😊 IAM POLICY- ACCESS TO LIST BUCKETS

IAM, or Identity and Access Management, is a framework that helps organizations manage and control access to their resources in cloud environments. An IAM policy is a set of rules that define the permissions and access control for entities such as users, groups, and roles within an IAM system.

In cloud computing platforms like Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure, IAM policies are used to specify what actions users, groups, or roles are allowed or denied to perform on various resources. These resources can include virtual machines, databases, storage, and other cloud services.

IAM policies are typically written in a specific syntax or using a visual interface provided by the cloud platform. They consist of statements that define the following:

1. **Effect:** Whether the policy allows or denies the specified actions.
2. **Action:** The specific actions or operations that are allowed or denied.
3. **Resource:** The resources on which the actions can be performed (e.g., specific buckets in a storage service, instances in a compute service).
4. **Condition:** Optional conditions that must be satisfied for the policy to take effect.

Here's a simplified example of an IAM policy in AWS that allows a user to list the contents of a specific S3 bucket:

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": "s3:ListBucket",
   "Resource": "arn:aws:s3:::example-bucket",
   "Condition": {}
  }
 ]
}
```

In this example:

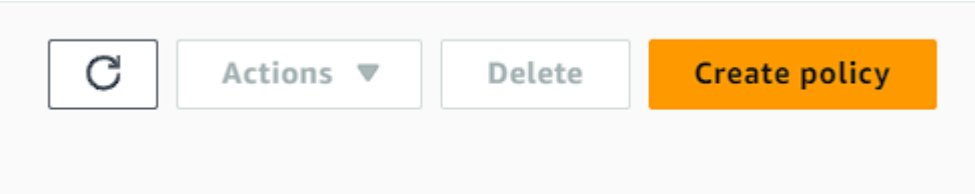
- **Effect:** Allows the action.
- **Action:** `s3:ListBucket` allows listing the contents of an S3 bucket.

- **Resource:** Specifies the ARN (Amazon Resource Name) of the target S3 bucket.
- **Condition:** No additional conditions are specified in this simple example.

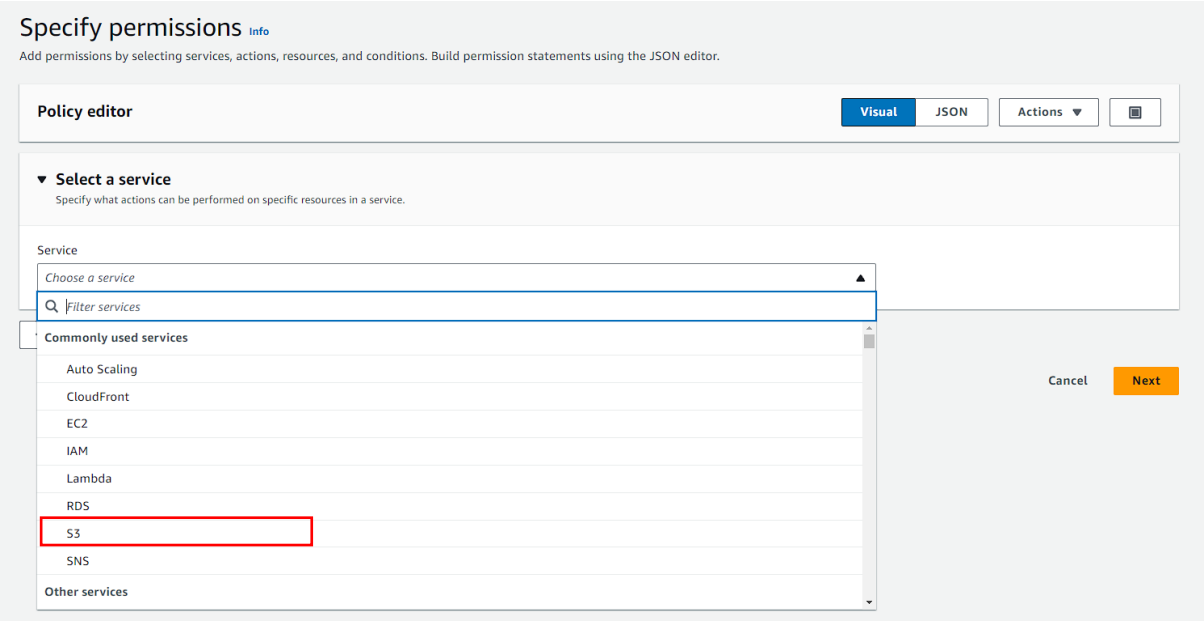
IAM policies play a crucial role in ensuring that users and systems have the appropriate level of access to resources while maintaining security and compliance in cloud environments.

