

PROJECT REPORT
ON
“IAM User Policies in AWS:
AmazonS3FullAccess and Deny Policy”
IN



185, Zone-I, Maharana Pratap Nagar, Bhopal, Madhya Pradesh 462011

SUBMITTED TO:

NAME - ABHISHEK KORI
BATCH - B9IT
COURSE - CYBER SECURITY

SUBMITTED BY:

YASH SIR

Report on IAM User Policies in AWS: AmazonS3FullAccess and Deny Policy

Summary:: AWS provides IAM policies to manage permissions for various entities like users, groups, and roles. These policies can either grant or deny access to AWS resources. AWS uses an explicit deny rule to override grants if multiple policies are applied to a user. For instance, consider a scenario where an IAM user is assigned the AmazonS3FullAccess policy, which enables full access to Amazon S3 resources but also has another policy with explicit deny rules. In such cases, the explicit deny rule takes precedence over the permission granted by the AmazonS3FullAccess policy.

AWS Identity and Access Management (IAM)

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. With IAM, you can centrally manage permissions that control which AWS resources users can access. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

Policies and permissions in IAM

You manage access in AWS by creating policies and attaching them to IAM identities (users, groups of users, or roles) or AWS resources. A policy is an object in AWS that defines its permissions when associated with an identity or resource. AWS evaluates these policies when an IAM principal (user or role) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. AWS supports six types of policies: identity-based policies, resource-based policies, permissions boundaries, Organizations SCPs, ACLs, and session policies.

IAM policies define permissions for action regardless of the method that you use to operate. For example, suppose a policy allows the Getuser action. In that case, a user with that policy can get user information from the AWS Management Console, the AWS CLI, or the AWS API.

What is Amazon S3? And AmazonS3FullAccess,

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance.

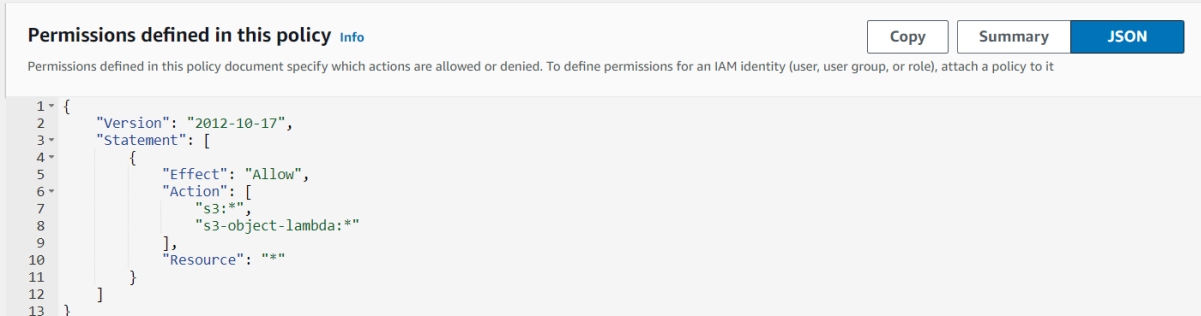
AmazonS3FullAccess is an AWS-managed policy that: Provides full access to all buckets via the AWS Management Console. Using this policy you can attach AmazonS3FullAccess to your users, groups, and roles.

Test Setup:

We conducted an experiment to test IAM policies in an AWS environment. We created a new IAM user 'abhishek' and attached two policies to their account.

1. AmazonS3FullAccess Policy:

This policy grants full access to Amazon S3 resources.



The screenshot displays the 'Permissions defined in this policy' section of the AWS IAM console. It includes a header with 'Copy', 'Summary', and 'JSON' buttons. Below the header, a JSON policy document is shown with line numbers 1 through 13 on the left. The policy grants full access to all Amazon S3 resources.

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Action": [  
7         "s3:*",  
8         "s3-object-lambda:*"  
9       ],  
10      "Resource": "*"   
11    }  
12  ]  
13 }
```

2. Deny Policy:

This policy denies access to the Amazon S3 buckets.

Permissions defined in this policy [Info](#)

Copy

Edit

Summary

JSON

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Deny",  
6       "Action": [  
7         "s3:*",  
8         "s3-object-lambda:*"  
9       ],  
10      "Resource": "*"   
11    }  
12  ]  
13 }
```

Experiment Results :

After testing the IAM user 'Abhishek'' with present policies, we observed the following outcomes:

1. AmazonS3FullAccess Policy:
The user was able to perform any action on any S3 resource, as expected.

General purpose buckets (1) [Info](#)

↺

Copy ARN

Empty

Delete

Create bucket

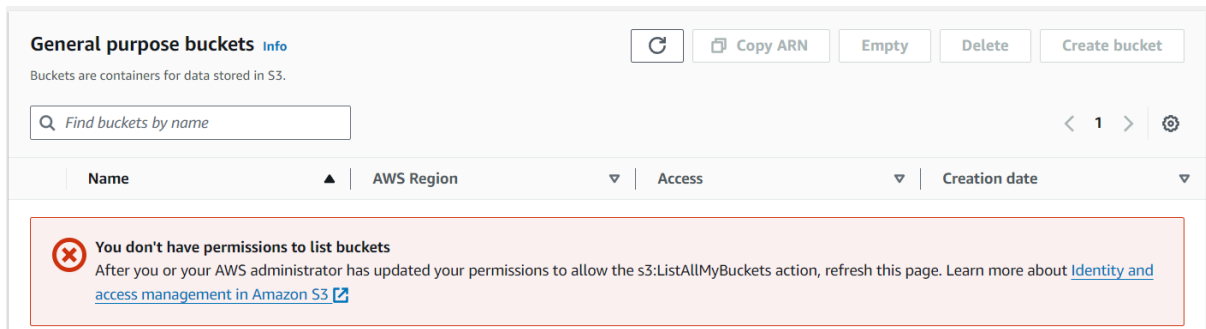
Buckets are containers for data stored in S3.

Find buckets by name

< 1 > ⚙

	Name	AWS Region	Access	Creation date
<input type="radio"/>	awsreport01	Asia Pacific (Mumbai) ap-south-1	<u>Bucket and objects not public</u>	March 4, 2024, 19:12:00 (UTC+05:30)

2. Deny Policy:
The user was denied access to the S3 resource in the deny policy.



Policy Evaluation Logic:

AWS IAM evaluates policies using a three-step process:

1. Identity-based policies:
AWS evaluates identity-based policies
(those directly attached to the user) first.
2. Resource-based policies:
AWS then evaluates resource-based policies
(those attached to the resource, e.g., S3 bucket policies)
3. Inline policies:
Finally, AWS evaluates inline policies
(those embedded directly into the user or group).

Conclusion:

When an IAM user in AWS has multiple policies applied, the principle of least privilege is always followed. In other words, if any explicit denies are present in the policies, they will take precedence over explicit allows. This means that even if a user is granted access to perform all S3 actions with the AmazonS3FullAccess policy, they may still be denied access to a specific S3 bucket if a deny policy exists for that bucket.