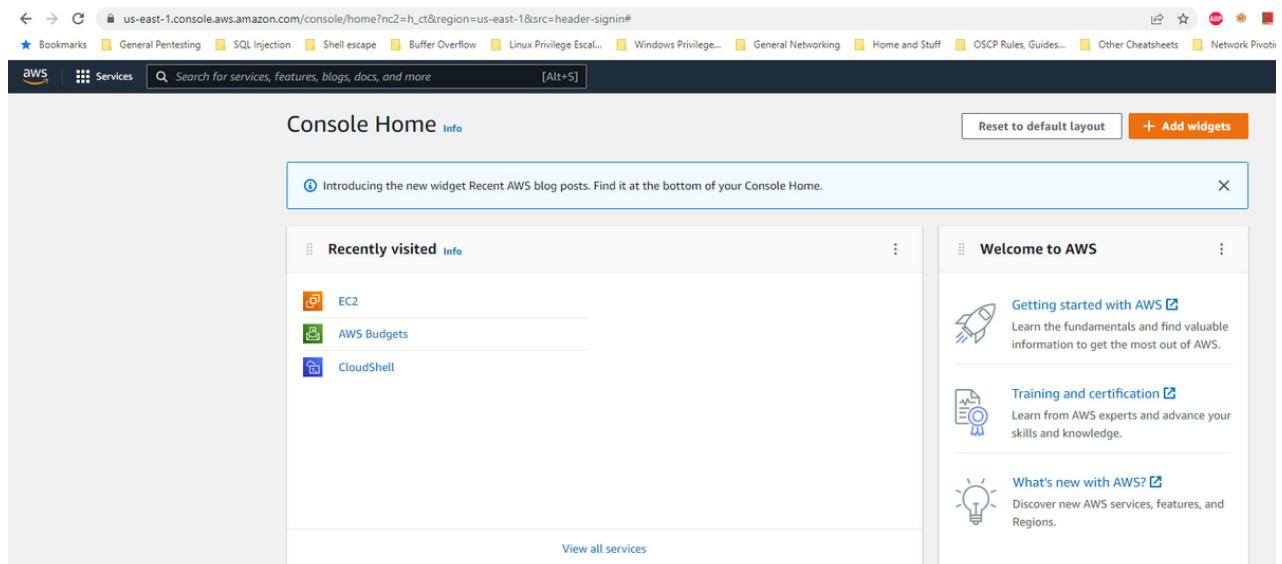


Cloud Walkthrough: How to make AWS IAM and Resource Policies.

The following walkthrough will help demonstrate how to properly setup AWS IAM and Resource policies regarding certain use cases for organizations. Below will be useful use cases regarding how to apply IAM policies regarding different IAM types and how to restrict access to an organizational S3 bucket. Furthermore, the use cases will also surround how to restrict access with Resource policies aimed at the specific S3 bucket as well.

Create Test Users, Roles, and Groups.

Step 1: Create test users "greg", "rick", "claire", and "brian".



Search results for 'iam'

Services (6)

- Features (17)
- Blogs (1,372)
- Documentation (115,700)
- Events (5)
- Marketplace (358)

Services

IAM Manage access to AWS resources

Top features

- Groups
- Users
- Roles
- Policies
- Access Analyzer

IAM Identity Center (successor to AWS Single Sign-On) Manage workforce user access to multiple AWS accounts and cloud applications

Resource Access Manager Share AWS resources with other accounts or AWS Organizations

Amazon VPC IP Address Manager Managed IP address management service

Features

Groups IAM feature

Roles IAM feature

See all 6 results ▾

Reset to default layout + Add widgets

Welcome to AWS

Getting started with AWS Learn the fundamentals and find valuable information to get the most out of AWS.

Training and certification Learn from AWS experts and advance your skills and knowledge.

What's new with AWS? Discover new AWS services, features, and Regions.

us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#users

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users**
- Roles
- Policies
- Identity providers
- Account settings

Access reports

- Access analyzer
- Archive rules
- Analyzers
- Settings
- Credential report
- Organization activity
- Service control policies (SCPs)

IAM > Users

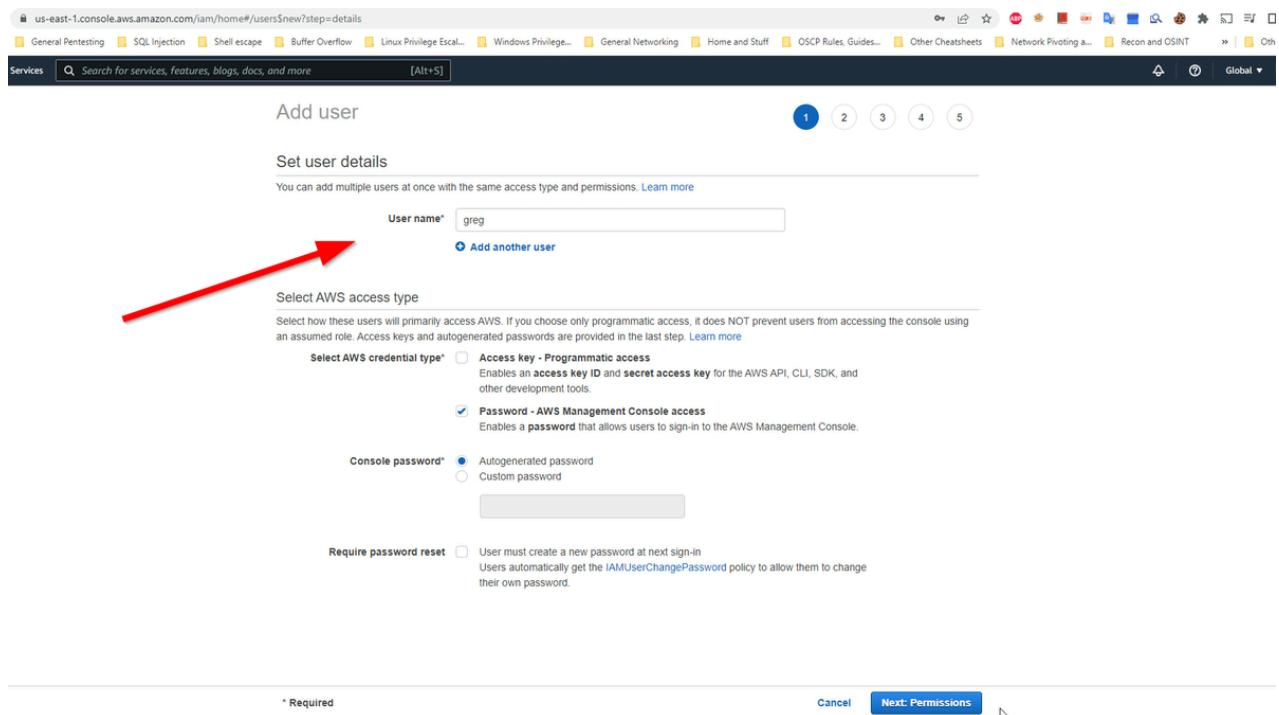
Users (0) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Find users by username or access key

