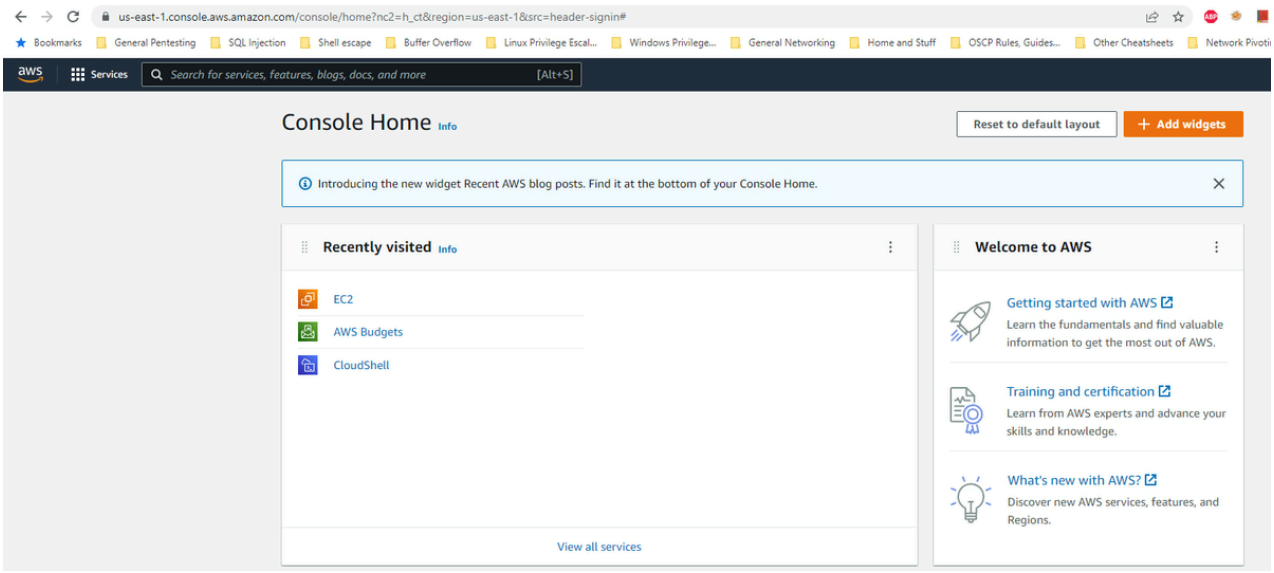
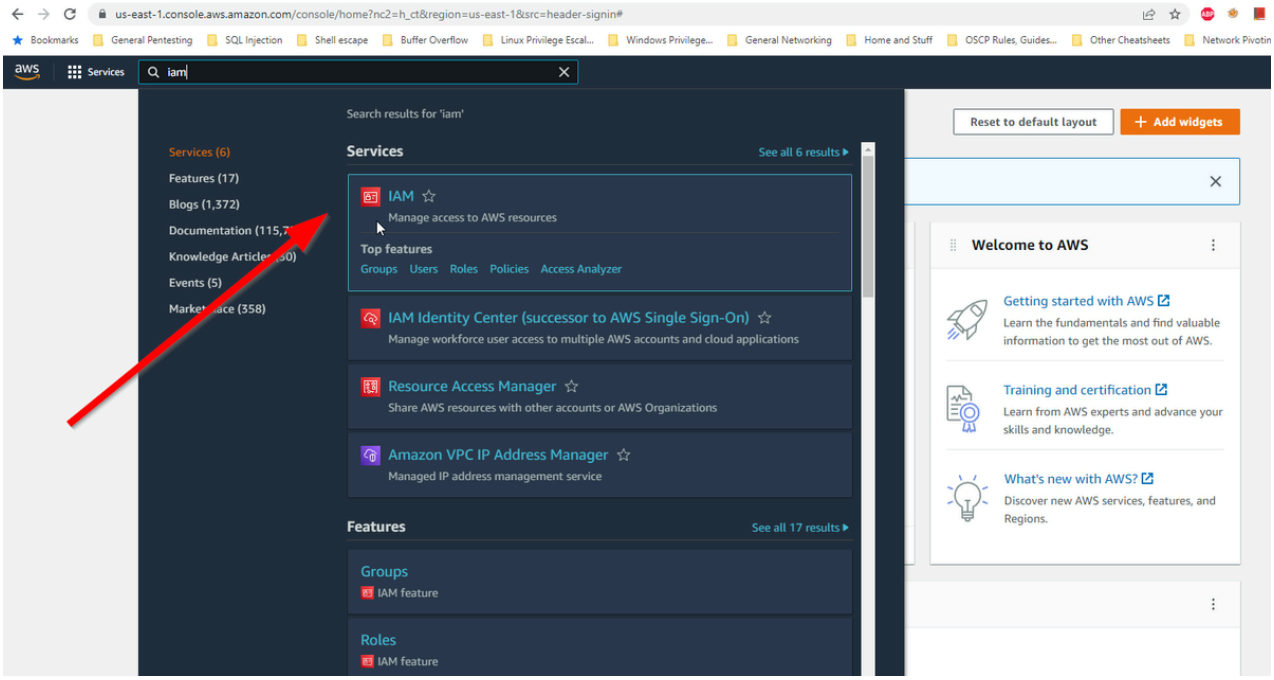
Cloud Walkthrough: How to make AWS IAM and Resource Policies.

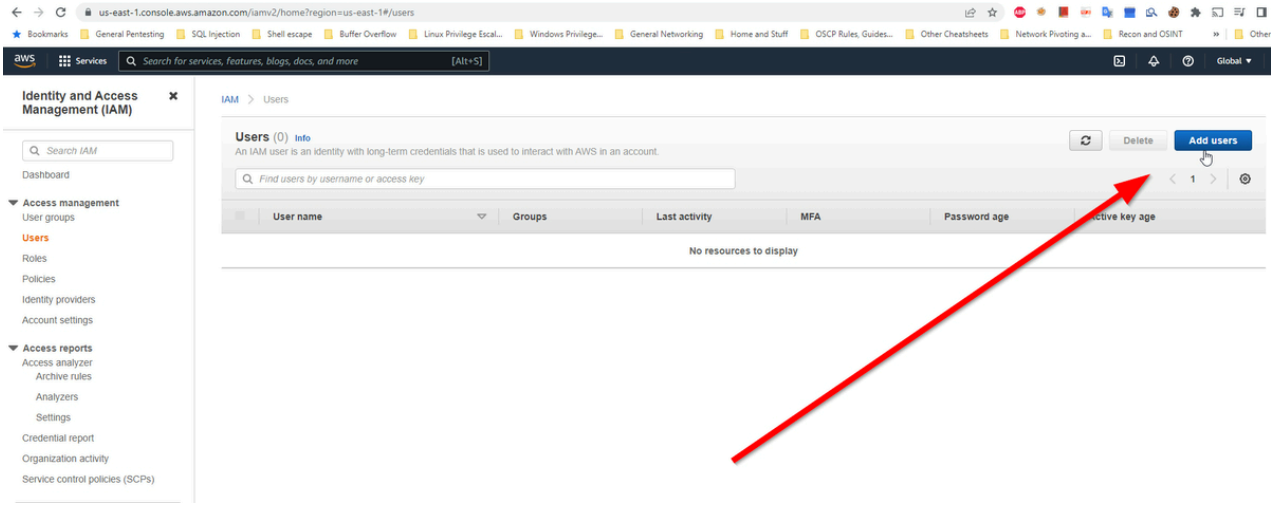
The following walkthrough will be help demonstrate how to properly setup AWS IAM and Resource policies regarding certain use cases for organizations. Below will be useful use cases regarding how to apply IAM policies regarding different IAM types and how to restrict access to an organizational S3 bucket.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

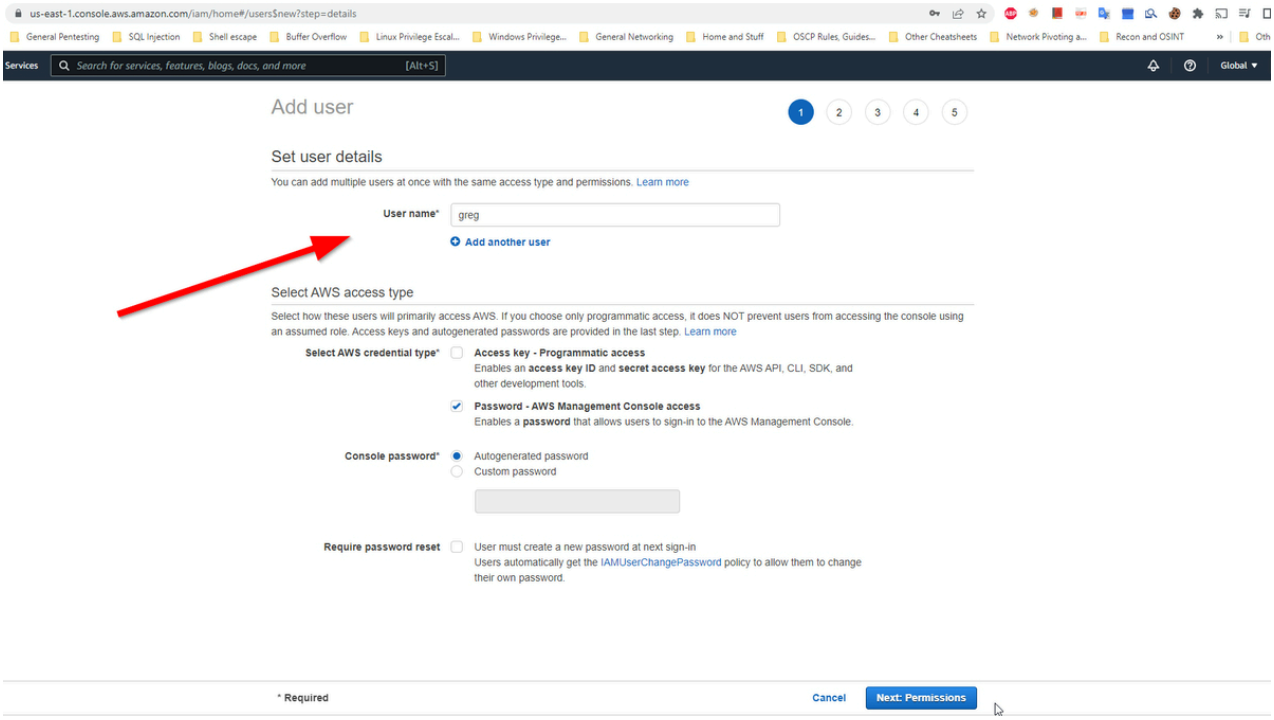
Create Test Users, Roles, and Groups.

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



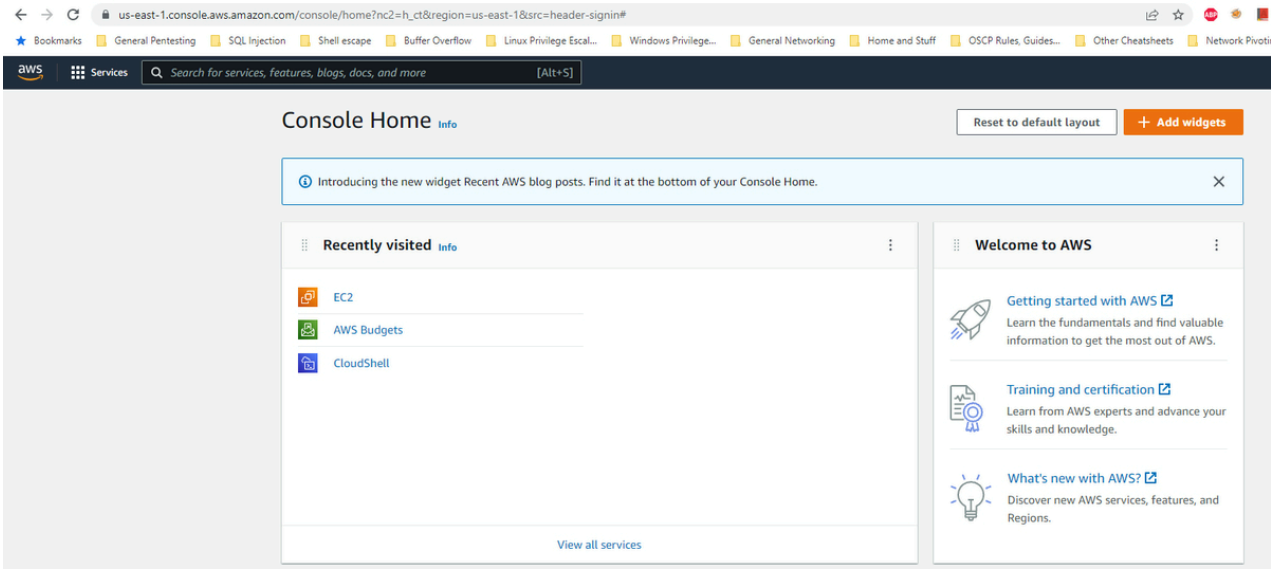

us-east-1.console.aws.amazon.com/console/home?nc2=h_ct&region=us-east-1&src=header-signin#
aws
Services
iam
Search results for 'iam'
Services (6)
Features (17)
Blogs (1,372)
Services
See all 6 results
IAM
Manage access to AWS resources
Top features
Groups Users Roles Policies Access Analyzer
IAM Identity 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
See all 17 results
Groups
IAM feature
Roles
IAM feature
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.