😁 TO BEGIN WITH THE LAB

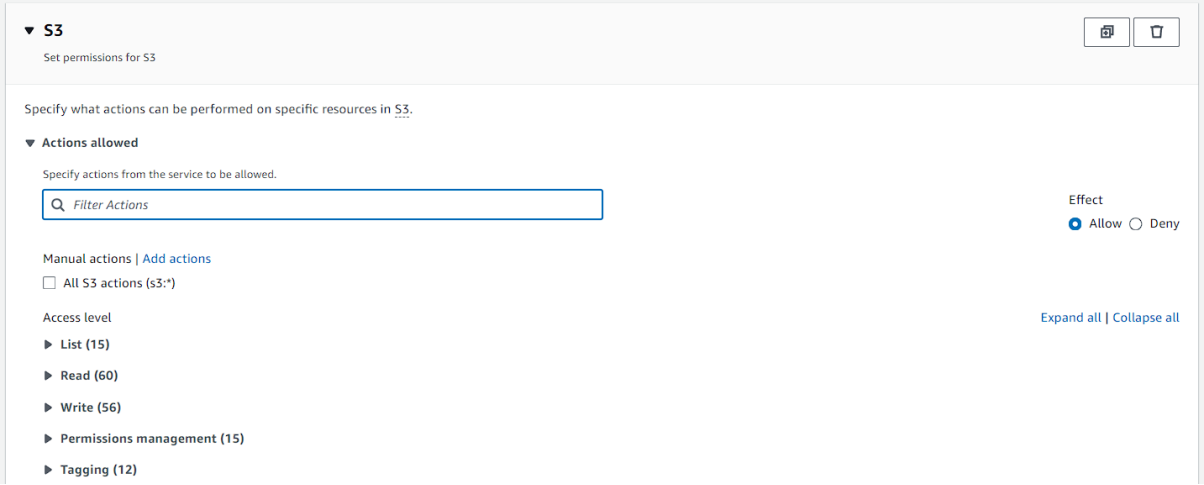
1. Login to AWS Console.
2. Then navigate IAM and remove the S3 read only access policy from your user.
3. In this lab you are going to create your own IAM Policy and attach it with the IAM user that you created earlier.
4. Once you have removed the policy from your IAM user, it's time to navigate to policies.
5. There you need to click on create policy.



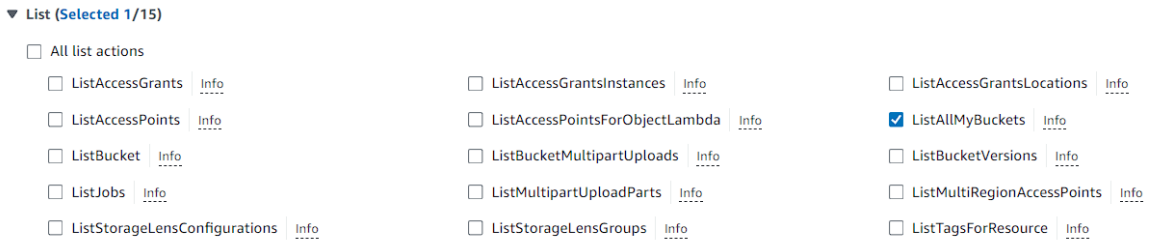
6. In the create policies tab you will see that you have an option to choose the service prior to write your own JSON code.



