

Grant IAM User Access to Only One S3 Bucket or Folder Using IAM Policy

7 min read · Sep 20, 2021



Zenesys Solution Inc.

Follow



Listen

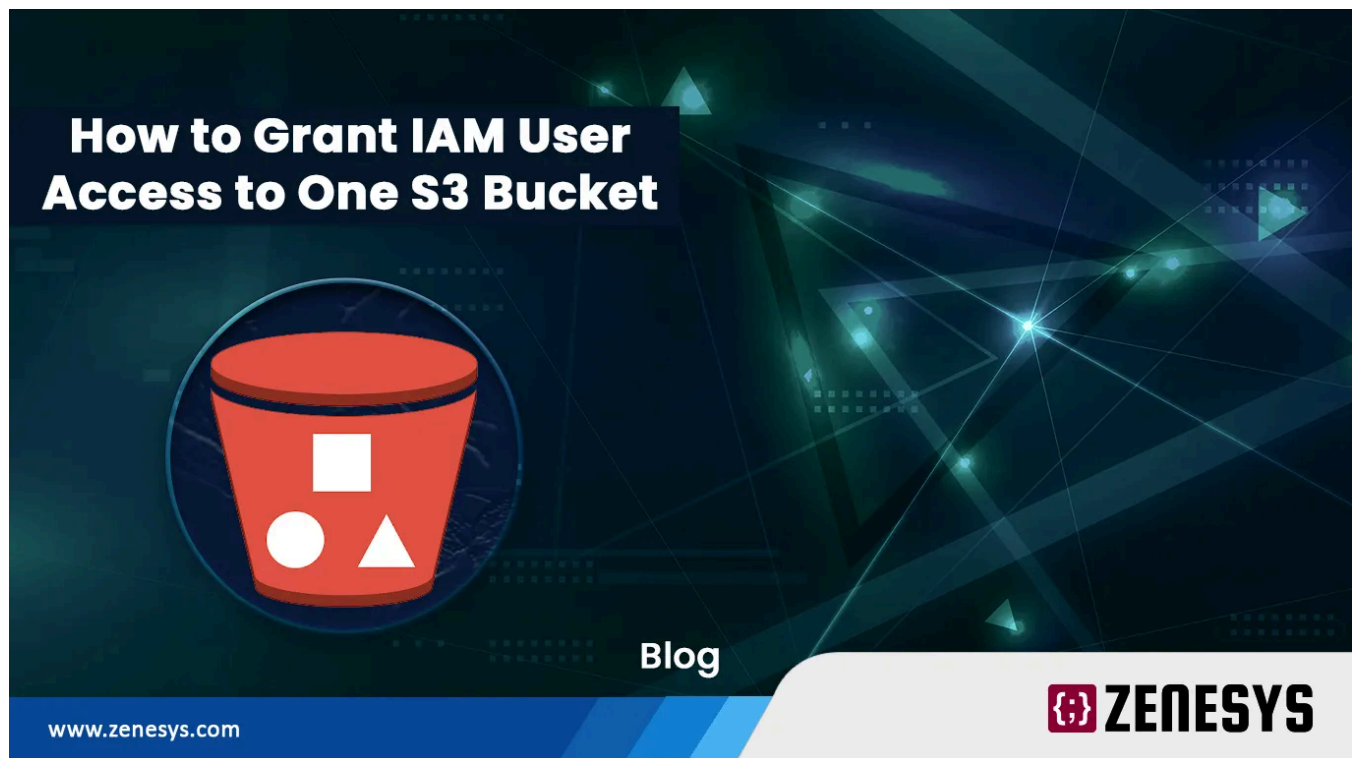


Share



More

In this guide , We will learn to configure an IAM policy using which we can provide access to an IAM user for a Specific S3 Bucket and/or folder within the S3 Bucket.



How to Grant IAM User Access to Only One S3 Bucket

Table of Contents

- What is IAM?
- What is the IAM Policy?
- Creating IAM Policy — Granting Single bucket Access

- Creating IAM user & Assigning IAM Policy
- Creating IAM Role & Attaching IAM Policy
- Grant IAM user access to a Folder in S3 Bucket

What is IAM?