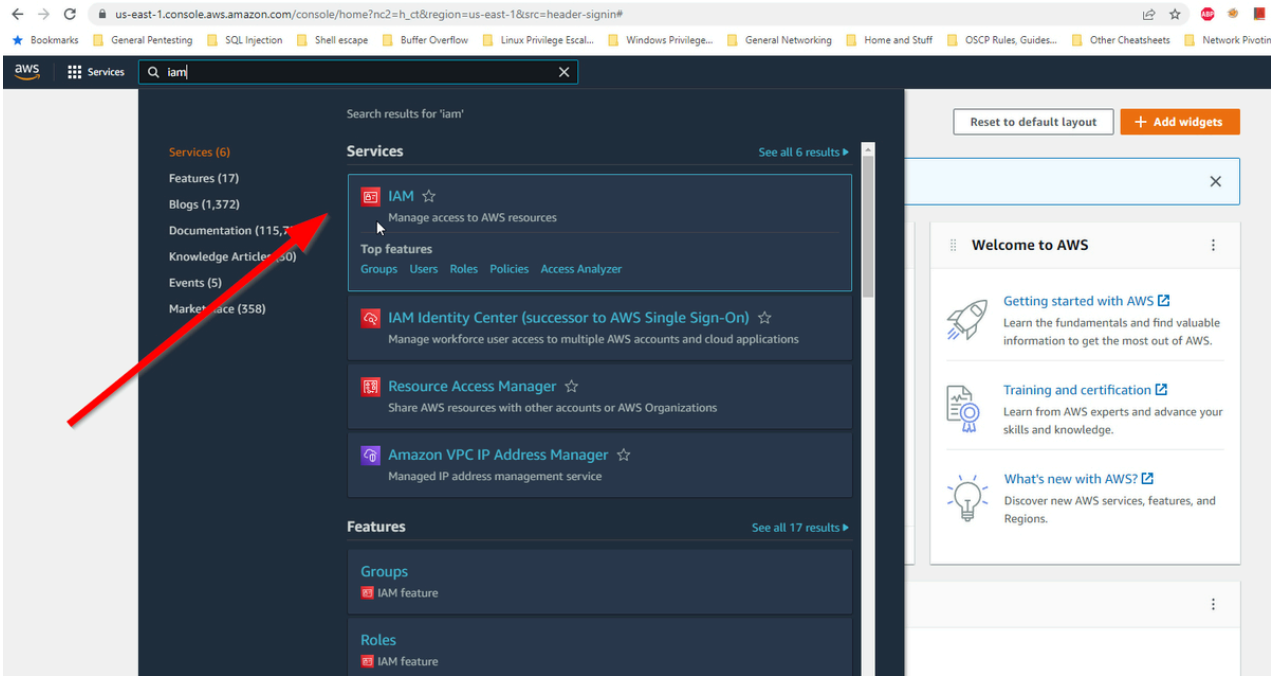

us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/users
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers
Settings
Credential report
Organization 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
Delete
Add users
User name
Groups
Last activity
MFA
Password age
Active key age
No resources to display

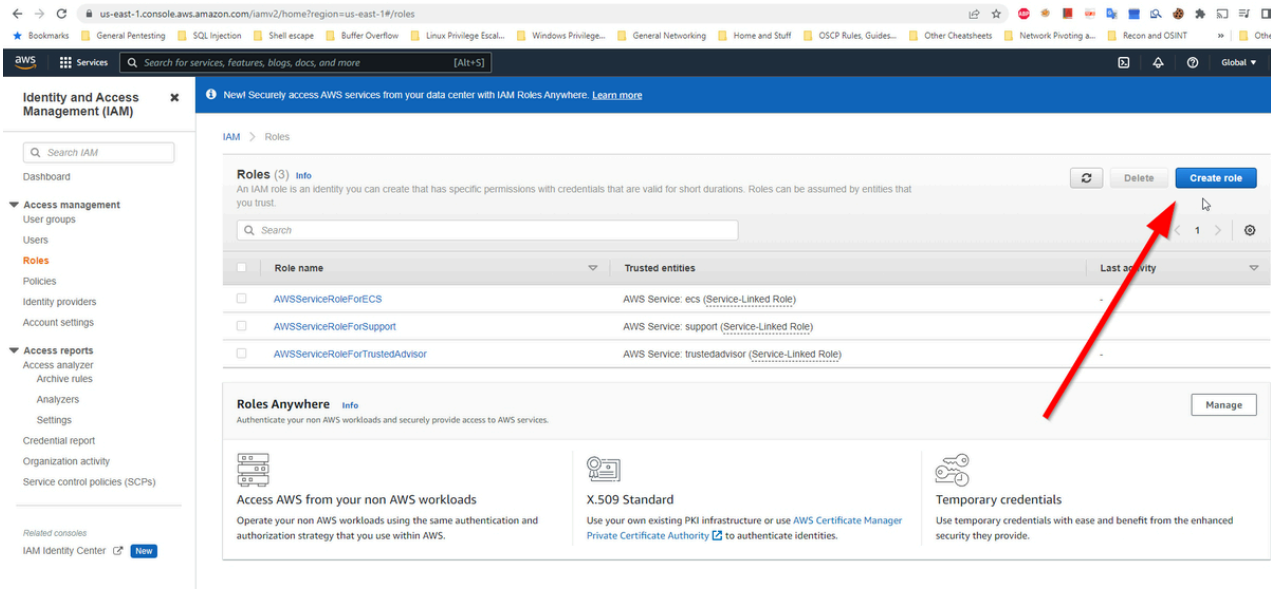


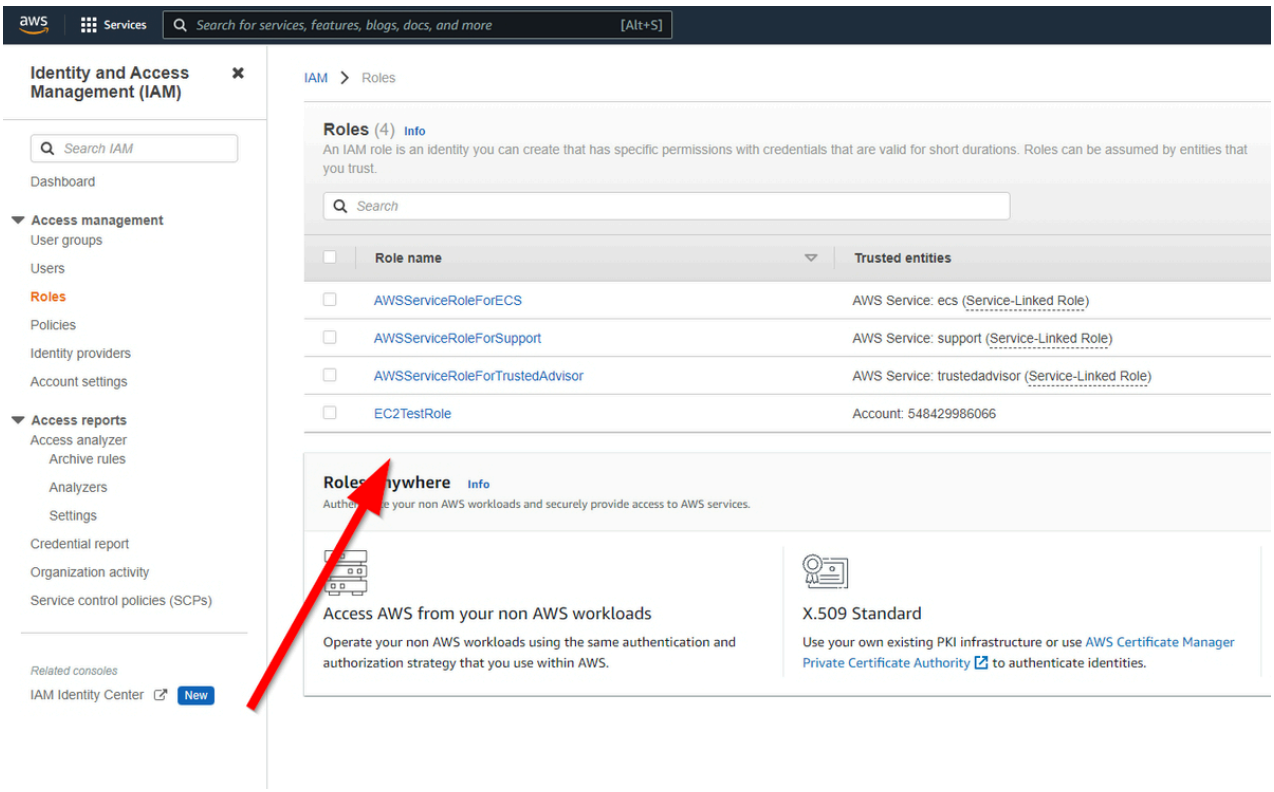
Please Note: Other IAM users were created but didnt want to include too many screenshots. IAM users were created with no permissions or tags which should be added later in the IAM walkthrough.

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

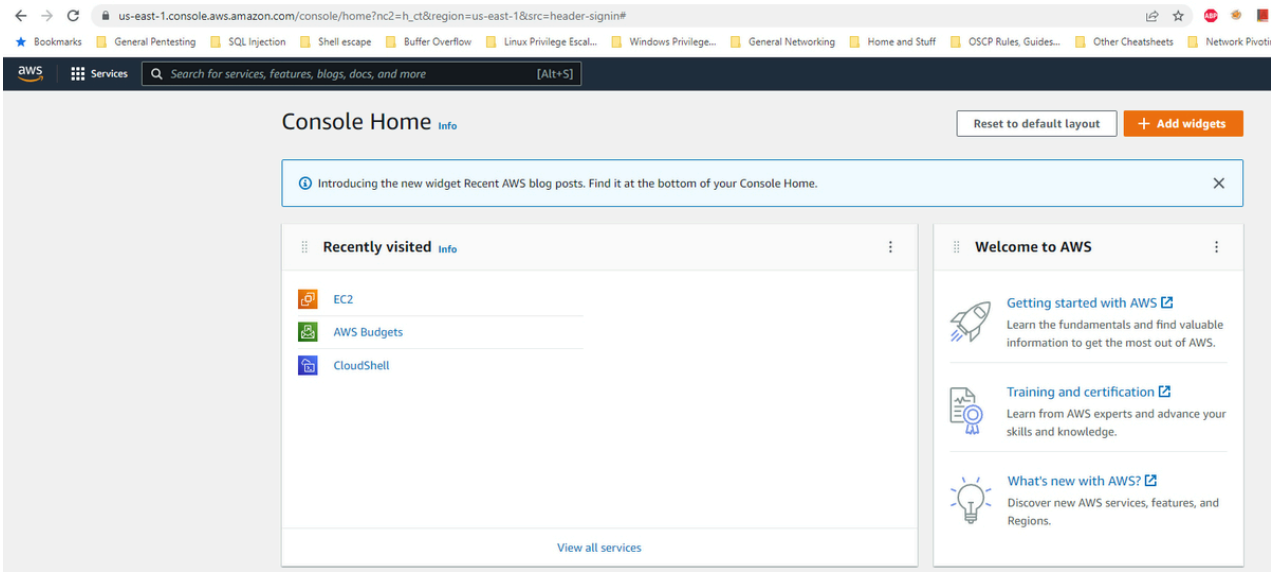


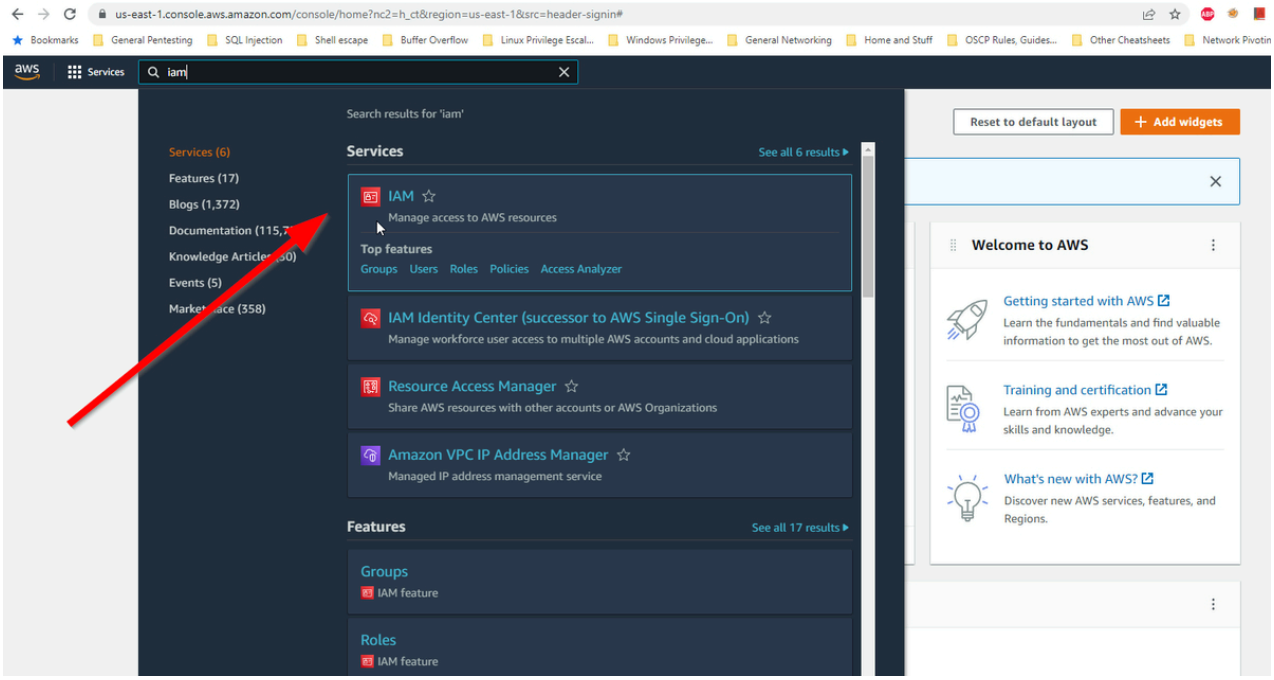

us-east-1.console.aws.amazon.com/console/home?nc2=h_ct&region=us-east-1&src=header-signin#
aws
Services
iam
Search results for 'iam'
Services (6)
Features (17)
Blogs (1,372)
Services
See all 6 results
IAM
Manage access to AWS resources
Top features
Groups Users Roles Policies Access Analyzer
IAM Identity 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
See all 17 results
Groups
IAM feature
Roles
IAM feature
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.