User name	Groups	Last activity	MFA	Password age	Access key age
No resources to display					

Add users



Add user

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name* [Add another user](#)

Select AWS access type

Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Select AWS credential type* **Access key - Programmatic access** Enables an access key ID and secret access key for the AWS API, CLI, SDK, and other development tools.

Password - AWS Management Console access Enables a password that allows users to sign-in to the AWS Management Console.

Console password* Autogenerated password Custom password

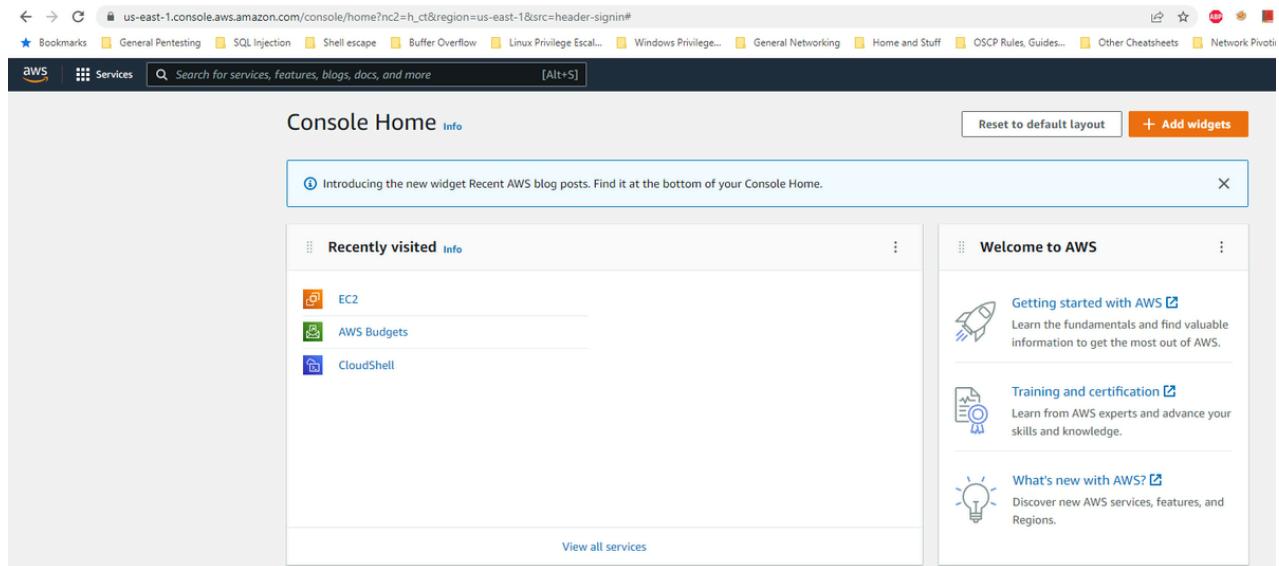
Require password reset User must create a new password at next sign-in
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

* Required

[Cancel](#) [Next: Permissions](#)

Please Note: Other IAM users were created but didn't want to include too many screenshots. IAM users were created with no permissions or tags which should be added later in the IAM walkthrough.

Step 2: Create test roles "Audit", "Test", "EC2TestRole", "DB"



Console Home [Info](#)

Introducing the new widget Recent AWS blog posts. Find it at the bottom of your Console Home.

Recently visited [Info](#)

-  EC
-  AWS Budgets
-  CloudShell

[View all services](#)

Welcome to AWS

Getting started with AWS [Get Started](#)
Learn the fundamentals and find valuable information to get the most out of AWS.

Training and certification [Get Certified](#)
Learn from AWS experts and advance your skills and knowledge.

What's new with AWS [Discover](#)
Discover new AWS services, features, and Regions.

The screenshot shows the AWS Home page with a search bar at the top containing 'iam'. Below the search bar, there is a sidebar with various service categories and a main content area titled 'Services'.

Services (6)

- Features (17)
- Blogs (1,372)
- Documentation (115,770)
- Events (5)
- Marketplace (358)

Services

IAM Manage access to AWS resources

Top features

- Groups
- Users
- Roles
- Policies
- Access Analyzer

IAM Identity Center (successor to AWS Single Sign-On) Manage workforce user access to multiple AWS accounts and cloud applications

Resource Access Manager Share AWS resources with other accounts or AWS Organizations

Amazon VPC IP Address Manager Managed IP address management service

Features

Groups

Roles

A red arrow points from the top-left towards the 'IAM' service card.

The screenshot shows the IAM Roles management page. The left sidebar includes 'Identity and Access Management (IAM)' and 'Access management' sections. The main content area displays a list of roles.

Roles (3) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
AWSServiceRoleForECS	AWS Service: ecs (Service-Linked Role)	
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	

Create role

Roles Anywhere Info

Authenticate your non AWS workloads and securely provide access to AWS services.

Access AWS from your non AWS workloads

Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS.

X.509 Standard

Use your own existing PKI infrastructure or use AWS Certificate Manager Private Certificate Authority to authenticate identities.

Temporary credentials

Use temporary credentials with ease and benefit from the enhanced security they provide.