IAM stands for Identity and Access Management. IAM is a web service Which provides authentication and authorization to the users and resources hosted in the Amazon Web Service.

What is the IAM Policy?

We can manage access in AWS by creating policies and attaching them to IAM identities such as users, groups of users, or roles and AWS resources.

A policy defines a permission which can be attached to the resources and the users.

When the IAM principal (user or role) makes a request, every time AWS evaluate those policies. Permissions determine access and the policies are stored as JSON documents.

Supported policy types:

- Identity-based
- Resource-based
- permissions boundaries
- Organizations SCPs
- ACLs
- session policies.

When you create an IAM user, you can choose to allow console or programmatic access. If console access is allowed, the IAM user can sign into the console using a username and password. Or if programmatic access is allowed, the user can use access keys to work with the CLI or API.

Creating IAM Policy

The IAM policy that we are going to create grants full access for an IAM user to the Single S3 bucket , so that he / she can manage the files and folders within the specified S3 bucket.

Login to IAM Console.

From the navigation pane , Choose Policies

Dashboard

▼ Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

IAM Console Dashboard

There are two types of policies:

1. The policies that are managed by AWS are referred to as the **AWS managed Policy**.
2. The Policies that we create are called **Customer managed Policy**.

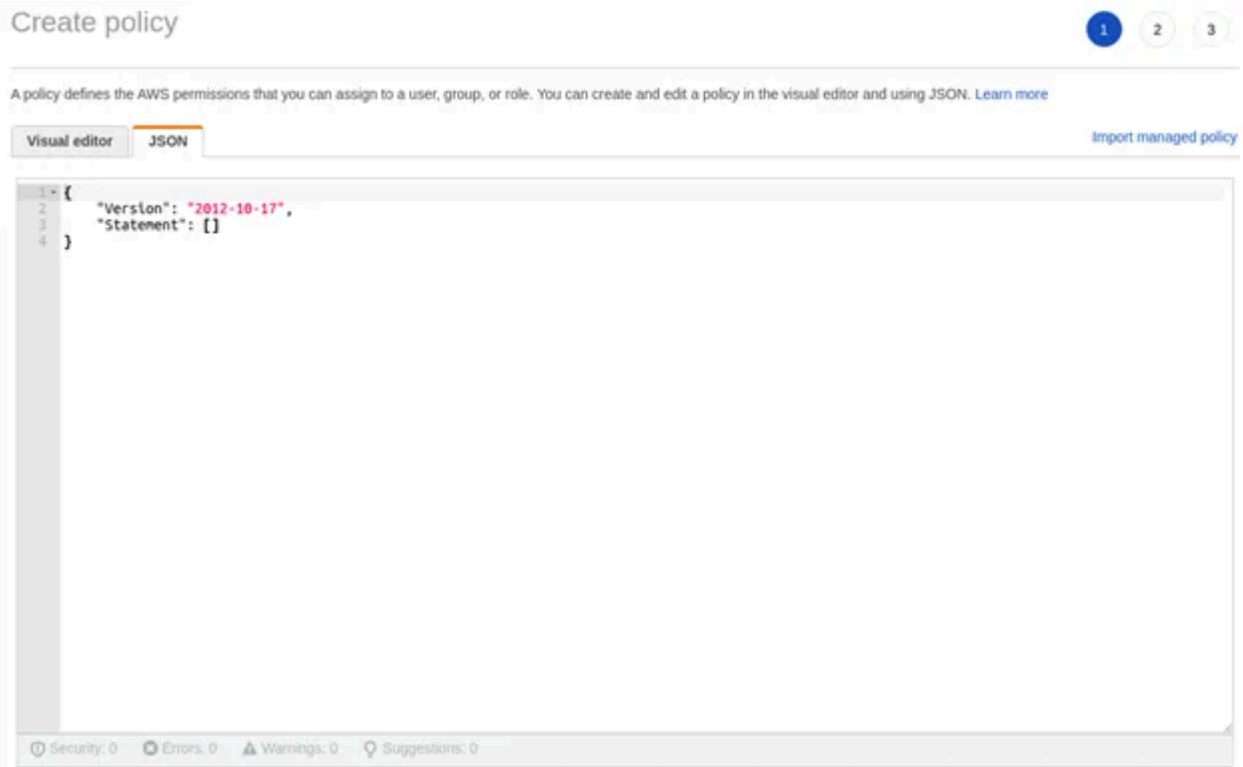
Let us implement an IAM policy which will grant an IAM user full access to that specific S3 bucket.