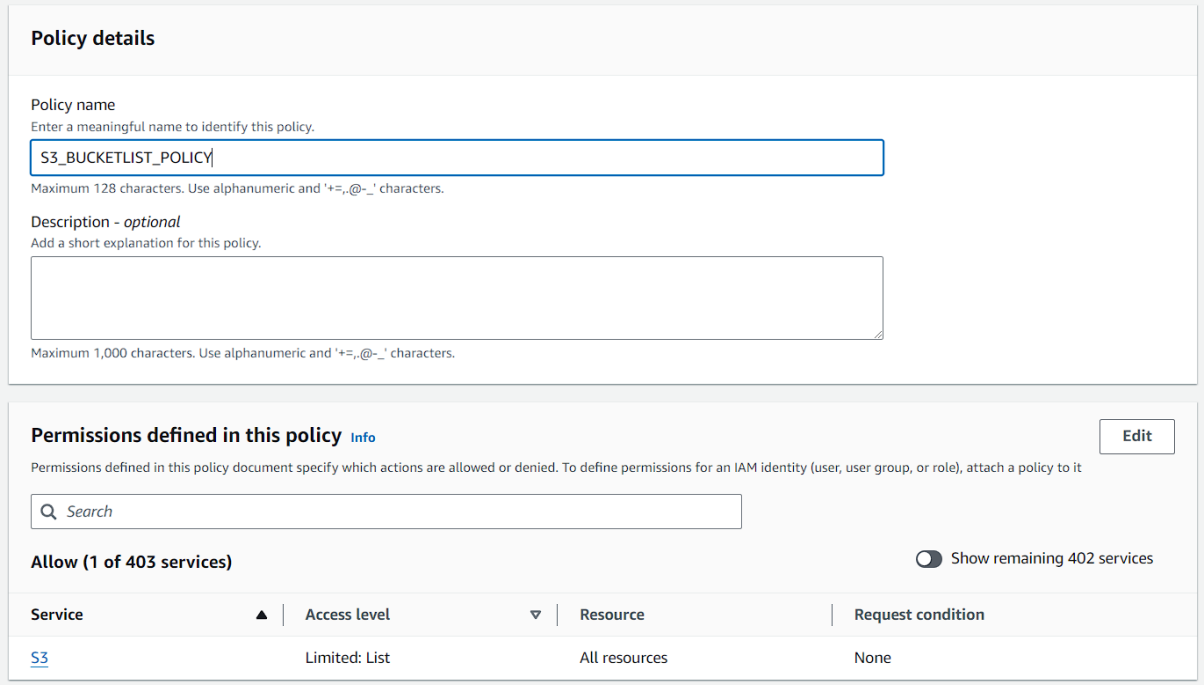
7. So, here you can see S3 service in place choose it.
8. After choosing S3 you will that you have different action to choose from.
9. If you will expand each of the action one by one you will see you have a lot options through which you can create your own policies.



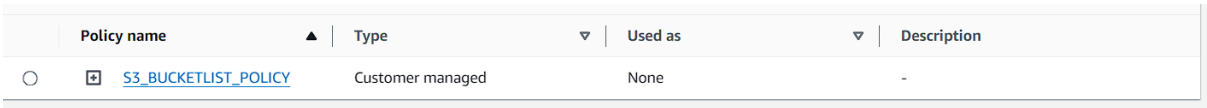
10. But for now, expand List, you will a bunch of different options which you can choose but you have to select List all my buckets.