A red arrow points from the bottom-right towards the 'Create role' button.

Screenshot of the AWS IAM Roles page. The left sidebar shows navigation options like Dashboard, Access management, and Access reports. The main content area displays a list of roles (4) with columns for Role name and Trusted entities. A red arrow points from the bottom-left towards the 'Roles anywhere' section at the bottom of the page.

Role name	Trusted entities
AWSServiceRoleForECS	AWS Service: ecs (Service-Linked Role)
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)
EC2TestRole	Account: 548429986066

Roles anywhere Info
Authorize your non AWS workloads and securely provide access to AWS services.

Access AWS from your non AWS workloads
Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS.

X.509 Standard
Use your own existing PKI infrastructure or use [AWS Certificate Manager Private Certificate Authority](#) to authenticate identities.

Step 3: Create test groups "Contractors", "MSP", "Admin", "Test"

Screenshot of the AWS Console Home page. The top navigation bar includes links for General Pentesting, SQL Injection, Shell escape, Buffer Overflow, Linux Privilege Escal., Windows Privilege Escal., General Networking, Home and Stuff, OSCP Rules, Guides..., Other Cheatsheets, and Network Pivot. The main content area features the 'Recently visited' section (listing EC2, AWS Budgets, CloudShell) and the 'Welcome to AWS' section (listing Getting started with AWS, Training and certification, and What's new with AWS?).

The screenshot shows the AWS Lambda console search results for the term 'iam'. The search bar at the top contains 'iam'. On the left, a sidebar lists various AWS services and features. A red arrow points from the sidebar towards the search results. The main area displays a list of services under 'Services' and 'Features' categories. The 'IAM' service is highlighted with a blue box and a star icon. Other listed services include IAM Identity Center, Resource Access Manager, and Amazon VPC IP Address Manager.

The screenshot shows the AWS IAM User Groups page. The URL is us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/groups. The left sidebar shows the IAM navigation menu, with 'User groups' selected. A red arrow points from the sidebar towards the main content area. The main content area shows a table titled 'User groups (0) Info'. The table has columns for 'Group name', 'Users', 'Permissions', and 'Creation date'. A single row is present with the message 'No resources to display'. At the top right of the table, there are buttons for 'Delete' and 'Create group'.

The screenshot shows the 'Create New User Group' page in the AWS IAM console. The left sidebar shows 'Identity and Access Management (IAM)' with 'User groups' selected. The main form has a title 'Name the group' and a 'User group name' field containing 'Admin'. Below it is a table titled 'Add users to the group - Optional' showing four users: brian, claire, greg, and rick, all selected. At the bottom is a section 'Attach permissions policies - Optional' with a table showing one policy named 'AdministratorAccess'.

Create Test S3 Buckets.

Navigate to the S3 Service page from the AWS Console.

The screenshot shows the AWS Services page with a search bar at the top. On the left, a sidebar lists 'Services (7)', 'Features (12)', 'Blogs (1,122)', 'Documentation (112,113)', 'Knowledge Articles (30)', 'Tutorials (7)', 'Events (14)', and 'Marketplace (615)'. The main area displays cards for various services: 'S3' (Scalable Storage in the Cloud), 'S3 Glacier' (Archive Storage in the Cloud), 'Athena' (Query Data in S3 using SQL), and 'AWS Snow Family' (Large Scale Data Transport). A red arrow points to the 'S3' card.

Choose the "Create bucket" option on the S3 homepage of the AWS Console.

The screenshot shows the AWS S3 console homepage. In the top right corner, there is a 'Create a bucket' button. Below it, there is a 'Pricing' section. A red arrow points from the 'Create a bucket' button towards the 'Create bucket' link in the 'General configuration' section of the 'Create bucket' wizard.

Create an AWS S3 Bucket with the name of "gfuen-testbucket" and "gfuen-testbucket2" for the IAM Policy use cases that will help further in the IAM walkthrough. Be sure to keep all of the safe default configuration options on when creating an S3 bucket especially the checkbox for allowing public access to the created S3 bucket.

The screenshot shows the 'Create bucket' wizard in the AWS S3 console. The first step, 'General configuration', is displayed. The 'Bucket name' field contains 'gfuen-testbucket'. The 'AWS Region' dropdown is set to 'US East (N. Virginia) us-east-1'. Below these fields is a section for 'Copy settings from existing bucket - optional', which includes a 'Choose bucket' button. The second step, 'Object Ownership', is also visible, showing two options: 'ACLs disabled (recommended)' (selected) and 'ACLs enabled'. The 'Object Ownership' field is set to 'Bucket owner enforced'.

Block Public Access settings for this bucket

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 this bucket and its 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 this bucket 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.

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

- Disable
- Enable

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

- Disable
- Enable

Tags (1) - optional

Track storage cost or other criteria by tagging your bucket. [Learn more](#)

Key Value - optional

name	test1	Remove
------	-------	------------------------

[Add tag](#)

Default encryption

Automatically encrypt new objects stored in this bucket. [Learn more](#)

Server-side encryption

- Disable
- Enable

► Advanced settings

The screenshot shows the AWS S3 service page. A green banner at the top indicates "Successfully created bucket 'gfuen-testbucket'". Below it, a blue bar asks if the user is missing ways to reduce storage costs and enhance data protection. The main area displays an "Account snapshot" and a table titled "Buckets (1) Info". The table shows one bucket named "gfuen-testbucket" located in "US East (N. Virginia) us-east-1". The bucket is described as "Bucket and objects not public" and was created on "September 15, 2022, 16:24:09 (UTC-05:00)". A red arrow points from the left margin towards the "gfuen-testbucket" row.

Name	AWS Region	Access	Creation date
gfuen-testbucket	US East (N. Virginia) us-east-1	Bucket and objects not public	September 15, 2022, 16:24:09 (UTC-05:00)

Put two files inside of the test S3 bucket for IAM Policies. The first file will be a "HelloWorld" file and the second file will be a "Secret" file containing credentials.

The screenshot shows the "Objects (2)" section of the "gfuen-testbucket" page. It lists two objects: "HelloWorld.txt" and "SecretFile.txt". Both files are of type "txt" and were uploaded on "September 15, 2022". A red box highlights the list of objects.