For this tutorial , Let's say you have the bucket named as **singles3bucketaccess**



Bucket Name

To Create policy → **Create policy** , and click **JSON**.



Create policy

In the JSON column, Remove the existing policy and add the below policies.

```

{

  "Version": "2021-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets"
      ],
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::singles3bucketaccess",
        "arn:aws:s3:::singles3bucketaccess/*"
      ]
    }
  ]
}

```

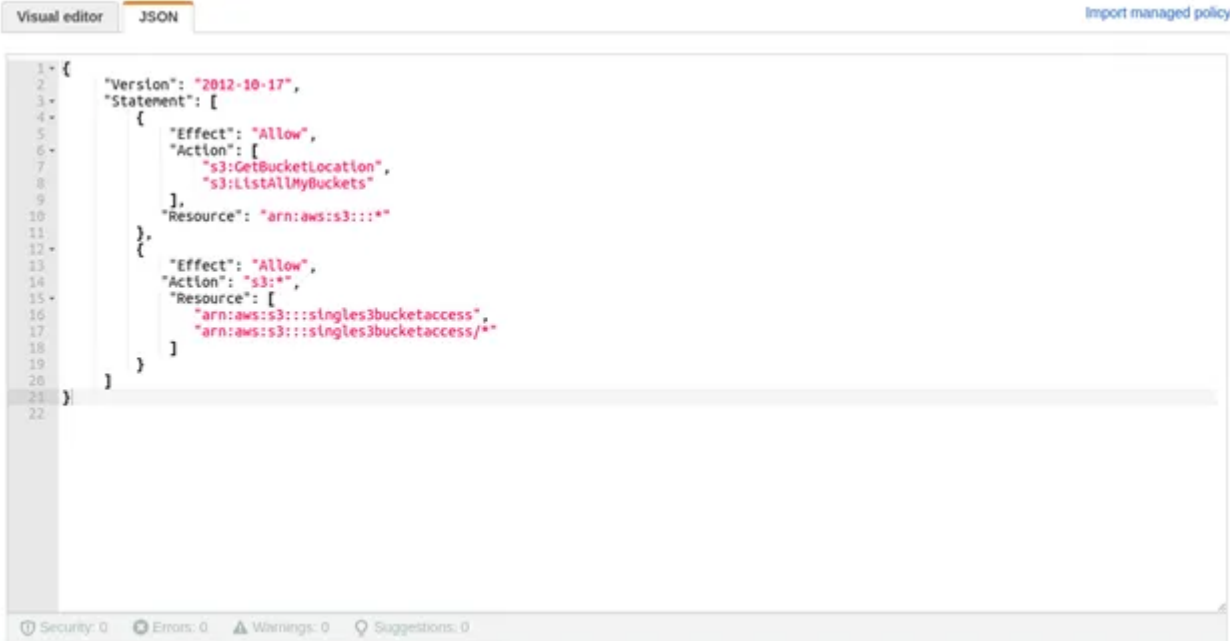
```

}
]
}

```

The first half of the IAM policy grants permission for an IAM user to list all the available S3 buckets / S3 console.

The second half of the IAM policy grants permission for an IAM user to access all the files and folders created within the S3 bucket : singles3bucketaccess.



```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "s3:GetBucketLocation",
8         "s3:ListAllMyBuckets"
9       ],
10      "Resource": "arn:aws:s3:*"
11    },
12    {
13      "Effect": "Allow",
14      "Action": "s3:*",
15      "Resource": [
16        "arn:aws:s3::singles3bucketaccess",
17        "arn:aws:s3::singles3bucketaccess/*"
18      ]
19    }
20  ]
21 }
22

```

S3 bucket

Instead of granting full access to the S3 bucket , We can provide specific access to a S3 bucket action such as **GetObject** , **ListJobs** etc.

Click Next: Tags , Next: Review

Provide a name and click **Create Policy**.

Review policy

Name*
Use alphanumeric and '+', '@', '-' characters. Maximum 128 characters.