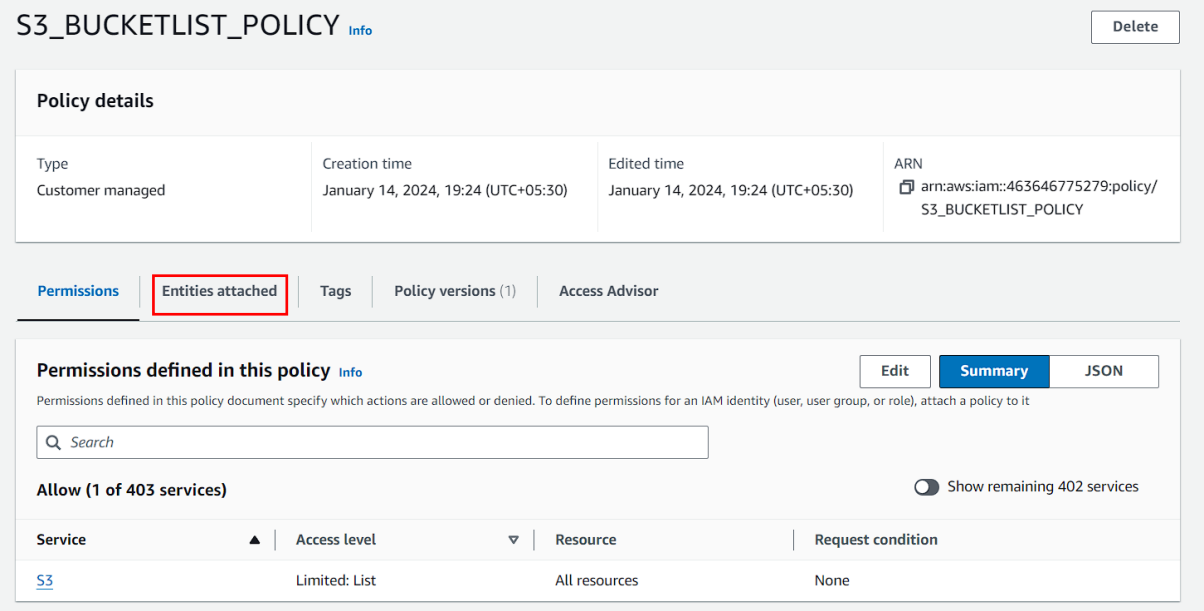
11. Then just go to next page and name your policy then create it.



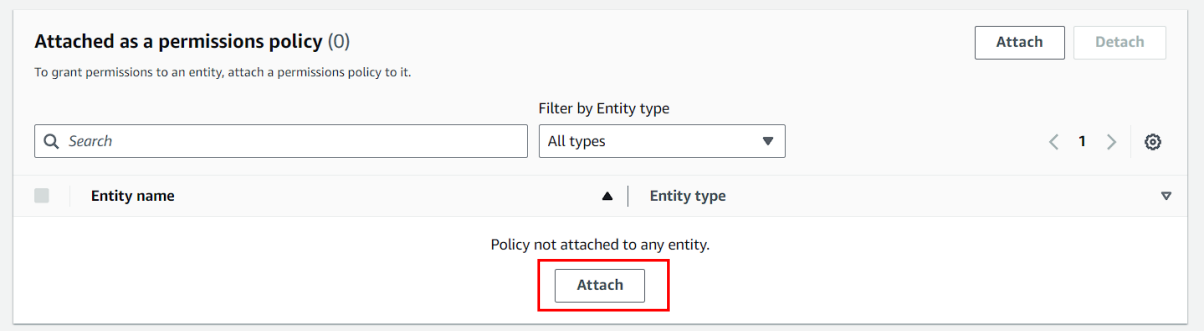
12. Here you can see your policy.



