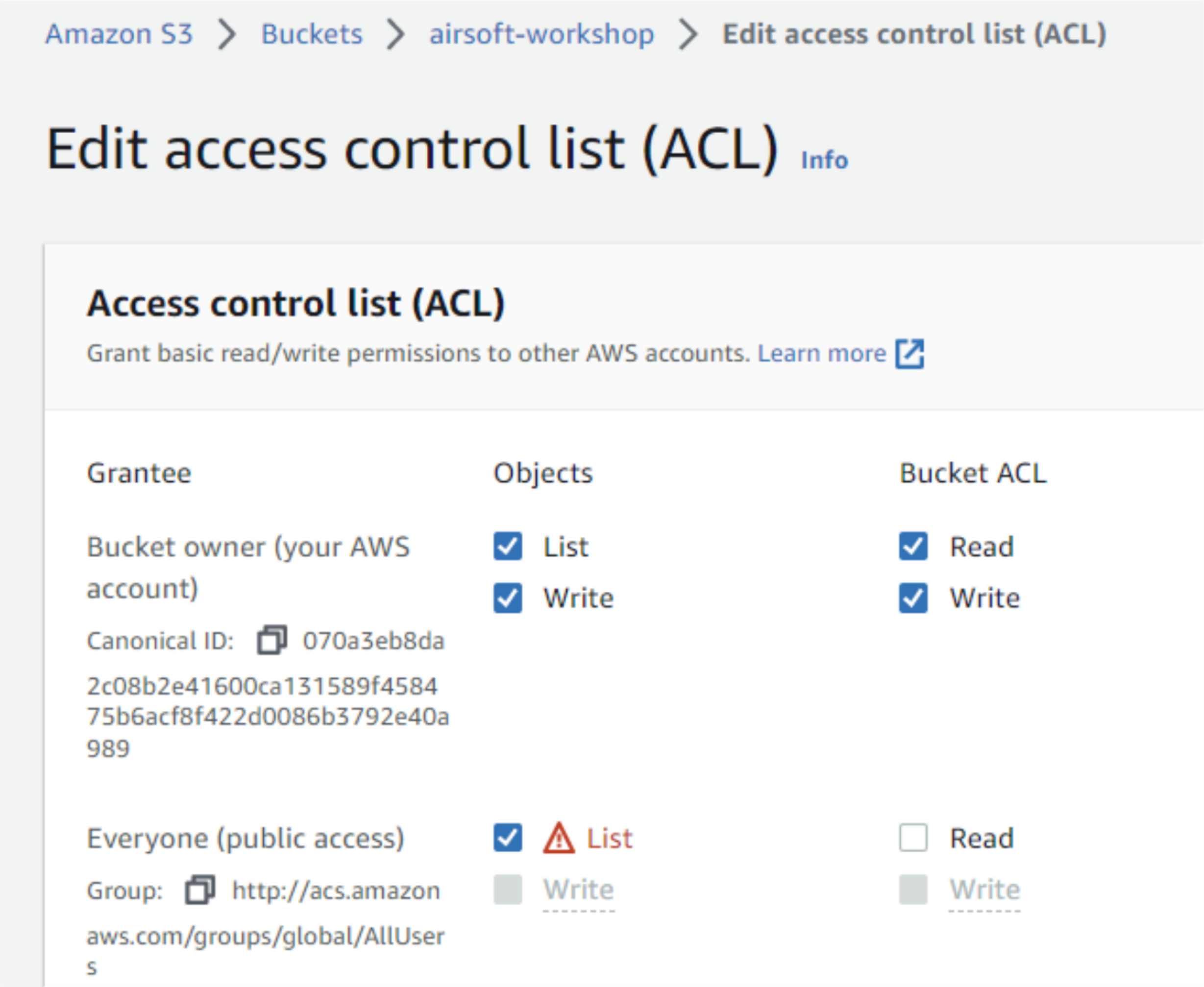
□ I acknowledge that ACLs will be restored.

Object Ownership

● Bucket owner preferred
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

○ Object writer
The object writer remains the object owner.

ⓘ If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control canned ACL is required for object uploads. Learn more



Amazon S3 > Buckets > airsoft-workshop > Edit access control list (ACL)

Edit access control list (ACL) Info

Access control list (ACL)

Grant basic read/write permissions to other AWS accounts. Learn more

Grantee	Objects	Bucket ACL
Bucket owner (your AWS account) Canonical ID: 070a3eb8da2c08b2e41600ca131589f458475b6acf8f422d0086b3792e40a989	☑ List ☑ Write	☑ Read ☑ Write
Everyone (public access) Group: http://acs.amazonaws.com/groups/global/AllUsers	☑ ⚠ List □ Write	□ Read □ Write

save changes. next we neeed to set up the IAM