Name	Type	Last modified	Size	Storage class
HelloWorld.txt	txt	September 15, 2022, 16:46:04 (UTC-05:00)	12.0 B	Standard
SecretFile.txt	txt	September 15, 2022, 16:46:05 (UTC-05:00)	15.0 B	Standard

Contents of HelloWorld.txt:

```
Hello world!
```

Contents of SecretFile.txt:

```
admin:password1
```

Use Case 1: Allow READ, LIST, and WRITE access to S3 Bucket for IAM User using an

IAM policy.

Create an IAM Policy "Policy1.json" for user "brian" to access to all objects in S3 bucket "gfuen-testbucket".

Policy1.json

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "S3Access1",  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutObject",  
                "s3:GetObject",  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::gfuen-testbucket",  
                "arn:aws:s3:::gfuen-testbucket/*"  
            ]  
        }  
    ]  
}
```

The screenshot shows the AWS IAM 'Create policy' interface. At the top, there's a navigation bar with 'aws' and 'Services'. A search bar says 'Search for services, features, blogs, docs, and more [Alt+S]'. Below the search bar, there are three numbered tabs: '1' (selected), '2', and '3'. The main area is titled 'Create policy' and contains a help message: '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'. There are two tabs at the top of this area: 'Visual editor' (disabled) and 'JSON' (selected). To the right of the tabs is a link 'Import managed policy'. The JSON code area displays the following code:

```
1 - {  
2     "Version": "2012-10-17",  
3     "Statement": []  
4 }
```

The screenshot shows the AWS IAM Policies > Policy1 Summary page. The left sidebar lists various IAM management options like Dashboard, Access management, Policies, and Access reports. The main area displays the Policy ARN (arn:aws:iam::[REDACTED];policy/Policy1) and a Description field. Below these are tabs for Permissions, Policy usage, Tags, Policy versions, and Access Advisor. The Permissions tab is selected, showing the Policy summary and a JSON view. The JSON code is as follows:

```

1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
5              "Sid": "S3Access1",
6              "Effect": "Allow",
7              "Action": [
8                  "s3:PutObject",
9                  "s3:GetObject",
10                 "s3>ListBucket"
11             ],
12             "Resource": [
13                 "arn:aws:s3:::gfuen-testbucket",
14                 "arn:aws:s3:::gfuen-testbucket/*"
15             ]
16         }
17     ]
18 }

```

The following is where S3 actions can be found to construct a proper IAM Policy for S3 access.

S3 Actions Documentation: https://docs.aws.amazon.com/service-authorization/latest/reference/list_amazons3.html

The following is a great guide on what structure to follow for an IAM Policy.

IAM Policy Structure: <https://dev.to/tanmaygi/aws-identity-and-access-management-practical-guide-cheat-sheet-3528>

Attack Policy1 to AWS IAM user "brian".

Add permissions to brian

Grant permissions

Use IAM policies to grant permissions. You can assign an existing policy or create a new one.

Create policy

Filter policies Policy name ▾

	Type	Used as
AdministratorAccess	Job function	None
AdministratorAccess-Amplify	AWS managed	None
AdministratorAccess-AWSElasticBeanstalk	AWS managed	None
AlexaForBusinessDeviceSetup	AWS managed	None
AlexaForBusinessFullAccess	AWS managed	None
AlexaForBusinessGatewayExecution	AWS managed	None
AlexaForBusinessLifesizeDelegatedAccessPolicy	AWS managed	None
AlexaForBusinessPolyDelegatedAccessPolicy	AWS managed	None
AlexaForBusinessReadOnlyAccess	AWS managed	None
AmazonAPIGatewayAdministrator	AWS managed	None
AmazonAPIGatewayInvokeFullAccess	AWS managed	None
AmazonAPIGatewayPushToCloudWatchLogs	AWS managed	None
AmazonAppFlowFullAccess	AWS managed	None

Cancel **Next: Review**

Identity and Access Management (IAM)

Dashboard

Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

Access reports

Access analyzer

Archive rules

Analyzers

Settings

Credential report

Organization activity

Service control policies (SCPs)

Search IAM

AWS account ID: [REDACTED]

Users > brian

Summary

User ARN: arn:aws:iam:[REDACTED];user/brian [Edit](#)

Path: /

Creation time: 2022-09-11 14:39 CDT

Permissions **Groups** **Tags** **Security credentials** **Access Advisor**

▼ Permissions policies (1 policy applied)

Add permissions

Policy name ▾

Attached directly

- ▶ Policy1

▶ Permissions boundary (not set)

▼ Generate policy based on CloudTrail events

You can generate a new policy based on the access activity for this user, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a least privileged policy that you can attach to this user. Share your [feedback](#) and help us improve the policy generation experience.

Generate policy

No requests to generate a policy in the past 7 days.

Given this, AWS user "brian" will need to download the Access keys in order to make calls with the AWS cli.

User ARN: arn:aws:iam::[REDACTED]:user/brian
Path: /
Creation time: 2022-09-11 14:39 CDT

Access key ID	Created	Last used
AKIA[REDACTED]	2022-09-16 19:35 CDT	2022-09-17 13:04 CDT with s3 in us-east-1

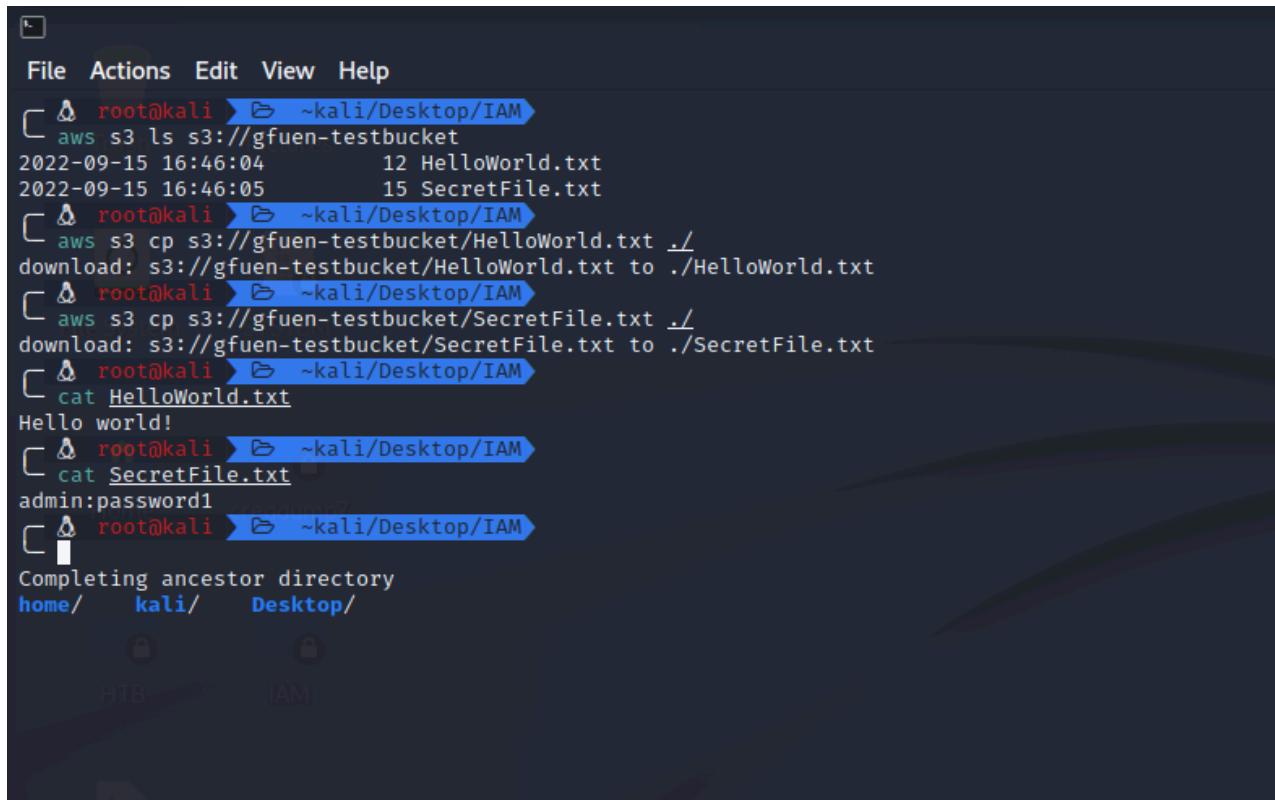
Likewise, now that user "brian" has an attached IAM policy for S3 access "brian" can login into the AWS CLI using his downloaded Access keys using the `aws configure` command.