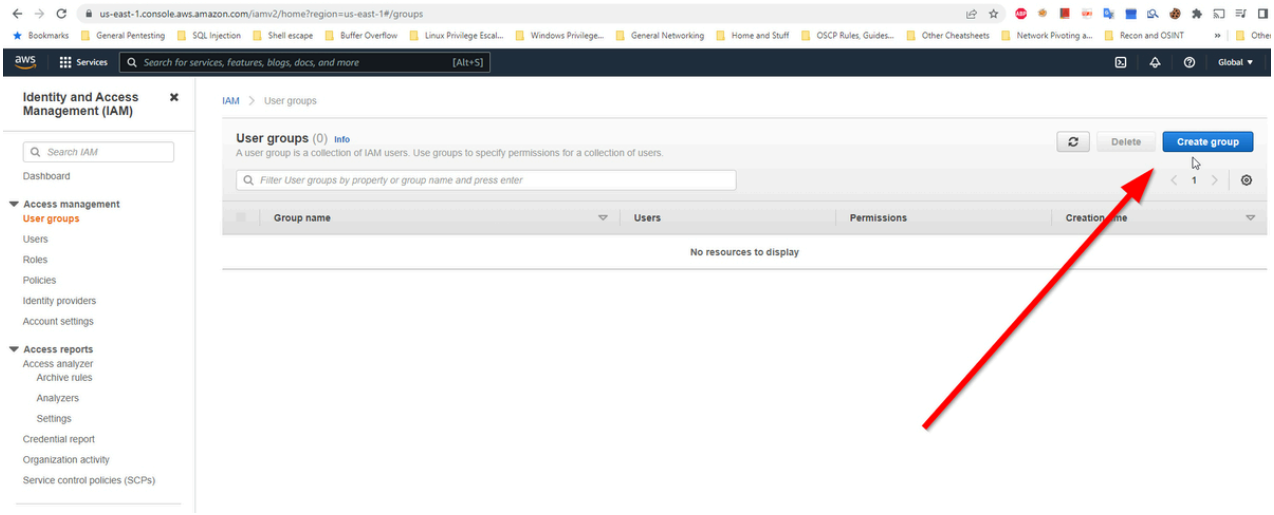

us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/roles
Identity and Access Management (IAM)
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers
Settings
Credential report
Organization activity
Service control policies (SCPs)
New! Securely access AWS services from your data center with IAM Roles Anywhere. Learn more
IAM > 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.
Delete
Create role
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)
Roles Anywhere Info
Authenticate your non AWS workloads and securely provide access to AWS services.
Manage
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.

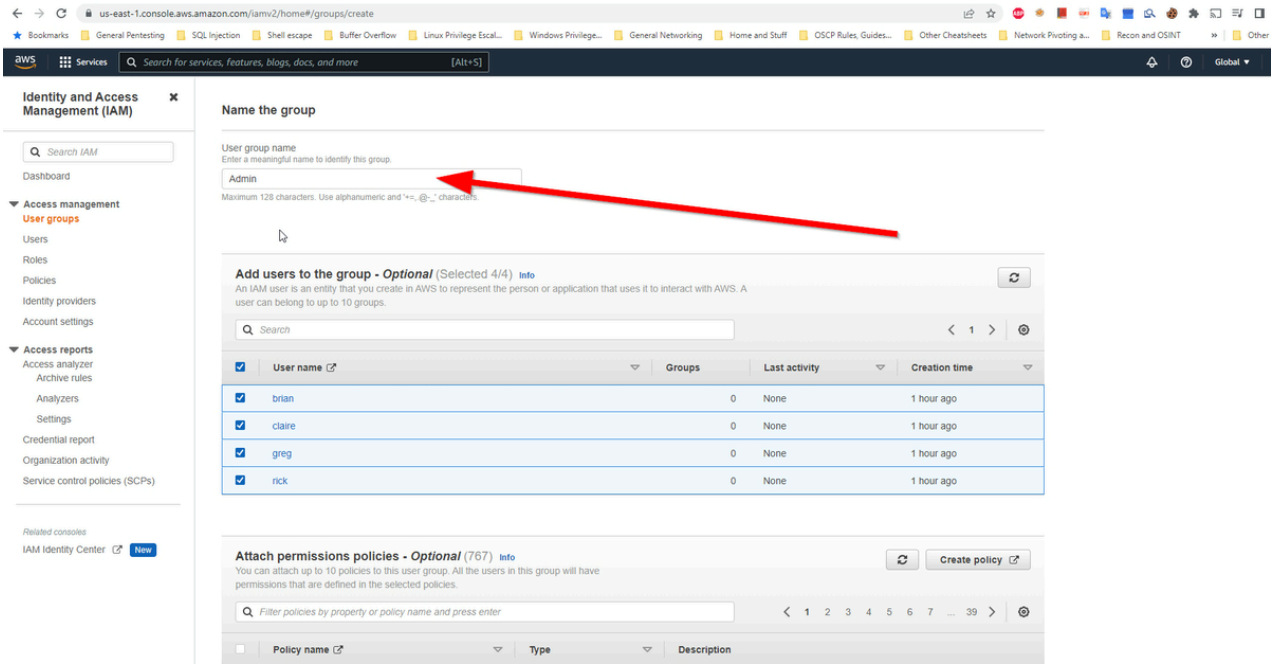


Step 3: Create test groups "Contractors", "MSP", "Admin", "Test"



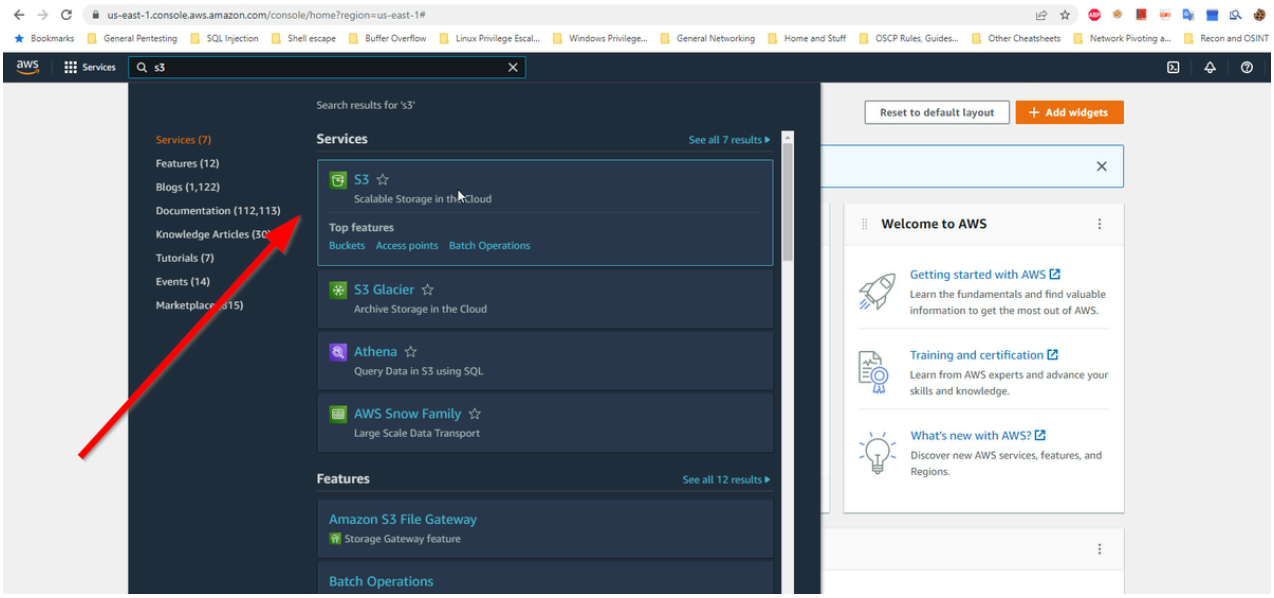

us-east-1.console.aws.amazon.com/console/home?nc2=h_ct&region=us-east-1&src=header-signin#
aws
Services
iam
Search results for 'iam'
Services (6)
Features (17)
Blogs (1,372)
Services
See all 6 results
IAM
Manage access to AWS resources
Top features
Groups Users Roles Policies Access Analyzer
IAM Identity 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
See all 17 results
Groups
IAM feature
Roles
IAM feature
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#/groups
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 > User groups
User groups (0) Info
A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.
Delete
Create group
Filter User groups by property or group name and press enter
Group name
Users
Permissions
Creation time
No resources to display

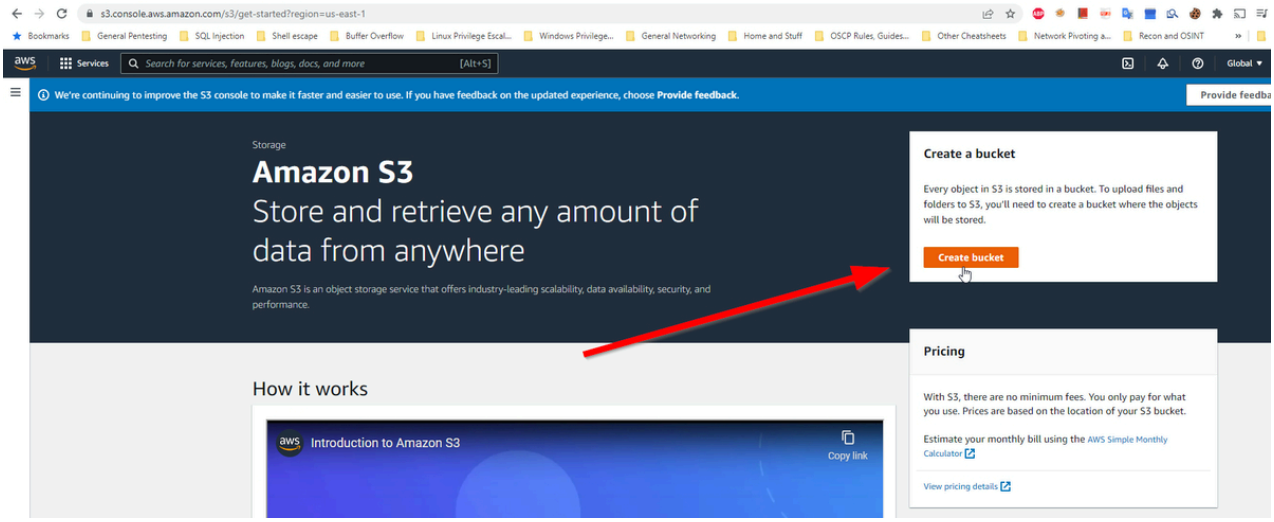


Create Test S3 Buckets.

