Raw Data Sharing

S3 Bucket

Date	**2023-04-26**
Author	**Gopi**
Verified	**Dries Van Loon**
Revision	**V1**

Table of Contents

1. Data Sharing Using AWS S3

In order to share the data using S3 we use the Cross Account Access feature built into S3. More on the topic:

https://aws.amazon.com/premiumsupport/knowledge-center/cross-account-access-s3/

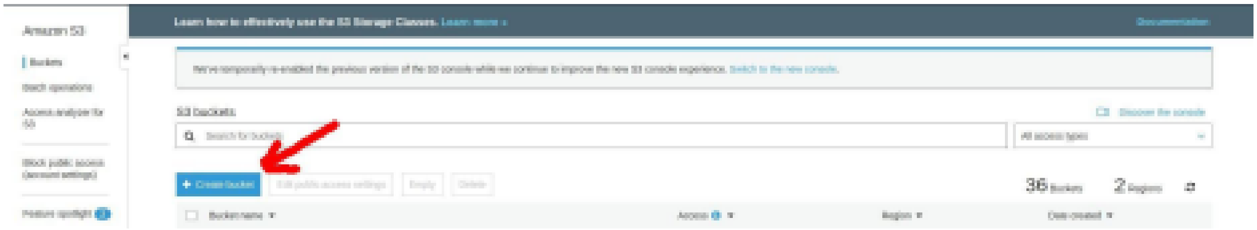
Benefits of using Cross Account Access

1. No credentials need to be shared between parties
2. Controlling security policies from within your own account
3. Owning the data that is stored in your bucket.

1.1. Steps to give Cross Account Access by Customer:

1. Create a bucket in your S3 account. You can choose any name that fits your naming convention.

a. Choose a name. After clicking on “Create bucket” button

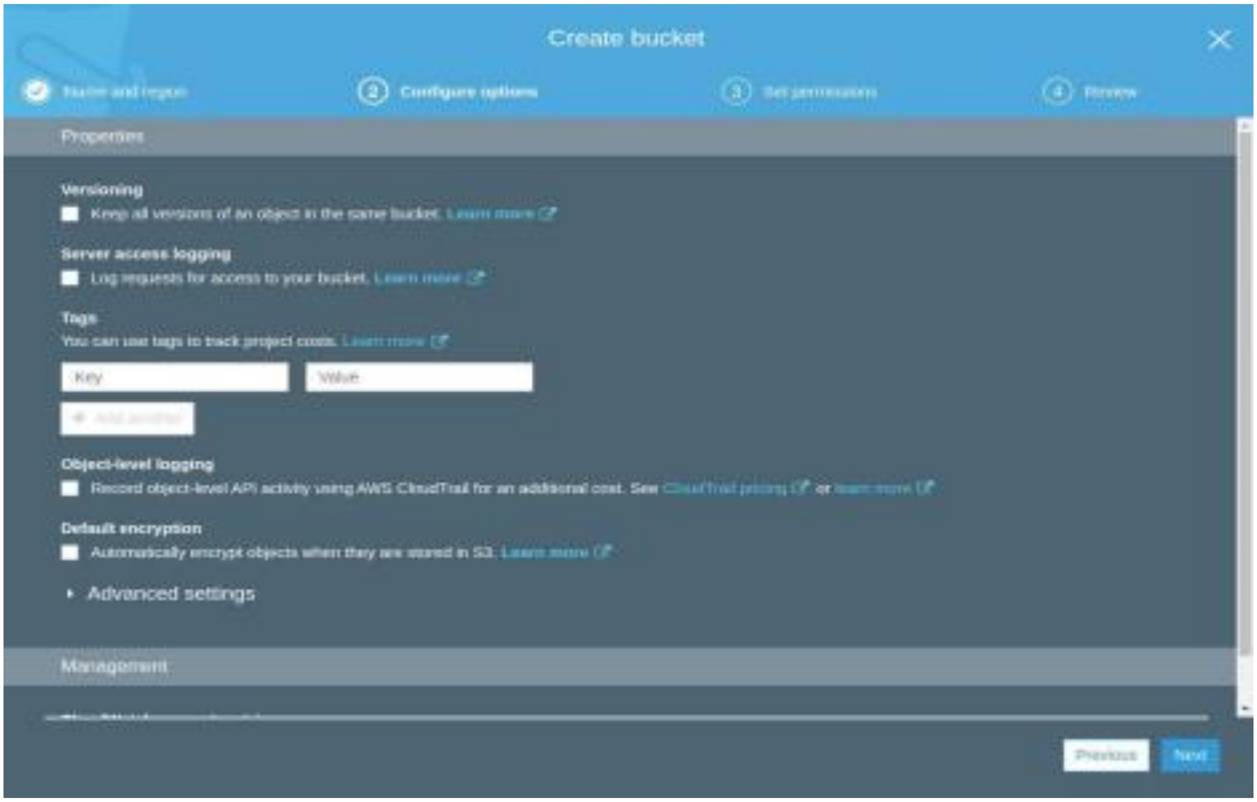


Learn how to effectively use the S3 Storage Classes. Learn more »