```
root@kali:~# aws configure
AWS Access Key ID [None]: AKIA[REDACTED]
AWS Secret Access Key [None]: c0VFN/[REDACTED]
Default region name [None]:
Default output format [None]:
root@kali:~# aws s3 ls s3://gfuen-testbucket
2022-09-15 16:46:04      12 HelloWorld.txt
2022-09-15 16:46:05      15 SecretFile.txt
root@kali:~#
```

Notice that the first `s3 ls` command I tried didn't work when logged into the AWS cli as user "brian". To put it another way, due to the fact that the IAM policy "Policy1" attached to user "brian" did not have a "s3>ListAllMyBuckets" permission I was unable to see the "gfuen-testbucket" unless specified using the AWS cli.

AWS s3>ListAllMyBuckets:

[https://docs.aws.amazon.com/cli/latest/reference/s3api/list-buckets.html ↗](https://docs.aws.amazon.com/cli/latest/reference/s3api/list-buckets.html)



The screenshot shows a terminal window with a dark background and light-colored text. At the top, there's a menu bar with 'File', 'Actions', 'Edit', 'View', and 'Help'. Below the menu, the terminal prompt is 'root@kali ~ kali/Desktop/IAM'. The user runs several commands:

- 'aws s3 ls s3://gfuuen-testbucket' lists two files: 'HelloWorld.txt' (size 12) and 'SecretFile.txt' (size 15).
- 'aws s3 cp s3://gfuuen-testbucket>HelloWorld.txt ./' downloads 'HelloWorld.txt' to the current directory.
- 'aws s3 cp s3://gfuuen-testbucket>SecretFile.txt ./' downloads 'SecretFile.txt' to the current directory.
- 'cat HelloWorld.txt' displays the contents of the file, which is 'Hello world!'.
- 'cat SecretFile.txt' displays the contents of the file, which is 'admin:password1'.

At the bottom of the terminal, there's a message: 'Completing ancestor directory' followed by a list of paths: 'home/ kali/ Desktop/'. The desktop environment visible behind the terminal window includes icons for HTB and IAM.

Moreover, in the end I am able to download and read all the files as AWS user "brian" from S3 "gfuuen-testbucket".

Use Case 2: Allow READ, LIST, and WRITE access to S3 Bucket for IAM Role for an S3 Resource Policy utilizing SourceIP.

Create an S3 Resource Policy for role "EC2TestRole" to access to all objects in S3 bucket "gfuuen-testbucket" based on SourceIP for more granular secure access. Navigate to the specified S3 bucket and under the permissions tab one can edit the S3 Resource policy for the selected S3 bucket to restrict access.

Bucket Policy

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "S3BucketAccess1",  
            "Effect": "Allow",  
            "Principal": {"  
                "AWS": "arn:aws:iam::<aws_account_id>:role/EC2TestRole"  
            },  
            "Action": [  
                "s3:PutObject",  
                "s3:GetObject"  
            ],  
            "Resource": "arn:aws:s3:::gfuen-testbucket/*",  
            "Condition": {  
                "NotIpAddress": {"  
                    "aws:SourceIp": "<SourceIP>/24"  
                }  
            }  
        },  
        {  
            "Sid": "S3ListBucketAccess1",  
            "Effect": "Allow",  
            "Principal": {"  
                "AWS": "arn:aws:iam::<aws_account_id>:role/EC2TestRole"  
            },  
            "Action": "s3>ListBucket",  
            "Resource": "arn:aws:s3:::gfuen-testbucket",  
            "Condition": {  
                "NotIpAddress": {"  
                    "aws:SourceIp": "<SourceIP>/24"  
                }  
            }  
        }  
    ]  
}
```