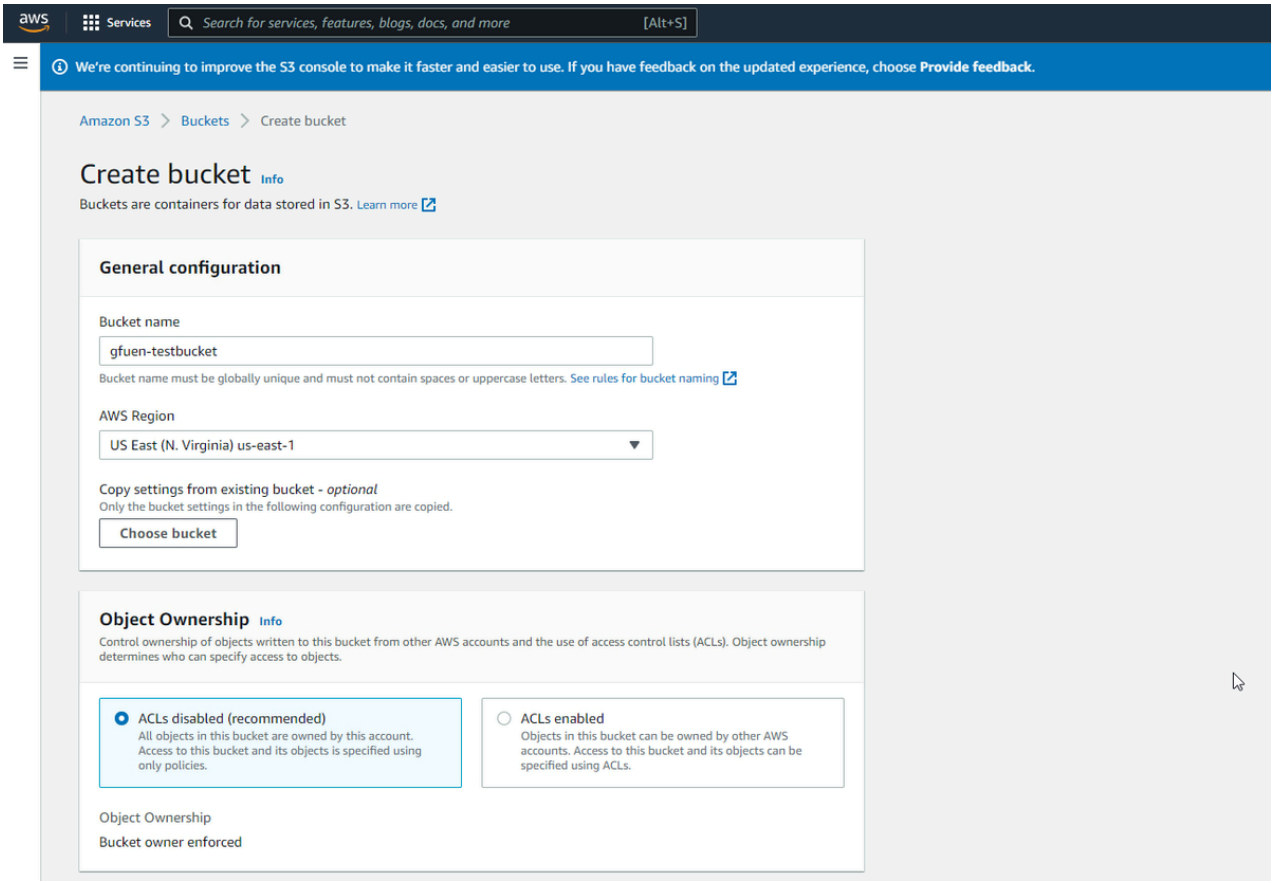
Navigate to the S3 Service page from the AWS Console.

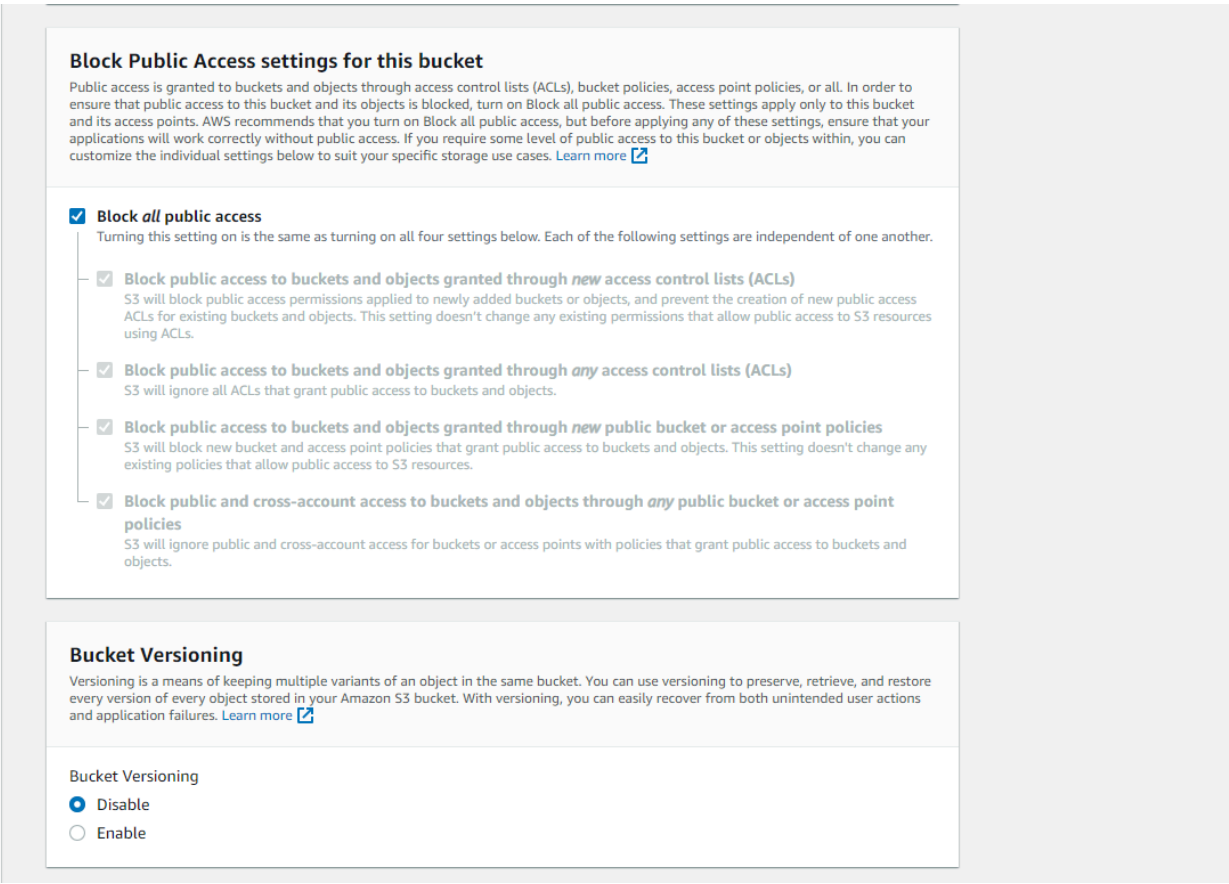


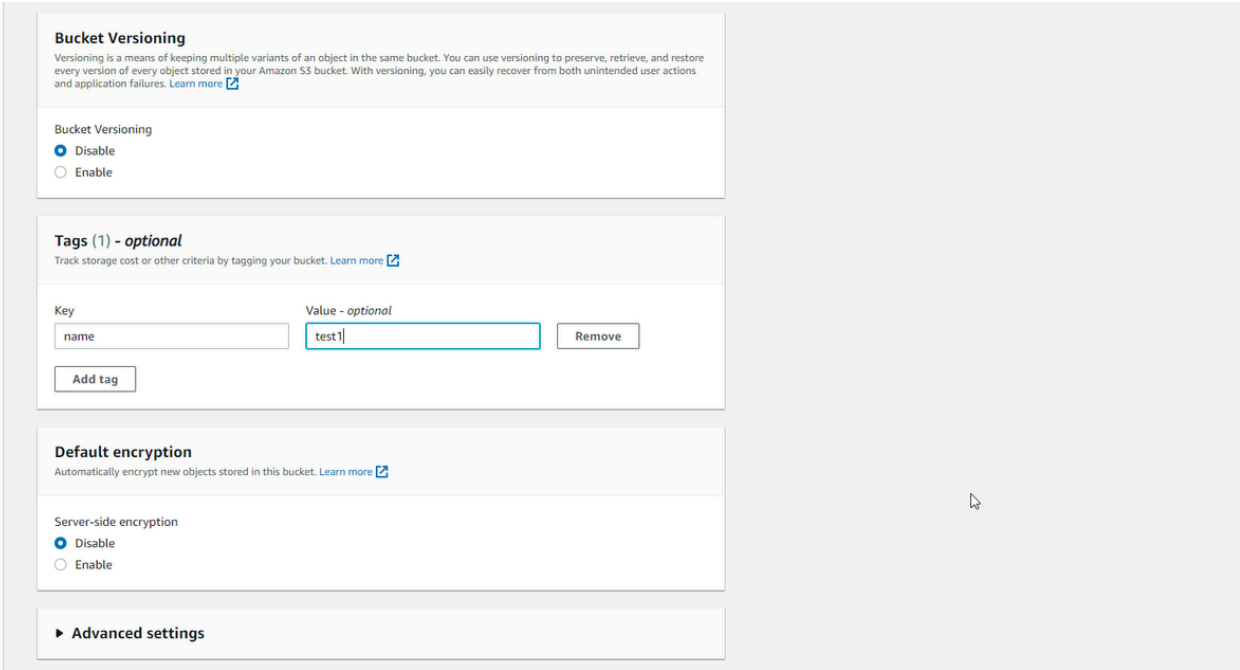
Choose the "Create bucket" option on the S3 homepage of the AWS Console.

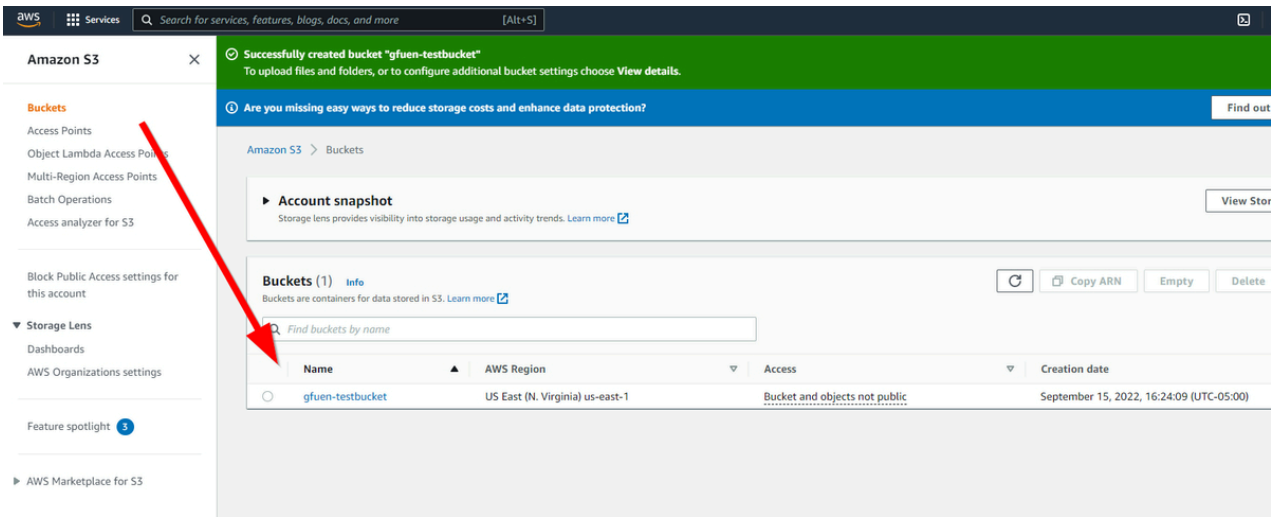


Create an AWS S3 Bucket with the name of "gfuen-testbucket" and "gfuen-testbucket2" for the IAM Policy use cases that will help further in the IAM walkthrough. Be sure to keep all of the safe default configuration options on when creating an S3 bucket especially the checkbox for allowing public access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

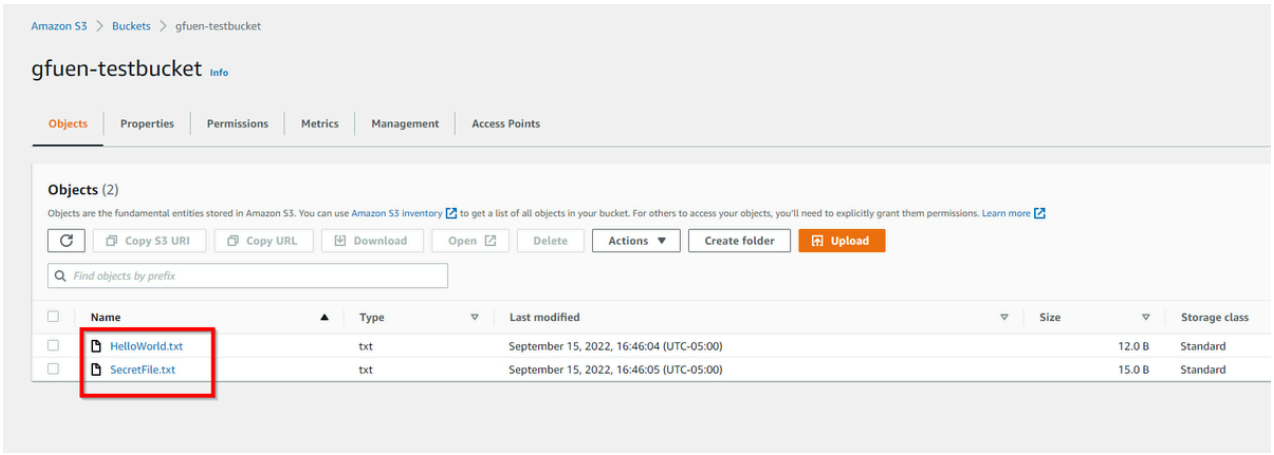



Block Public Access settings for this bucket
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings 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



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.