Description
Maximum 1000 characters. Use alphanumeric and '+', '@', '-' characters.

Summary

This policy defines some actions, resources, or conditions that do not provide permissions. To grant access, policies must have an action that has an applicable resource or condition. For details, choose [Show remaining](#). [Learn more](#)

Q Filter

Service ▾	Access level	Resource	Request condition
Allow (1 of 284 services) Show remaining 283			
S3	Limited: List, Read, Write, Permissions management, Tagging	Multiple	None

Tags

Key	Value
No tags associated with the resource.	

* Required Cancel Previous [Create policy](#)

Review Policy

We have created the required IAM policy.

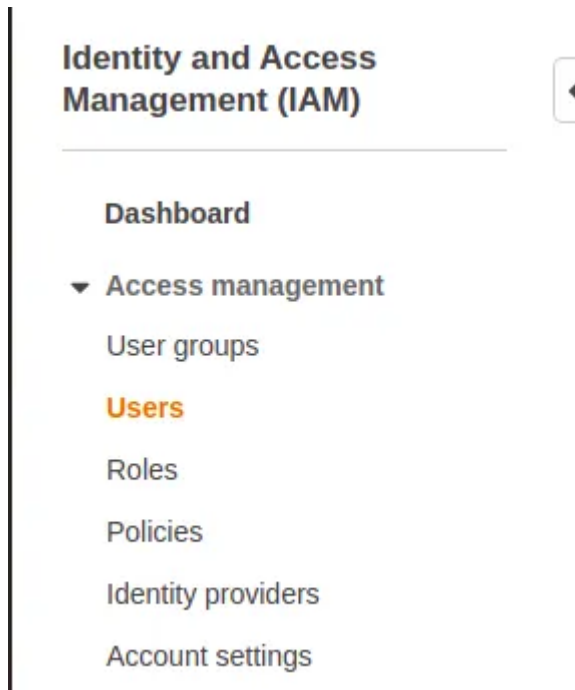
We can now assign this IAM policy to an IAM user so that he/she can access the specified S3 bucket.

Also read: [Creating EC2 Instances using Terraform](#)

Creating IAM User & Assigning the IAM Policy

Now It's time to attach the policy that we have created to an IAM user

To create IAM user → Choose **Users**



IAM dashboard

Click **Add user**

Enter a name for the IAM user and choose the type of AWS access they require.

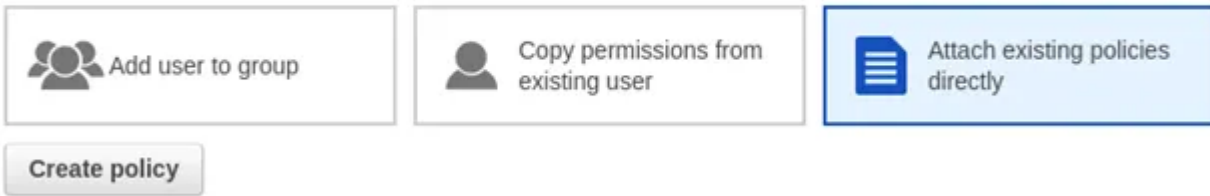
The image shows the "Set user details" form in the AWS IAM console. It has a title "Set user details" and a subtitle "You can add multiple users at once with the same access type and permissions. [Learn more](#)". There is a text input field for "User name*" containing the text "singles3bucketaccess". Below the input field is a blue button with a plus icon and the text "Add another user". The next section is "Select AWS access type" with a subtitle "Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)". Under "Access type*", there are two radio button options: "Programmatic access" (unchecked) and "AWS Management Console access" (checked). The "Programmatic access" option has a description: "Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools." The "AWS Management Console access" option has a description: "Enables a **password** that allows users to sign-in to the AWS Management Console." Under "Console password*", there are two radio button options: "Autogenerated password" (unchecked) and "Custom password" (checked). Below the "Custom password" option is a password input field with a masked password "*****" and a "Show password" checkbox (unchecked). At the bottom, there is a "Require password reset" checkbox (unchecked) with a description: "User must create a new password at next sign-in. Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password."