Permissions overview

Block public access (bucket settings)

Block all public access On

Bucket policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "S3BucketAccess1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::[REDACTED]:role/EC2TestRole"
      },
      "Action": [
        "s3:PutObject",
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:::gfuen-testbucket/*",
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": "[REDACTED]/24"
        }
      }
    },
    {
      "Sid": "S3ListBucketAccess1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::[REDACTED]:role/EC2TestRole"
      },
      "Action": "s3>ListBucket",
      "Resource": "arn:aws:s3:::gfuen-testbucket",
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": "[REDACTED]/24"
        }
      }
    }
  ]
}
```

Bucket policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "S3BucketAccess1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::[REDACTED]:role/EC2TestRole"
      },
      "Action": [
        "s3:PutObject",
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:::gfuen-testbucket/*",
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": "[REDACTED]/24"
        }
      }
    },
    {
      "Sid": "S3ListBucketAccess1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::[REDACTED]:role/EC2TestRole"
      },
      "Action": "s3>ListBucket",
      "Resource": "arn:aws:s3:::gfuen-testbucket",
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": "[REDACTED]/24"
        }
      }
    }
  ]
}
```

Copy

Create an IAM Policy "Policy3.json" for user "greg" in order to allow role "EC2TestRole" to be assumed by user "greg" within the AWS account.

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.

Visual editor **JSON**

```

1 [
2   "Version": "2012-10-17",
3   "Statement": []
4 ]

```

Policies > Policy3

Summary

Policy ARN: arn:aws:iam::██████████:policy/Policy3

Description:

Permissions **Policy usage** **Tags** **Policy versions** **Access Advisor**

Policy summary **{ } JSON** **Edit policy**

```

1 [
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "sts:AssumeRole",
8         "iam>ListRoles"
9       ],
10      "Resource": "arn:aws:iam::██████████:role/EC2TestRole"
11    }
12  ]

```

Policy3.json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole",
        "iam>ListRoles"
      ],
      "Resource": "arn:aws:iam::<aws_account_id>:role/EC2TestRole"
    }
  ]
}
```

Attach the newly created IAM policy "Policy3" to the AWS user "greg".

The screenshot shows the AWS IAM 'Add permissions to greg' wizard. The top navigation bar includes the AWS logo, 'Services' dropdown, a search bar ('Search for services, features, blogs, docs, and more'), and a keyboard shortcut ('[Alt+S]').

The main section is titled 'Add permissions to greg' and 'Grant permissions'. It says 'Use IAM policies to grant permissions. You can assign an existing policy or create a new one.' Below this are three buttons: 'Add user to group', 'Copy permissions from existing user', and 'Attach existing policies directly' (which is highlighted). A 'Create policy' button is also present.

A 'Filter policies' dropdown and a 'Search' input field are available. The main list displays a large number of AWS policies, each with a checkbox, a small icon, the policy name, and its type (e.g., Job, AWS). Some policies listed include 'AdministratorAccess', 'AdministratorAccess-Amplify', 'AdministratorAccess-AWSElasticBeanstalk', 'AlexaForBusinessDeviceSetup', 'AlexaForBusinessFullAccess', 'AlexaForBusinessGatewayExecution', 'AlexaForBusinessLifesizeDelegatedAccessPolicy', 'AlexaForBusinessPolyDelegatedAccessPolicy', 'AlexaForBusinessReadOnlyAccess', 'AmazonAPIGatewayAdministrator', 'AmazonAPIGatewayInvokeFullAccess', 'AmazonAPIGatewayPushToCloudWatchLogs', and 'AmazonAppFlowFullAccess'.

At the bottom right are 'Cancel' and 'Next: Review' buttons.

New feature to generate a policy based on CloudTrail events.
AWS uses your CloudTrail events to identify the services and actions used and generate a least privileged policy that you can attach to this user.

Users > greg

Summary

User ARN: arn:aws:iam::█████████████████████:user/greg

Path: /

Creation time: 2022-09-11 14:35 CDT

Permissions Groups (1) Tags (1) Security credentials Access Advisor

▼ Permissions policies (1 policy applied)

Add permissions

Policy name
Attached directly
▶ Policy3

▶ Permissions boundary (not set)

▼ Generate policy based on CloudTrail events

You can generate a new policy based on the access activity for this user, then customize, create, and attach it to this role. AWS uses your CloudTrail events to Share your [feedback](#) and help us improve the policy generation experience.

Generate policy

No requests to generate a policy in the past 7 days.

Furthermore, attach the following "Policy1.json" to AWS role "EC2TestRole" in order to enable S3 access to bucket "gfuen-testbucket".

Introducing the new IAM roles experience
We've redesigned the IAM roles experience to make it easier to use. Let us know what you think.

Policy was successfully attached to role.

IAM > Roles > EC2TestRole

EC2TestRole

Allows EC2 Instances to call AWS services on your behalf.

Summary

Creation date September 18, 2022, 09:08 (UTC-05:00)	ARN arn:aws:iam::█████████████████████:role/EC2TestRole	Link to switch roles in console https://signin.aws.amazon.com/switchrole?roleName=EC2TestRole&account=█████████████████████
Last activity Yesterday	Maximum session duration 1 hour	Instance profile ARN arn:aws:iam::█████████████████████:Role

Permissions Trust relationships Tags Access Advisor Revoke sessions

Permissions policies (1)
You can attach up to 10 managed policies.

Filter policies by property or policy name and press enter.

Policy name	Type	Description
Policy1	Customer managed	

Policy1.json

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "S3Access1",  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutObject",  
                "s3:GetObject",  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::gfuen-testbucket",  
                "arn:aws:s3:::gfuen-testbucket/*"  
            ]  
        }  
    ]  
}
```

Likewise, now that the AWS role "EC2TestRole" has proper permissions access can now be restricted from AWS principals to the S3 bucket "gfuen-testbucket". Given this, AWS user "greg" can login into the AWS console using his downloaded access keys using the `aws configure` command, and use the `aws sts assume-role` command in order to gain the aws role "EC2TestRole" identity in the AWS account. Then, the AWS role "EC2TestRole" can test READ, LIST, and WRITE access to the S3 bucket "gfuen-testbucket".