Contents of HelloWorld.txt:

```
Hello world!
```

Contents of SecretFile.txt:

```
admin:password1
```

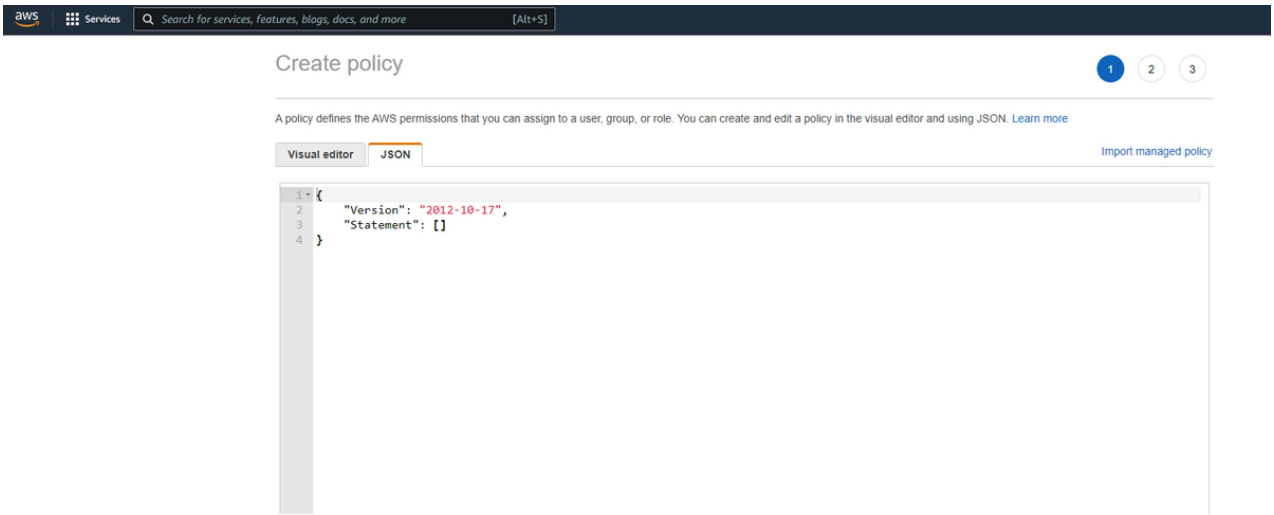
Use Case 1: Allow READ, LIST, and WRITE access to S3 Bucket for IAM User using an

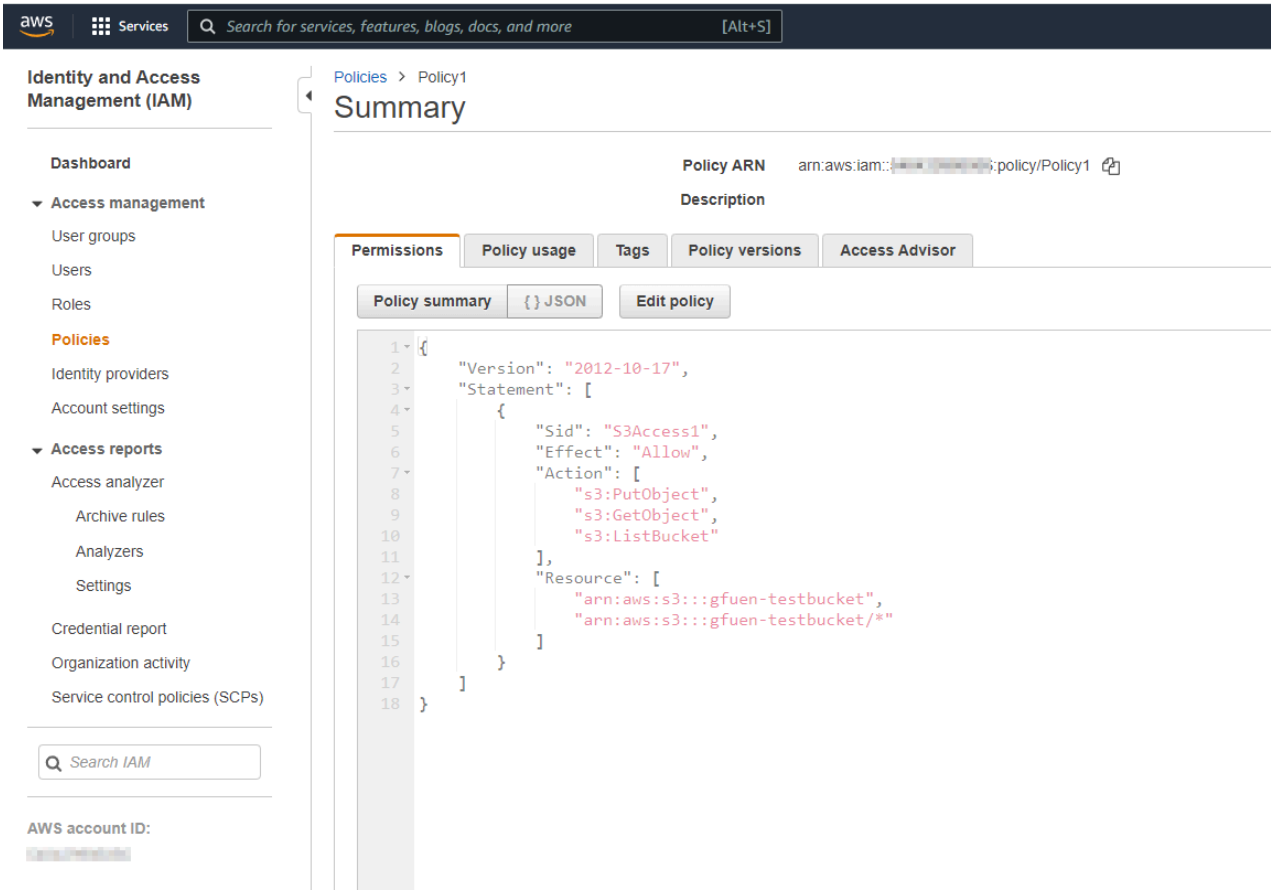
IAM policy.