User details

Click **Next: Permissions**

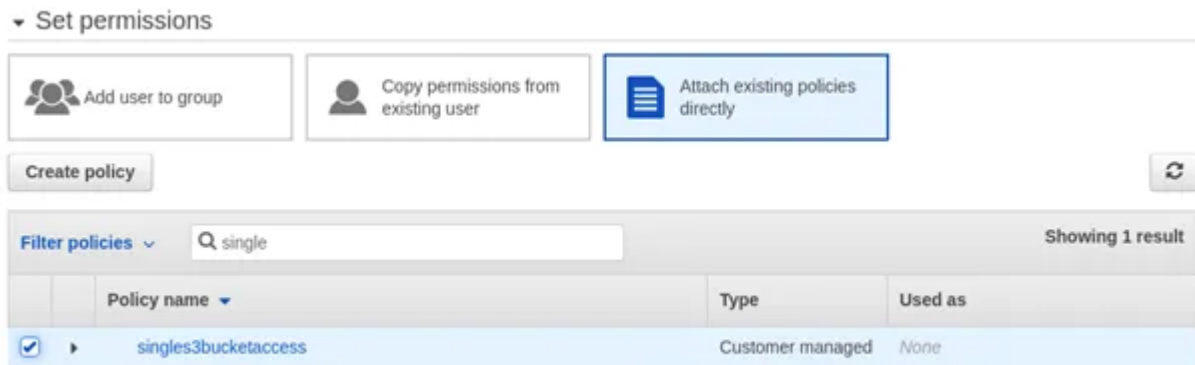
As we already have created the IAM policy , Choose **Attach existing policies directly**

Set permissions



Set Permission

Search for the IAM policy which was created before and then select it.



Set IAM policy

Click **Next: Review** → Choose **Create user**

Once the user is created with the required permissions, With the help of Access Keys or AWS Console (Depending on the access type assigned for that user), They can manage the S3 bucket and the files and folders inside the bucket.

Attaching IAM Policy to IAM Role

The IAM Role acts as a medium between multiple AWS services.

Lets say, You have an EC2 instance which wants to communicate with the S3 bucket to perform certain operations such as uploading and downloading files.

In this case, We will create an IAM policy with the required permissions and then the policy will be attached to an IAM Role.

The IAM role which was created will then be attached to an EC2 instance to perform S3 operations.

To Create an IAM Role, Login to the IAM Console.

From the Navigation pane , Choose **Roles**.

Click **Create role**

Choose **AWS service** as trusted entity and choose **EC2** as common use case

The screenshot shows the 'Create role' page in the AWS IAM console. The title is 'Create role' with a progress indicator showing step 1 of 4. The section is 'Select type of trusted entity'. There are four options: 'AWS service' (selected), 'Another AWS account', 'Web identity', and 'SAML 2.0 federation'. Below the options, there is a description: 'Allows AWS services to perform actions on your behalf. [Learn more](#)'. The next section is 'Choose a use case', with 'Common use cases' listed: 'EC2' (selected) and 'Lambda'.

Create Role-AWS service

Click **Next: Permissions**

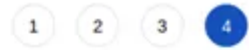
Choose the IAM policy that was created before

The screenshot shows the 'Create role' page in the AWS IAM console, step 2 of 4: 'Attach permissions policies'. The section is 'Attach permissions policies' with a dropdown arrow. Below it, there is a text: 'Choose one or more policies to attach to your new role.' There is a 'Create policy' button and a refresh icon. Below that, there is a search bar with 'Filter policies' and a search input containing 'single'. The results show 'Showing 1 result'. The table below has columns 'Policy name' and 'Used as'. The row shows a checkbox, a right arrow, the policy name 'singles3bucketaccess', and 'Permissions policy (1)'.

Create permission

Click **Next: Tags** , **Review** and then enter the name for the IAM role and click **Create role**.

Create role



Review

Provide the required information below and review this role before you create it.

Role name*

Use alphanumeric and '+,=, @, -, _' characters. Maximum 64 characters.

Role description

Maximum 1000 characters. Use alphanumeric and '+,=, @, -, _' characters.

Trusted entities AWS service: ec2.amazonaws.com

Policies Policies not attached

Permissions boundary Permissions boundary is not set

No tags were added.

Create Review

This IAM role can be attached with the EC2 instances so that the Instances can securely access the S3 bucket to perform S3 operations such as uploading and downloading files.