AWS Assume-Role Documentation:

[https://docs.aws.amazon.com/cli/latest/reference/sts/assume-role.html ↗](https://docs.aws.amazon.com/cli/latest/reference/sts/assume-role.html)

List the roles in the AWS account.

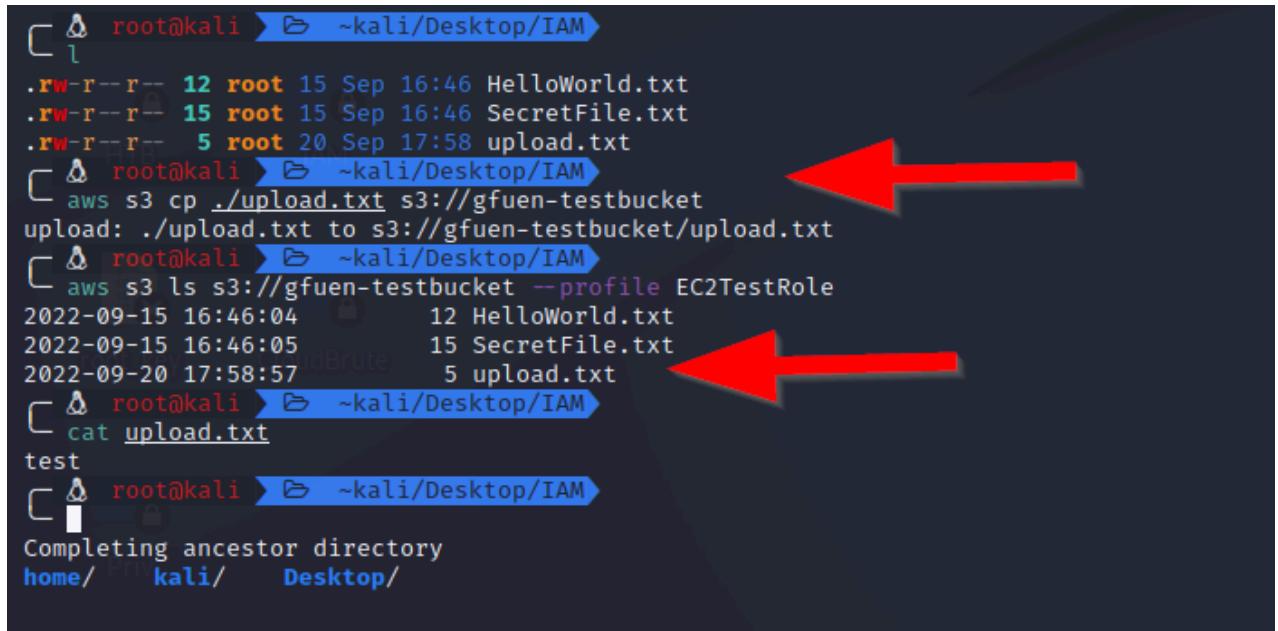
```
[root@kali ~]# aws configure --profile greg
AWS Access Key ID [None]: AKIAX7MHZPEJLV3A2DEV
AWS Secret Access Key [None]: bzMtAIZWr7TVyLmi6nqDjioEmVeSPoJmjo+vAwF
Default region name [None]:
Default output format [None]:
[root@kali ~]# aws iam list-roles --profile greg
{
    "Roles": [
        {
            "Path": "/aws-service-role/ecs.amazonaws.com/",
            "RoleName": "AWSServiceRoleForECS",
            "RoleId": "AROAAX7MHZPEJG6CRWJXYS",
            "Arn": "arn:aws:iam::54842986066:role/aws-service-role/ecs.amazonaws.com/AWSServiceRoleForECS",
            "CreateDate": "2022-02-04T02:36:15Z",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "ecs.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "Description": "Role to enable Amazon ECS to manage your cluster.",
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/aws-service-role/support.amazonaws.com/",
            "RoleName": "AWSServiceRoleForSupport",
            "RoleId": "AROAAX7MHZPEJIPNOCWDTK",
            "Arn": "arn:aws:iam::54842986066:role/aws-service-role/support.amazonaws.com/AWSServiceRoleForSupport",
            "CreateDate": "2022-01-19T00:31:18Z",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "support.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            }
        }
    ]
}
```

Assume the AWS role "EC2TestRole" using the AWS cli.

List items in S3 bucket. Download and read files from "qfuen-testbucket".

```
[root@kali:~kali/Desktop/CloudBrute]# aws sts get-caller-identity --profile EC2TestRole
{
    "UserId": "AROAX...:test-usecase2",
    "Account": "...",
    "Arn": "arn:aws:sts::...:assumed-role/EC2TestRole/test-usecase2"
}
[root@kali:~kali/Desktop/CloudBrute]# aws s3 ls s3://gfuen-testbucket --profile EC2TestRole
2022-09-15 16:46:04          12 HelloWorld.txt
2022-09-15 16:46:05          15 SecretFile.txt
[root@kali:~kali/Desktop/CloudBrute]# aws s3 cp s3://gfuen-testbucket/Secretfile.txt ./_
download: s3://gfuen-testbucket/Secretfile.txt to ./SecretFile.txt
[root@kali:~kali/Desktop/CloudBrute]# aws s3 cp s3://gfuen-testbucket>HelloWorld.txt ./_
download: s3://gfuen-testbucket>HelloWorld.txt to ./HelloWorld.txt
[root@kali:~kali/Desktop/CloudBrute]# cat SecretFile.txt
admin:password1
[root@kali:~kali/Desktop/CloudBrute]# cat HelloWorld.txt
Hello world!
[root@kali:~kali/Desktop/CloudBrute]#
```

Moreover, in the end I am able to download and read all the files as role "EC2TestRole" from S3 "gfuen-testbucket".