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/*"
            ]
        }
    ]
}
```





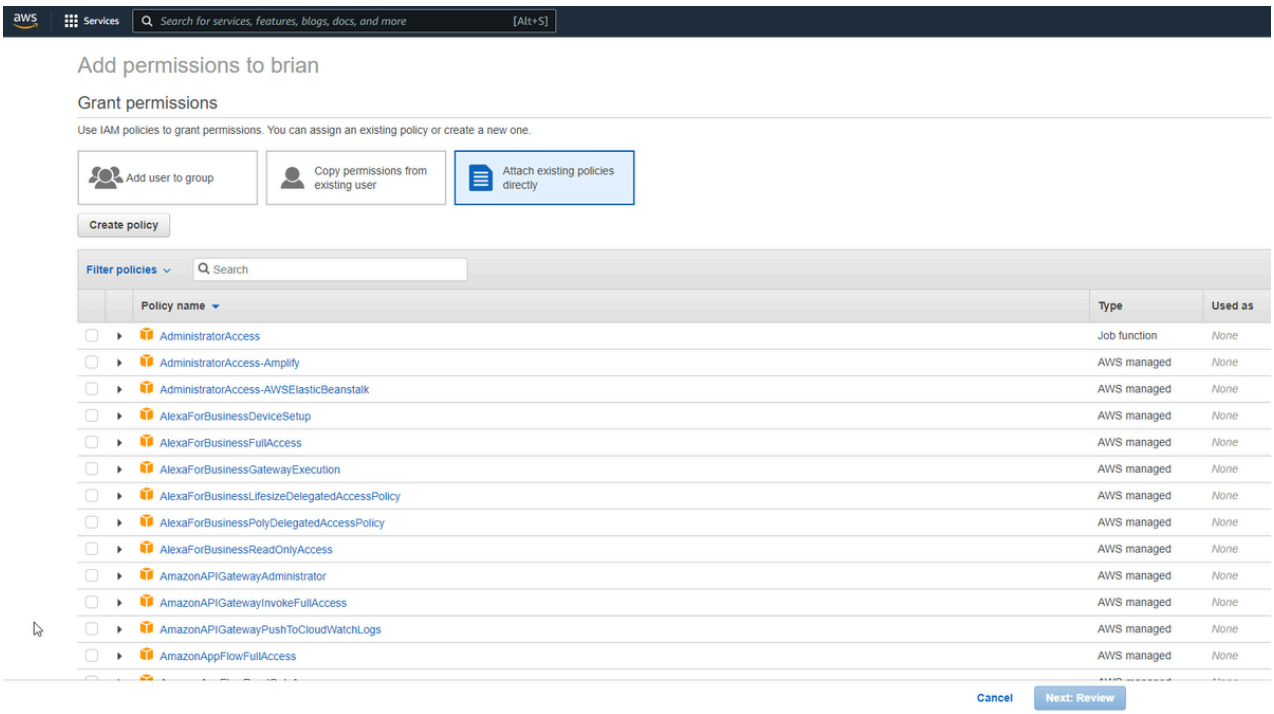
The following is where S3 actions can be found to construct a proper IAM Policy for S3 access.

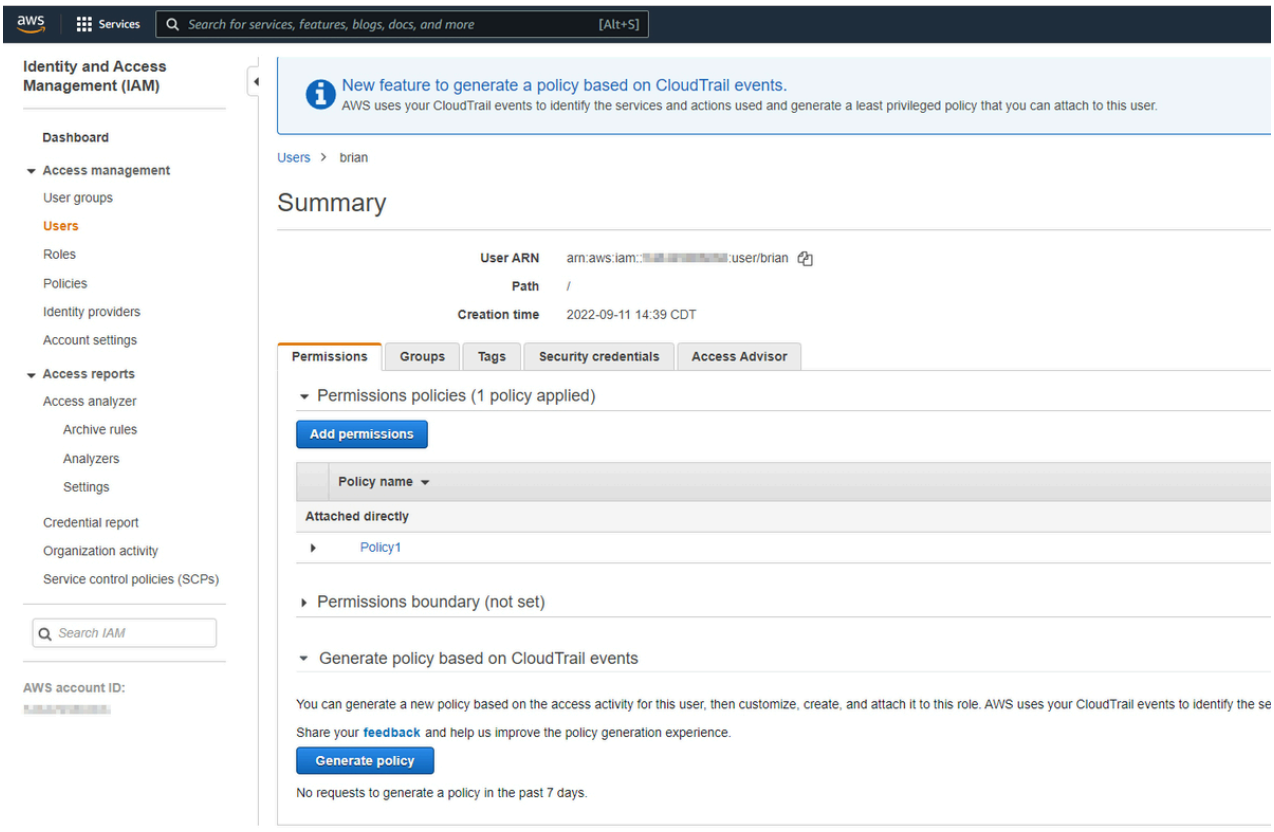
S3 Actions Documentation: https://docs.aws.amazon.com/service-authorization/latest/reference/list_amazons3.html ↗

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

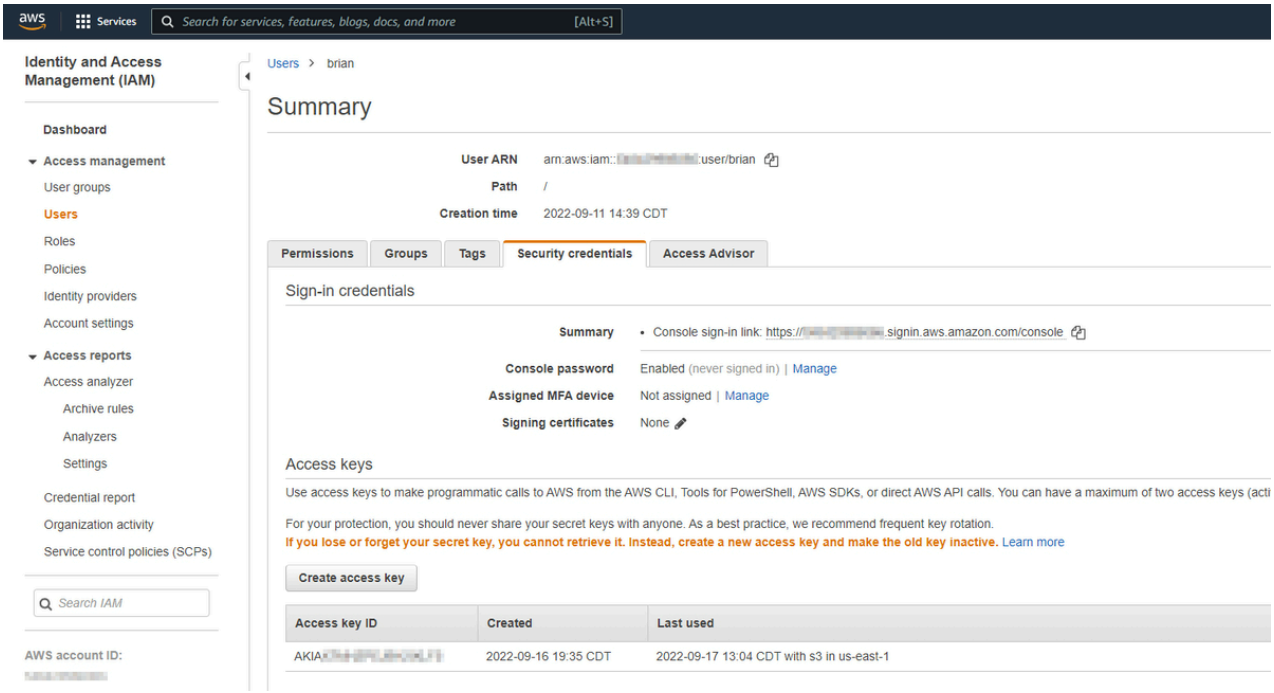
IAM Policy Structure: https://dev.to/tanmaygi/aws-identity-and-access-management-practical-guide-cheat-sheet-3528 ↗

Attack Policy1 to AWS IAM user "brian".

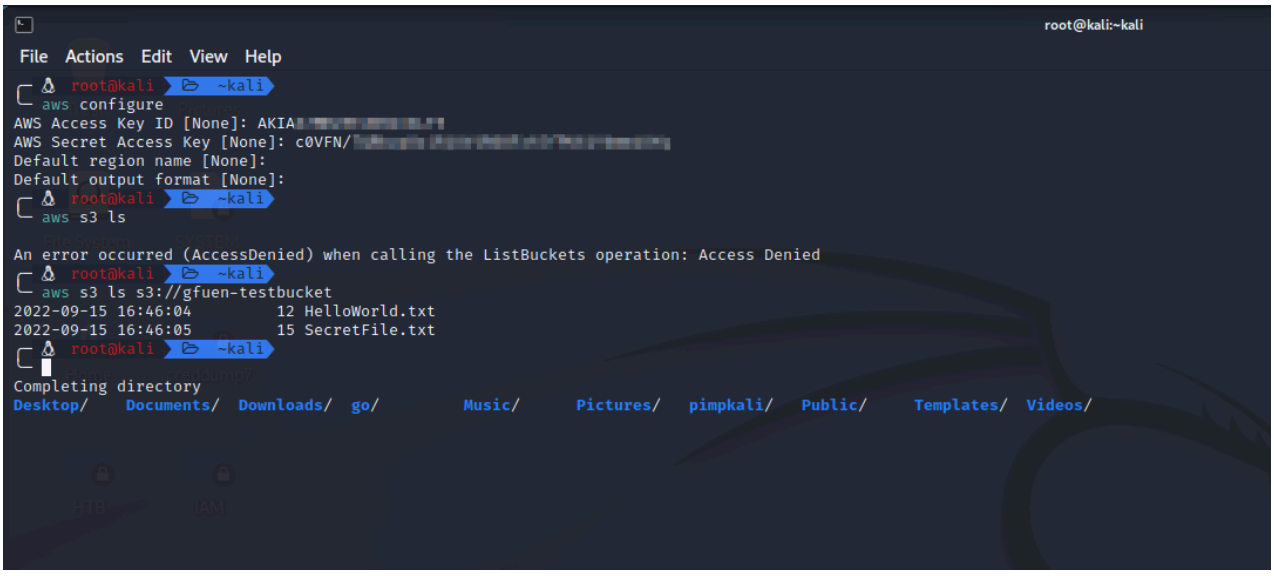




Given this, AWS user "brian" will need to download the Access keys in order to make calls with the AWS cli.

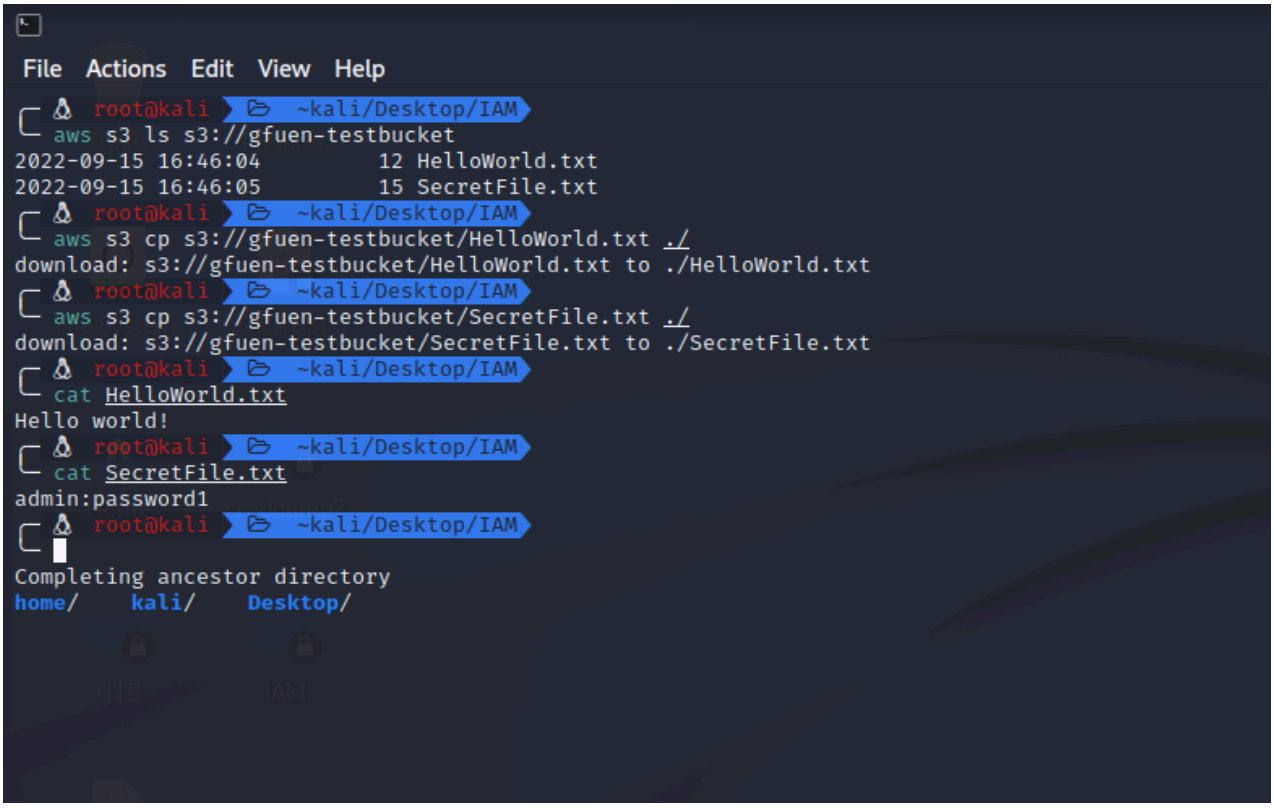


Likewise, now that user "brian" has an attached IAM policy for S3 access "brian" can login into the AWS CLI using his downloaded Access keys using the `aws configure` command.