We've temporarily re-enabled the previous version of the S3 console while we continue to improve the new S3 console experience. Switch to the new console.

36 Buckets 2 Regions

b. Configure your bucket. You can leave this with the default values if you don’t have any special need on the bucket.



Properties

Versioning

□ Keep all versions of an object in the same bucket. Learn more

Server access logging

□ Log requests for access to your bucket. Learn more

Tags

You can use tags to track project costs. Learn more

Key | Value

+ Add another

Object-level logging

□ Record object-level API activity using AWS CloudTrail for an additional cost. See CloudTrail pricing or learn more

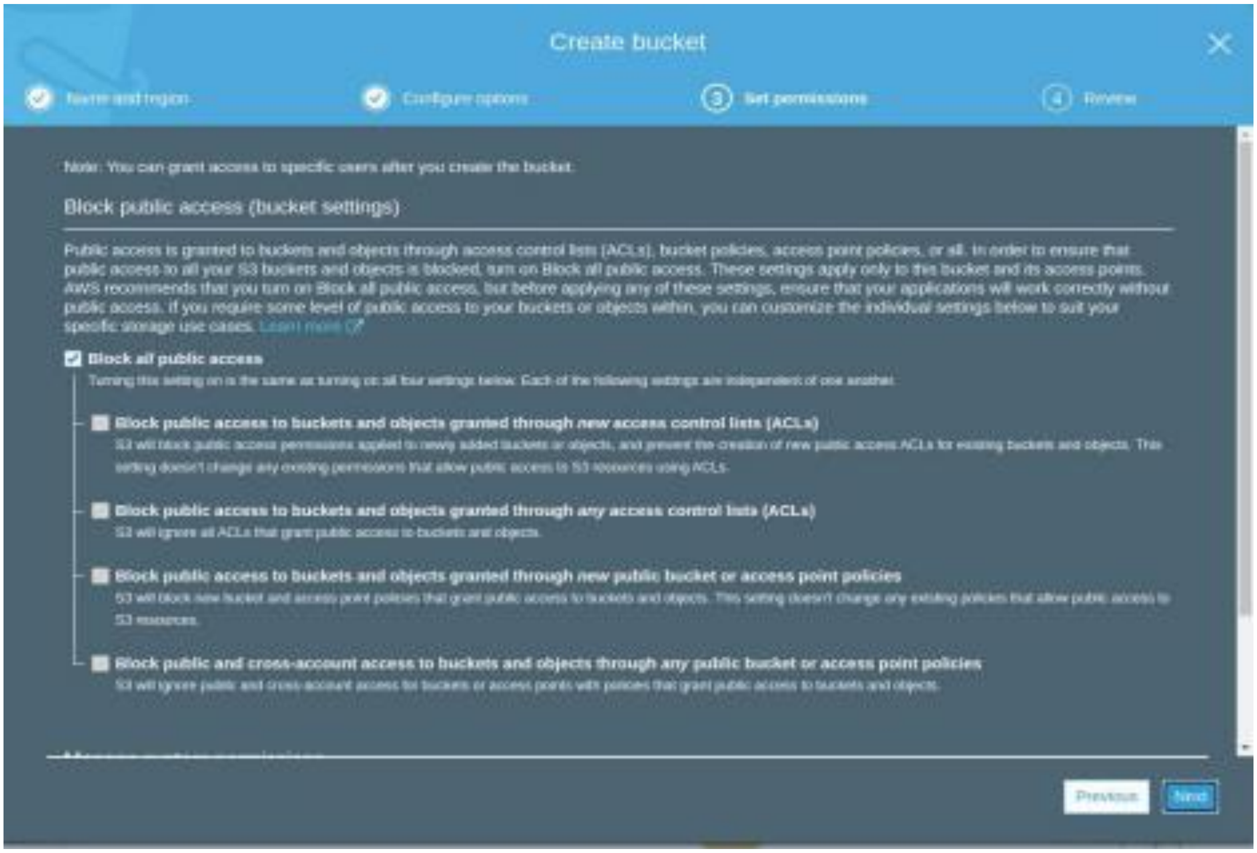
Default encryption

□ Automatically encrypt objects when they are stored in S3. Learn more

▸ Advanced settings

Management

c. Keep blocking all the public access on



Note: You can grant access to specific users after you create the bucket.