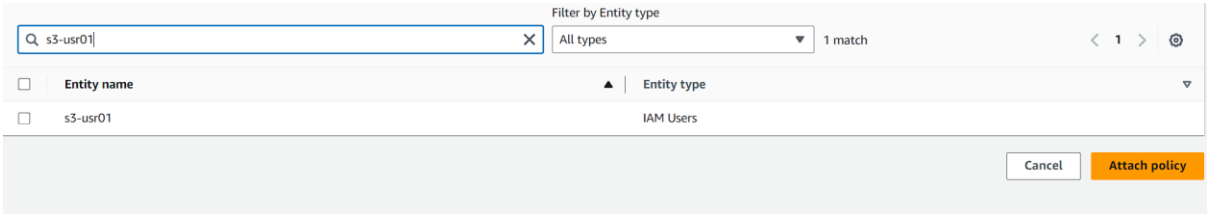
13. Now open your policy. Then click on Entities attached.



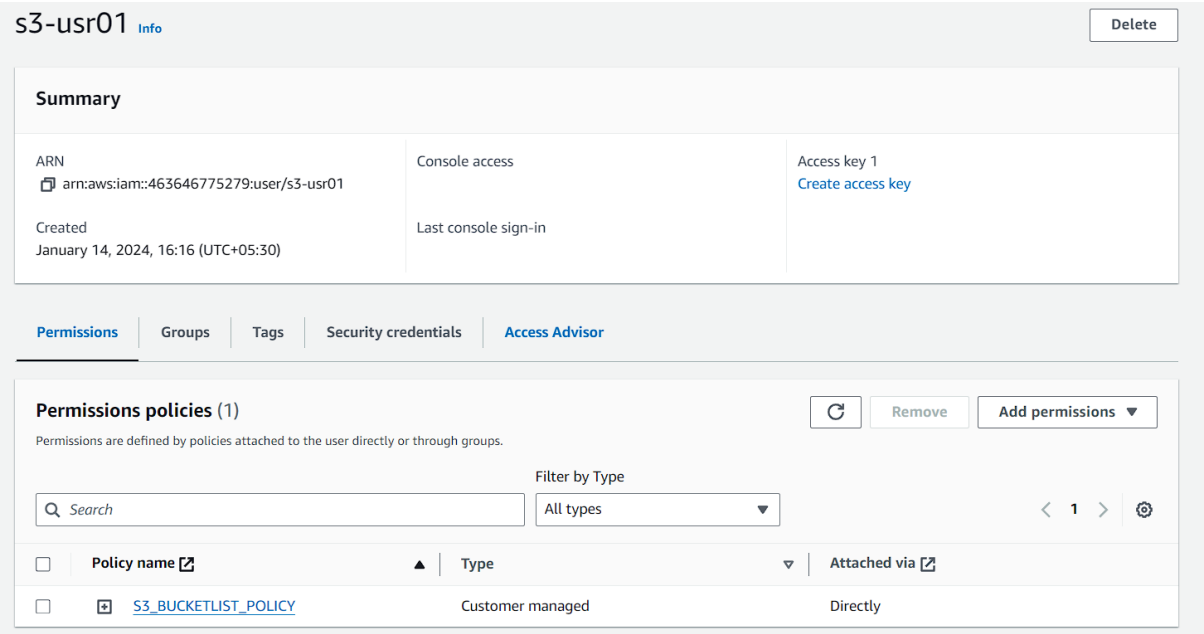
14. Now you have to click on attach.



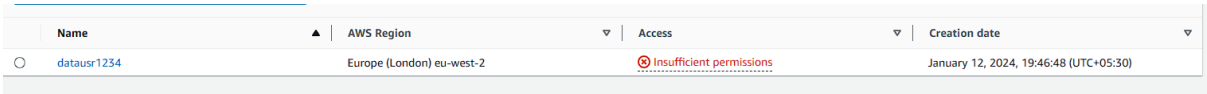
15. Here you to search for your IAM user account name. Then click on Attach policy.
16. This will attach your policy directly to your IAM user.
17. This is another method to attach policies to your IAM users.

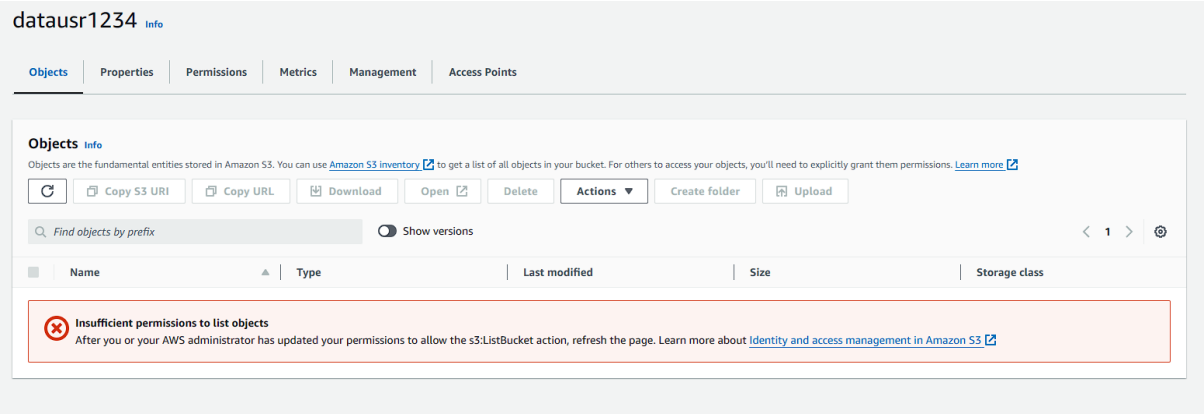


18. Just to cross verify, if you go back to your IAM user in IAM then you will see that the policy is attach to your user.



19. If you again login with you IAM user account in any other browser.
20. Then navigate to S3, you will see that you can see your bucket but the it is also showing you the you have insufficient permissions.





21. This is happening because the policy that you created, was just for listing the bucket not reading or making any changes to them.