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↗



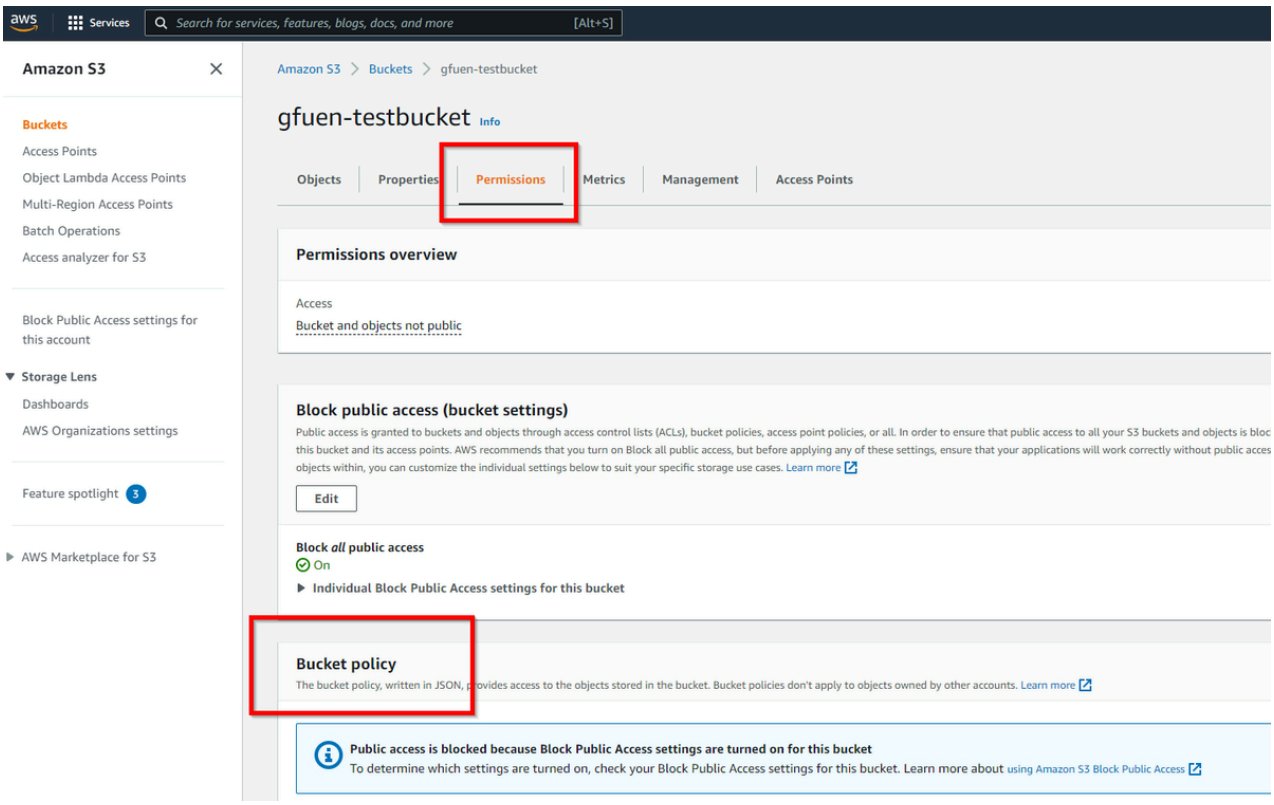
Moreover, in the end I am able to download and read all the files as AWS user "brian" from S3 "gfuen-testbucket".

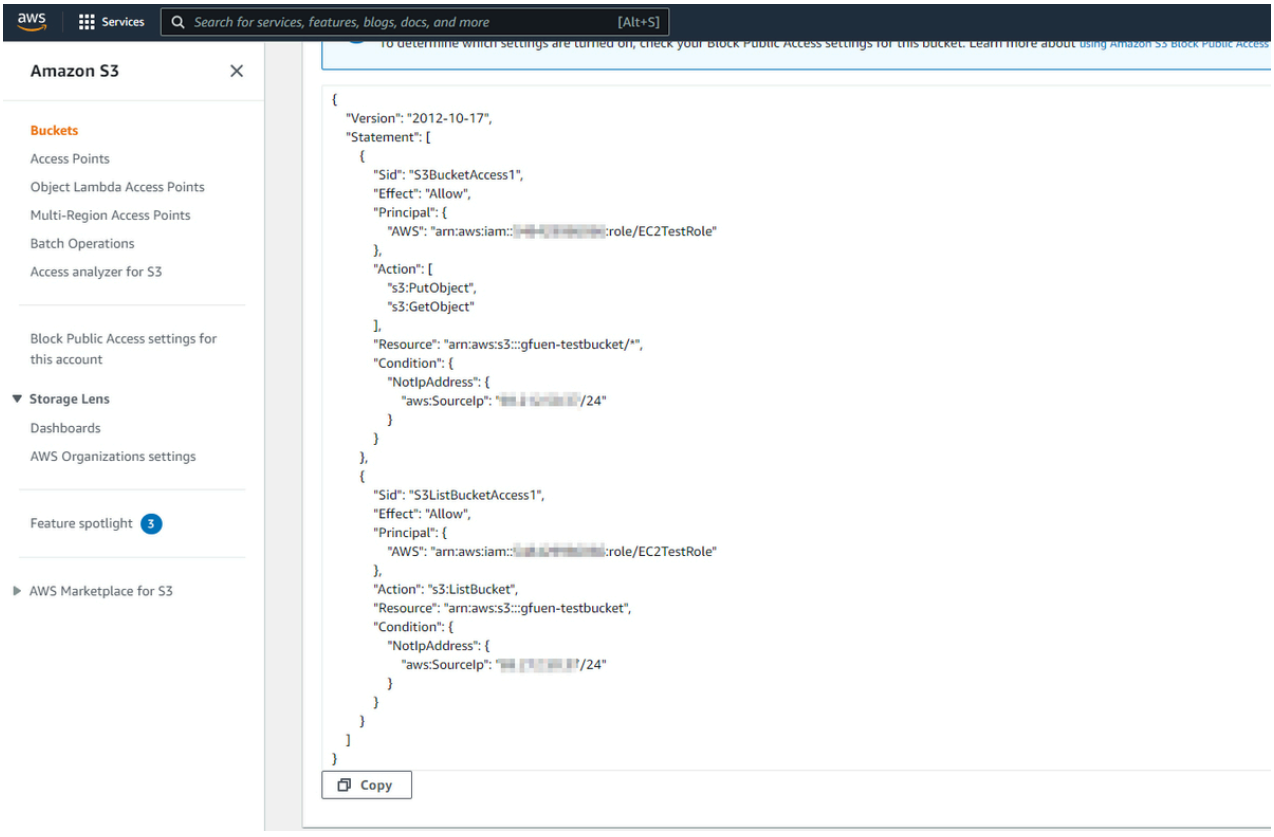
Use Case 2: Allow READ, LIST, and WRITE access to S3 Bucket for IAM Role for an S3 Resource Policy utilizing SourceIP.

Create an S3 Resource Policy for role "EC2TestRole" to access to all objects in S3 bucket "gfuen-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.

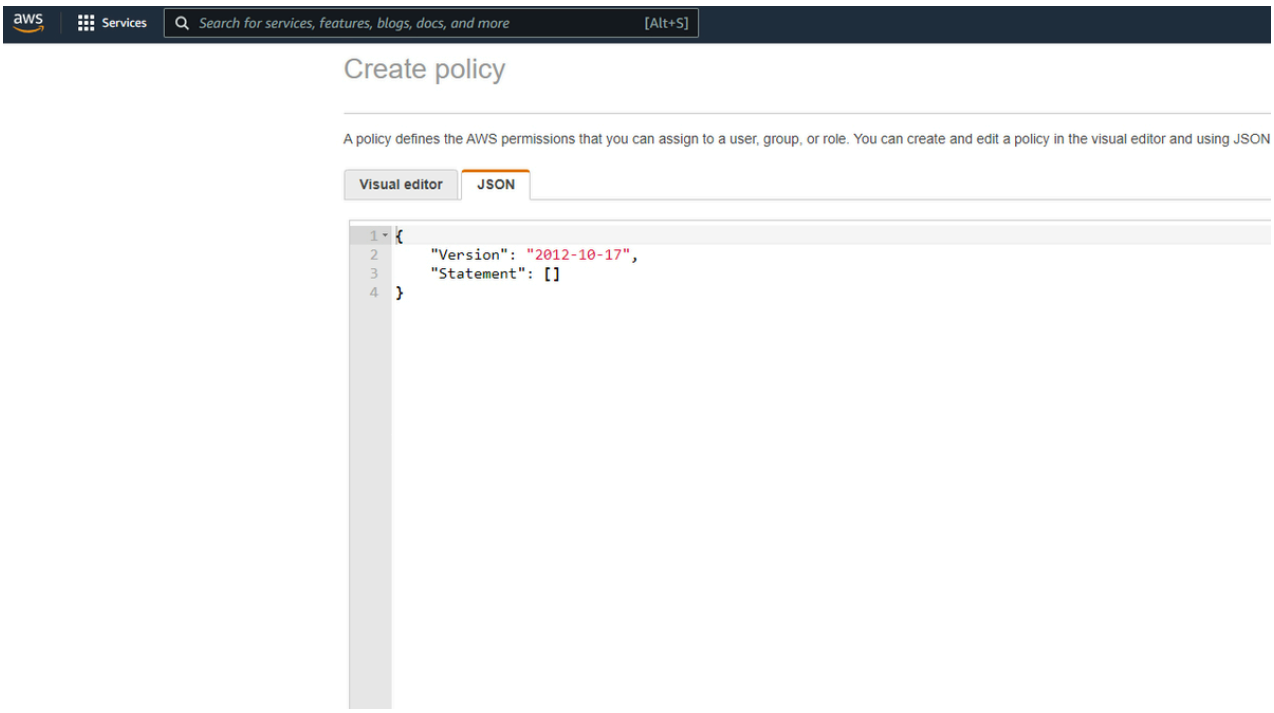
Bucket Policy

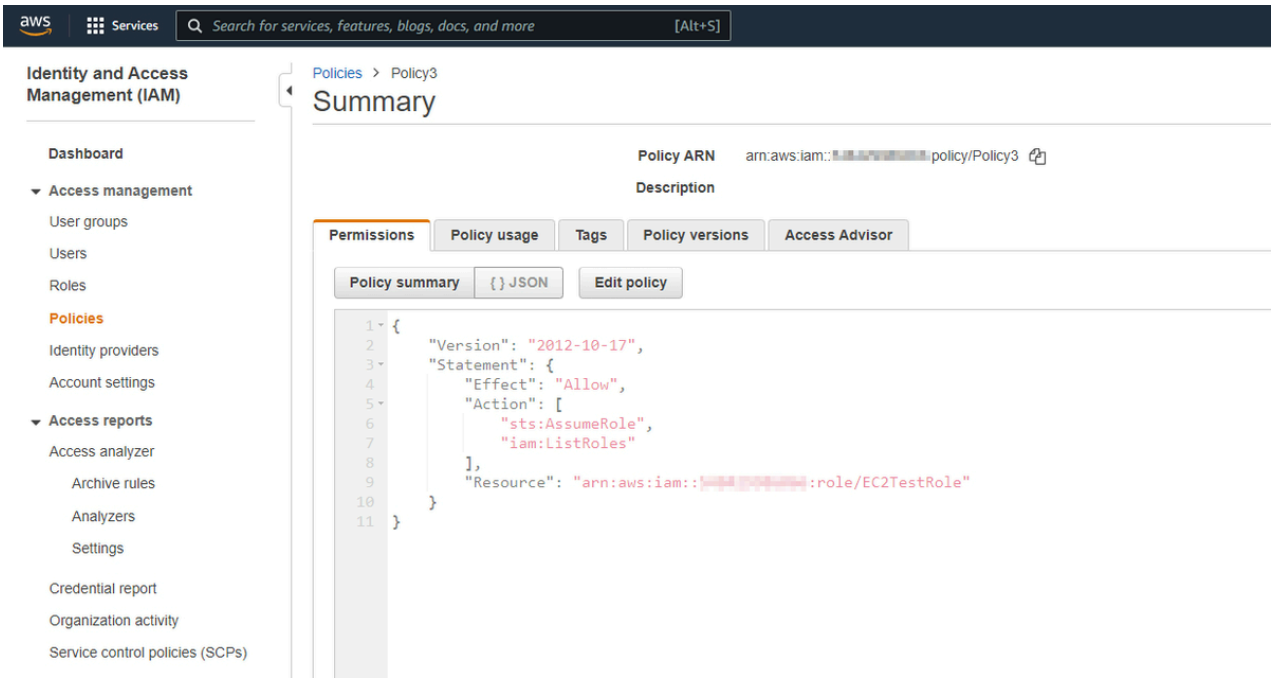
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3BucketAccess1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::<aws_account_id>:role/EC2TestRole"
            },
            "Action": [
                "s3:PutObject",
                "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"
                }
            }
        }
    ]
}
```