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

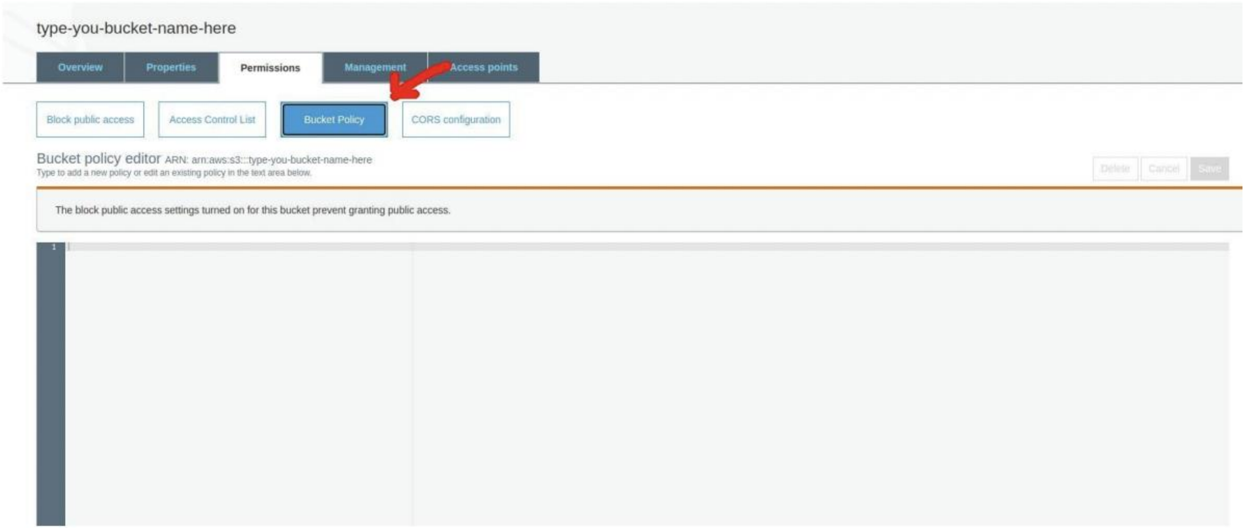
☑ **Block all public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- □ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
  S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- □ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
  S3 will ignore all ACLs that grant public access to buckets and objects.
- □ **Block public access to buckets and objects granted through *new* public bucket or access point policies**
  S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- □ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
  S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

d. Finalize and create the bucket

1.2. Changing the bucket policy

a. Editing bucket policy by clicking on “*Permission*” tab and “*Bucket Policy*”



type-you-bucket-name-here

Bucket policy editor ARN: arn:aws:s3:::type-you-bucket-name-here

Type to add a new policy or edit an existing policy in the text area below.

The block public access settings turned on for this bucket prevent granting public access.

b. Adding a policy similar to the following:

```
{
    "Version": "2012-10-17",
    "Id": "Policy1592948336660",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::495516375372:role/bucketWriter"
            },
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": "arn:aws:s3:::<bucket-name>/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control"
                },
                "IpAddress": {
                    "aws:SourceIp": "35.164.136.18"
                }
            }
        }
    ]
}
```

1.3. Understanding this policy:

1. Principal

```
"Principal": {
  "AWS": "arn:aws:iam::495516375372:role/bucketWriter"
}
```

Will make sure that the bucketWriter role in Nanoprecise account, determined by the account number: 495516375372 will have access to this bucket.

2. Allowed Actions

```
"Action": [
  "s3:PutObject",
  "s3:PutObjectACL"
 ],
```

This will make that Nanoprecise can only put a new object and set the access control list (ACL) permissions for a new or existing object in an S3 bucket

3. Conditions

```
"Condition": {
  "StringEquals": {
    "s3:x-amz-acl": "bucket-owner-full-control"
  },
  "IpAddress": {
    "aws:SourceIp": [
    "35.164.136.18"
    ]
  }
}
```

This will make sure that all the objects put in the bucket will have the bucket owner full control option on. Also this will whitelist the Nanoprecise IP address that can make put the object in the customer s3 bucket.

Please Ask us to provide you the IP address that needs to be whitelisted, as well as the role to grant access to.