The screenshot shows a terminal session on a Kali Linux system. The user is root and is navigating through a directory structure under ~kali/Desktop/IAM. They list files (HelloWorld.txt, SecretFile.txt, upload.txt) and upload a new file named upload.txt to an S3 bucket named gfuen-testbucket. They then list the contents of the S3 bucket, which includes the uploaded file. Finally, they read the content of the upload.txt file.

```
root@kali:~/.kali/Desktop/IAM# ls
.rw-r--r-- 12 root 15 Sep 16:46 HelloWorld.txt
.rw-r--r-- 15 root 15 Sep 16:46 SecretFile.txt
.rw-r--r-- 5 root 20 Sep 17:58 upload.txt
root@kali:~/.kali/Desktop/IAM# aws s3 cp ./upload.txt s3://gfuen-testbucket
upload: ./upload.txt to s3://gfuen-testbucket/upload.txt
root@kali:~/.kali/Desktop/IAM# aws s3 ls s3://gfuen-testbucket --profile EC2TestRole
2022-09-15 16:46:04          12 HelloWorld.txt
2022-09-15 16:46:05          15 SecretFile.txt
2022-09-20 17:58:57          5 upload.txt
root@kali:~/.kali/Desktop/IAM# cat upload.txt
test
root@kali:~/.kali/Desktop/IAM#
Completing ancestor directory
home/  kali/  Desktop/
```

Finally, a test upload.txt file is able to be uploaded into the S3 bucket "gfuen-testbucket" with the permissions surrounding AWS role "EC2TestRole".

Use Case 3: Allow READ, LIST, and WRITE access to S3 Bucket for IAM Group based upon Resource tag.

Add the resource tag pair "env" "Test" to the S3 bucket "gfuen-testbucket2" in order to simulate an S3 bucket in an enterprise that should only be available to those on the Testing team.

The screenshot shows the AWS S3 Bucket Properties page for 'gfuen-testbucket2'. The 'Properties' tab is selected. The bucket name 'gfuen-testbucket2' is highlighted with a red box. In the 'Tags' section, the 'Env' tag is also highlighted with a red box. Other visible details include the AWS Region (US East (N. Virginia) us-east-1), ARN (arn:aws:s3:::gfuen-testbucket2), and Creation date (October 12, 2022, 17:06:21 (UTC-05:00)).

Create an S3 IAM Policy that can be attached to IAM User "greg" to access to all objects in S3 bucket "gfuen-testbucket2" based on a Resource tag for more granular secure access.

Policy4.json

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutObject",  
                "s3:GetObject"  
            ],  
            "Resource": "arn:aws:s3:::gfuen-testbucket2/*",  
            "Condition": {  
                "StringEquals": {"aws:ResourceTag/env": "${aws:PrincipalTag}"},  
            },  
        },  
        {  
            "Effect": "Allow",  
            "Action": "s3>ListBucket",  
            "Resource": "arn:aws:s3:::gfuen-testbucket2",  
            "Condition": {  
                "StringEquals": {"aws:ResourceTag/project": "${aws:PrincipalTag}"},  
            },  
        }  
    ]  
}
```

The screenshot shows the AWS IAM 'Create policy' interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, a search bar ('Search for services, features, blogs, docs, and more'), and a keyboard shortcut '[Alt+S]'. Below the navigation is a title 'Create policy'. A descriptive text explains that a policy defines AWS permissions for users, groups, or roles. It mentions that policies can be created and edited using the visual editor or JSON. The visual editor tab is selected, but the JSON tab is also visible. The JSON code area displays the following policy definition:

```
1 * {  
2     "Version": "2012-10-17",  
3     "Statement": []  
4 }
```

Policies > Policy4

Summary

Policy ARN arn:aws:iam::█████████████████████:policy/Policy4 

Description

Permissions Policy usage Tags Policy versions Access Advisor

Policy summary  Edit policy

```

1 - {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "s3:PutObject",
8         "s3:GetObject"
9       ],
10      "Resource": "arn:aws:s3:::gfuen-testbucket2/*",
11      "Condition": {
12        "StringEquals": {
13          "aws:ResourceTag/env": "${aws:PrincipalTag/env}"
14        }
15      }
16    },
17    {
18      "Effect": "Allow",
19      "Action": "s3>ListBucket",
20      "Resource": "arn:aws:s3:::gfuen-testbucket2",
21      "Condition": {
22        "StringEquals": {
23          "aws:ResourceTag/project": "${aws:PrincipalTag/project}"
24        }
25      }
26    }
27  ]
28 }

```

Create an S3 Resource Bucket Policy for IAM User "greg" to access to all objects in S3 bucket "gfuen-testbucket2". Navigate to the specified S3 bucket and under the permissions tab one can edit the S3 Resource policy for the selected S3 bucket to restrict access.

Bucket Policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "S3BucketAccess6",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::548429986066:user/greg"
      },
      "Action": [
        "s3:PutObject",
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:::gfuen-testbucket2/*"
    },
    {
      "Sid": "S3ListBucketAccess6",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::548429986066:user/greg"
      },
      "Action": "s3>ListBucket",
      "Resource": "arn:aws:s3:::gfuen-testbucket2"
    }
  ]
}
```

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](#)

Public access is blocked because Block Public Access settings are turned on for this bucket

To determine which settings are turned on, check your Block Public Access settings for this bucket. Learn more about [using Amazon S3 Block Public Access](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "S3BucketAccess6",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::548429986066:user/greg"
      },
      "Action": [
        "s3:PutObject",
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:::gfuen-testbucket2/*"
    },
    {
      "Sid": "S3ListBucketAccess6",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::548429986066:user/greg"
      },
      "Action": "s3>ListBucket",
      "Resource": "arn:aws:s3:::gfuen-testbucket2"
    }
  ]
}
```

[Edit](#) [Delete](#)

[Copy](#)

```
[root@kali ~]# aws configure --profile greg
AWS Access Key ID [*****2DEV]: *****
AWS Secret Access Key [*****vAwF]: *****
Default region name [None]:
Default output format [None]:
[root@kali ~]# 
[!] Completing ancestor directory
home/   kali/   Desktop/
```