Grant IAM user access to a folder in S3 bucket

Let's set up a custom IAM policy which grants access to specific folders within the S3 bucket.

Use case : Let's assume you have lists of users who want to upload and download files from their respective folders within the S3 bucket.

In this case , We have to create a custom policy for each user allowing access only to their respective folders.

The first section of the policy allows the users to access the S3 console and lists the S3 buckets.

This is the minimum permission required for a user to access or list the S3 bucket.

Policy 1 : List all the S3 Buckets

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

```

“Action”: [
“s3:GetBucketLocation”,
“s3:ListAllMyBuckets”
],
“Resource”: “arn:aws:s3::*”
},

```

Next , The user should have the permission to list all the folders within the S3 bucket.

Policy 2 : List Folders in S3 Bucket

Replace Bucketname and Foldername in the below policy

```

{
“Sid”: “Statement1”,
“Action”: [“s3:ListBucket”],
“Effect”: “Allow”,
“Resource”: [“arn:aws:s3:::Bucketname”],
“Condition”:{“StringEquals”:{“s3:prefix”:[“”,“Foldername”]}}
}
},

```

Next is to create a policy which allows the user to list all the files within the folder.

You May Also Like: [Tracking S3 Bucket Changes using Lambda Function](#)

Policy 3 : List Files in a Folder

Replace Bucketname and Foldername in the below policy

```

{
“Sid”: “Statement2”,
“Action”: [“s3:ListBucket”],
“Effect”: “Allow”,
“Resource”: [“arn:aws:s3:::Bucketname”],
“Condition”:{“StringLike”:{“s3:prefix”:[“Foldername/*”]}}
}
},

```

The Final policy is to provide actual permissions the users can perform on the files within the Folder in a S3 bucket such as upload , download , delete etc.

Policy 4 : Permission to Manage Objects in S3 Folder

Replace **Bucketname** and **Foldername** in the below policy

```
{
  "Sid": "Statement3",
  "Effect": "Allow",
  "Action": ["s3:*"],
  "Resource": ["arn:aws:s3:::Bucketname/foldername/*"]
}
```

All the above policies can be merged into a single custom IAM policy and then it can be assigned to a User , Role etc.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets"
      ],
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Sid": "Statement1",
      "Action": ["s3:ListBucket"],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::singles3bucketaccess"],
      "Condition": {"StringEquals": {"s3:prefix": [",", "folder1"]}}
    }
  ]
}
```

```

“Sid”: “Statement2”,
“Action”: [“s3:ListBucket”],
“Effect”: “Allow”,
“Resource”: [“arn:aws:s3:::singles3bucketaccess”],
“Condition”:{“StringLike”:{“s3:prefix”:[“folder1/*”]}}
},
{
“Sid”: “Statement3”,
“Effect”: “Allow”,
“Action”: [“s3:*”],
“Resource”: [“arn:aws:s3:::singles3bucketaccess/folder1/*”]
}
]
}

```

Create policy

1 2 3

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. [Learn more](#)

Visual editor

JSON

Import managed policy

```

1 - {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "s3:GetBucketLocation",
8         "s3:ListAllMyBuckets"
9       ],
10      "Resource": "arn:aws:s3:*"
11    },
12    {
13      "Sid": "Statement1",
14      "Action": ["s3:ListBucket"],
15      "Effect": "Allow",
16      "Resource": ["arn:aws:s3:::singles3bucketaccess"],
17      "Condition": {"StringEquals": {"s3:prefix": ["", "folder1"]}}
18    },
19    {
20      "Sid": "Statement2",
21      "Action": ["s3:ListBucket"],
22      "Effect": "Allow",
23      "Resource": ["arn:aws:s3:::singles3bucketaccess"],
24      "Condition": {"StringLike": {"s3:prefix": ["folder1/*"]}}
25    },
26    {
27      "Sid": "Statement3",
28      "Effect": "Allow",
29      "Action": ["s3:*"],
30      "Resource": ["arn:aws:s3:::singles3bucketaccess/folder1/*"]
31    }
32  ]
33 }
34

```

The Above policy grants an IAM user access to the files in a folder named folder1 within the S3 bucket named : singles3bucketaccess.