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.

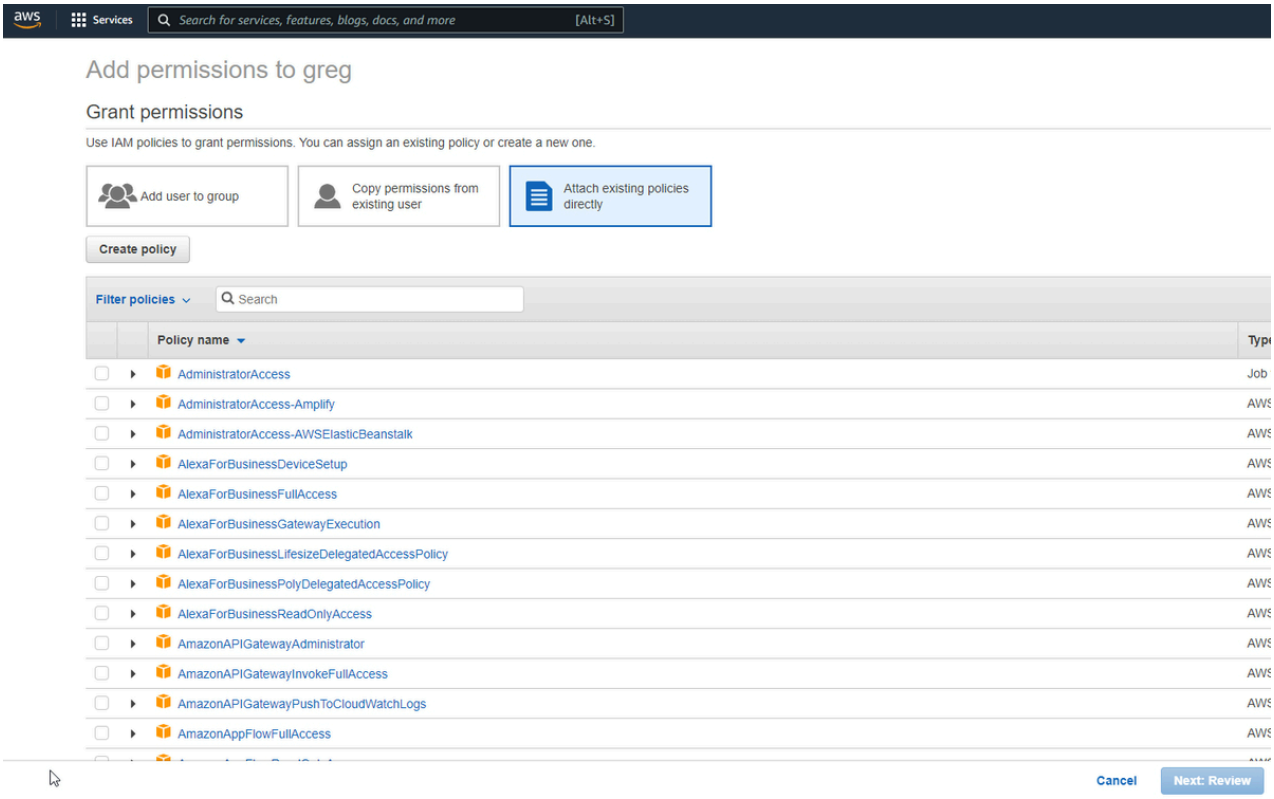


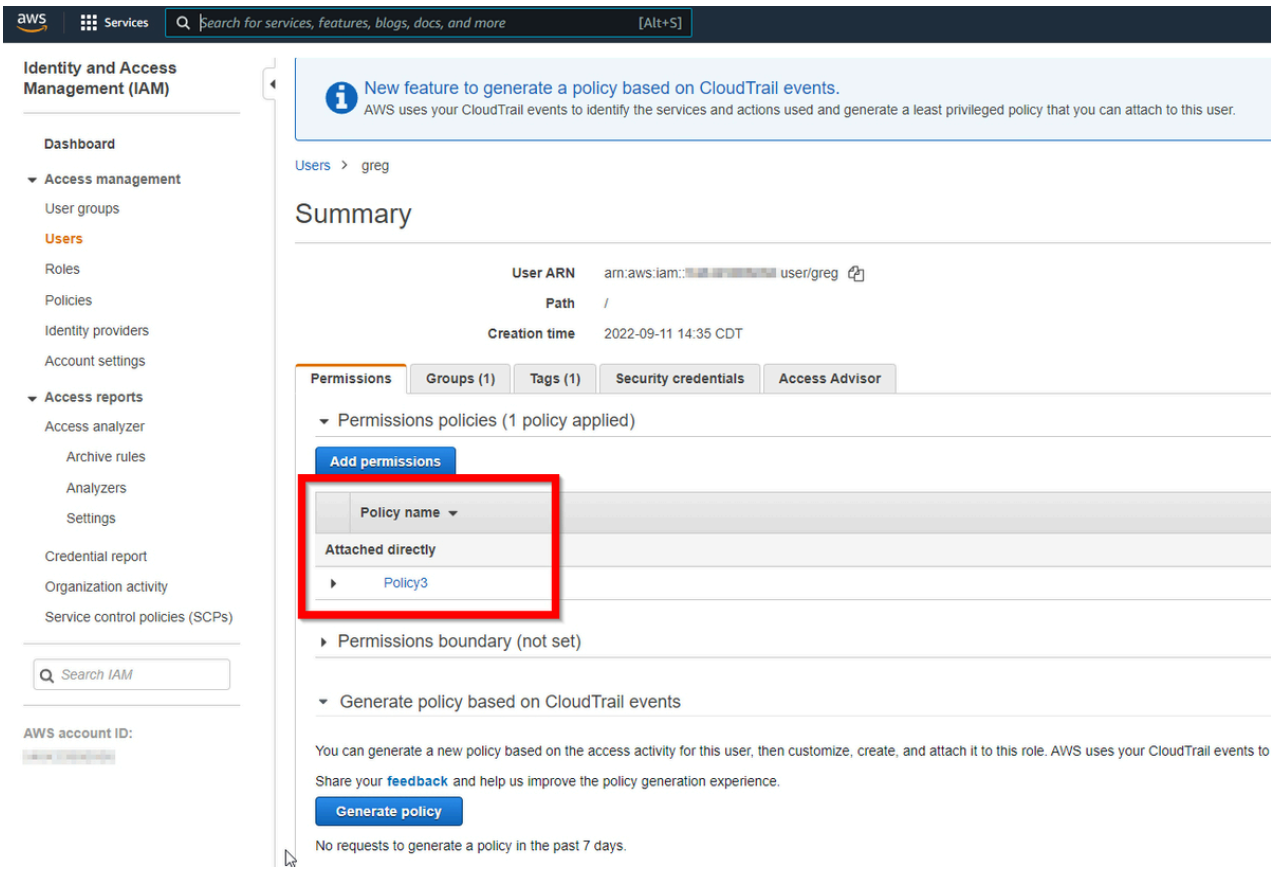


Policy3.json

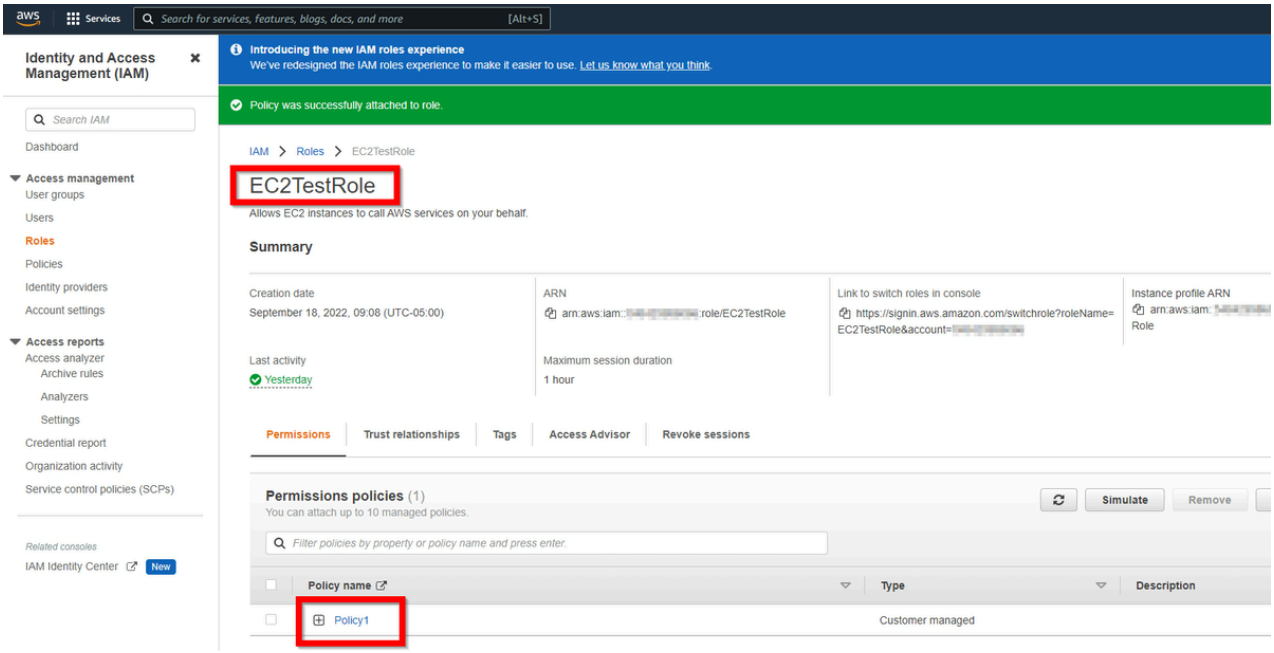
```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "sts:AssumeRole",
            "iam:ListRoles"
        ],
        "Resource": "arn:aws:iam::<aws_account_id>:role/EC2TestRole"
    }
}
```

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





Furthermore, attach the following "Policy1.json" to AWS role "EC2TestRole" in order to enable S3 access to 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/*"
            ]
        }
    ]
}
```

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".

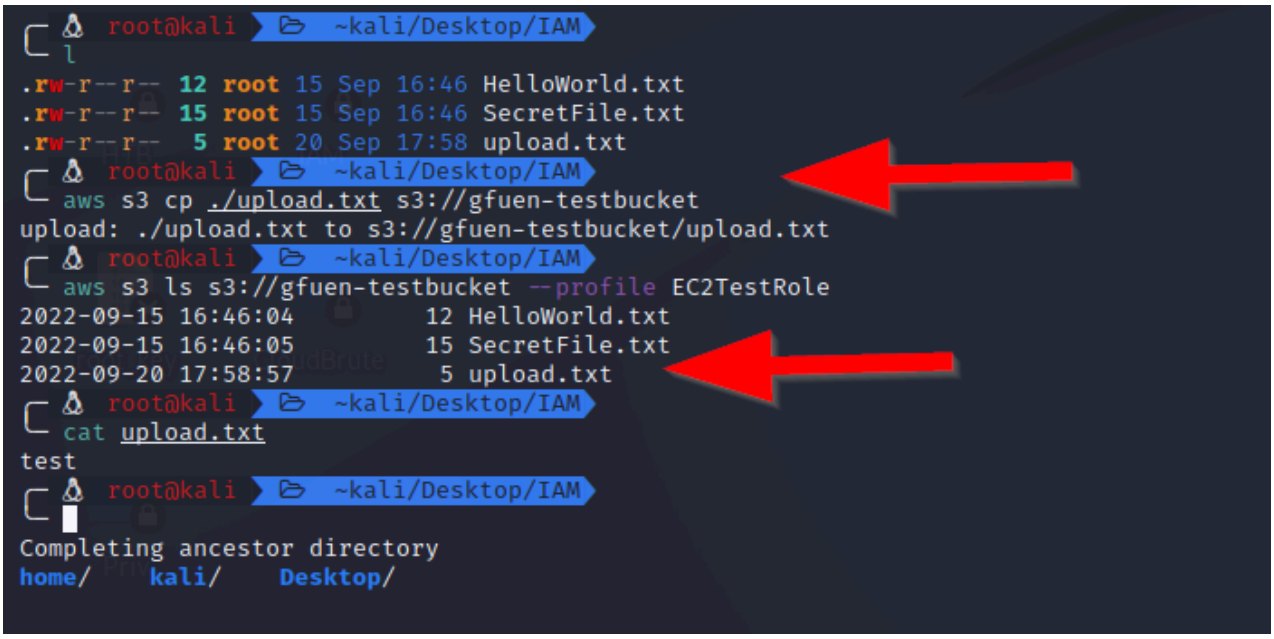
AWS Assume-Role Documentation:
https://docs.aws.amazon.com/cli/latest/reference/sts/assume-role.html ↗

List the roles in the AWS account.

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

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

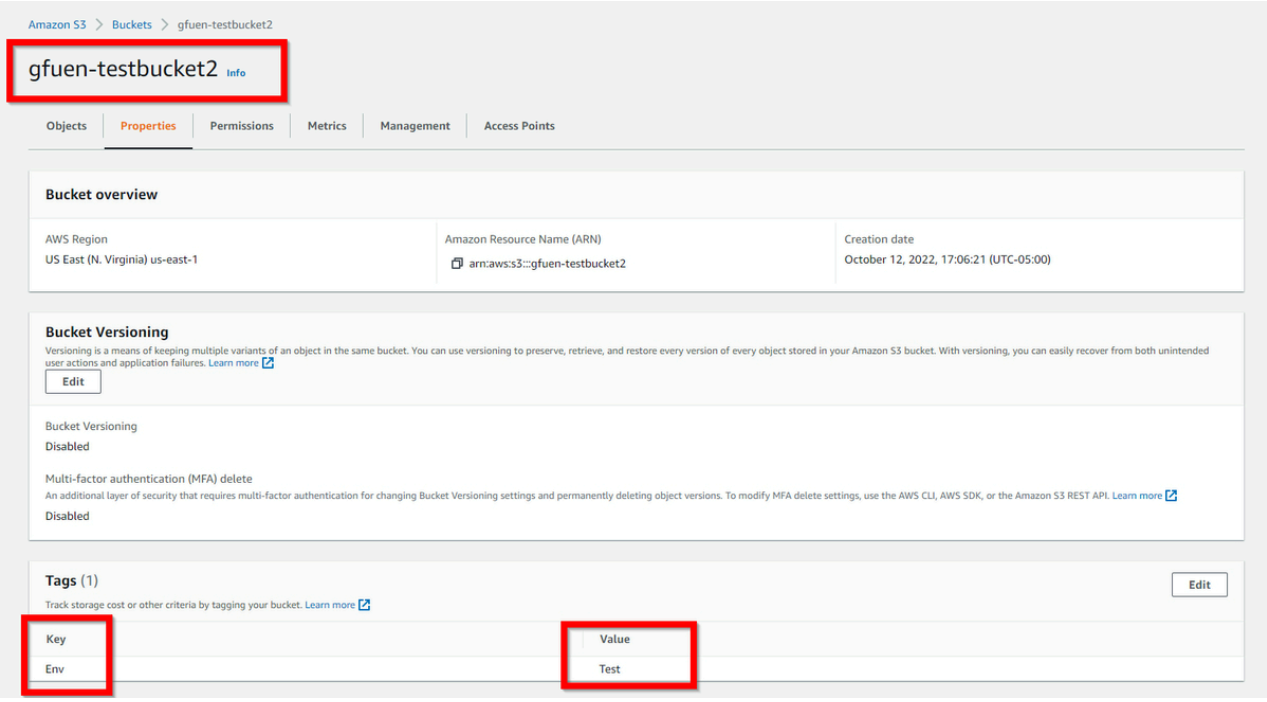
Moreover, in the end I am able to download and read all the files as role "EC2TestRole" from S3 "gfuen-testbucket".



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.

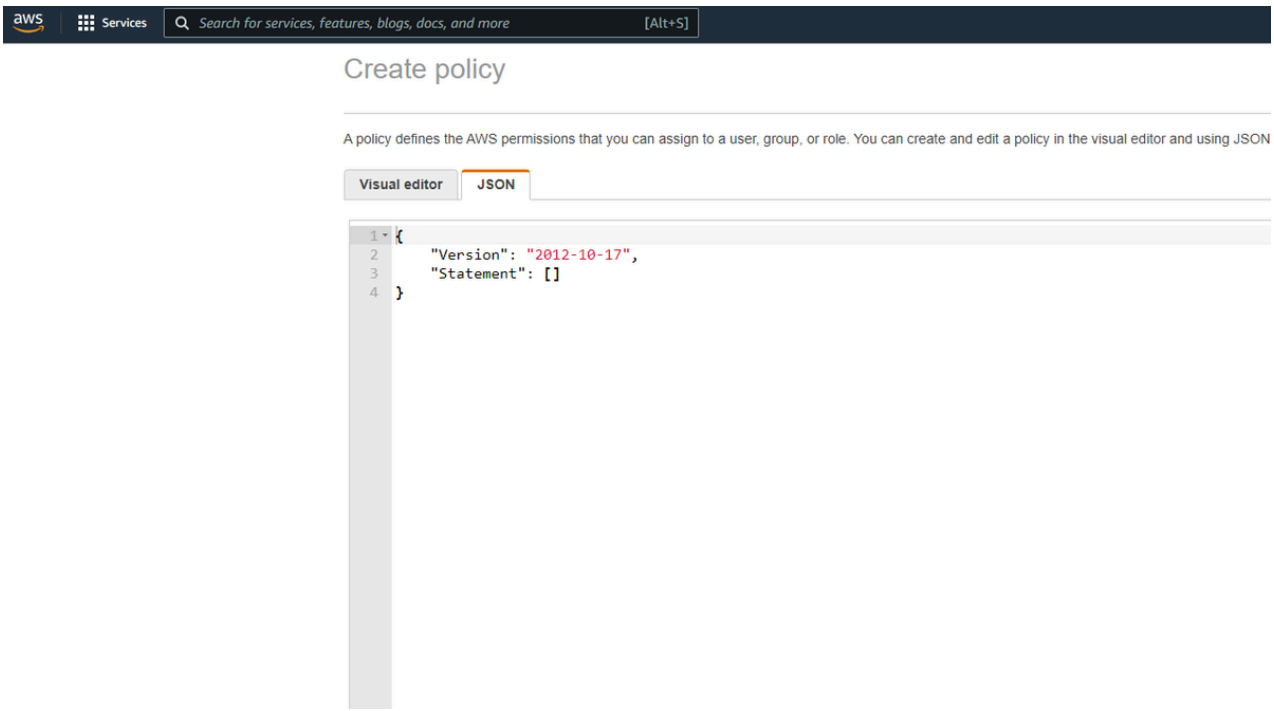
Add the resource tag pair "env" "Test" to the S3 bucket "gfuen-testbucket2" in order to simulate an S3 bucket in an enterprise that should only be available to those on the Testing team.

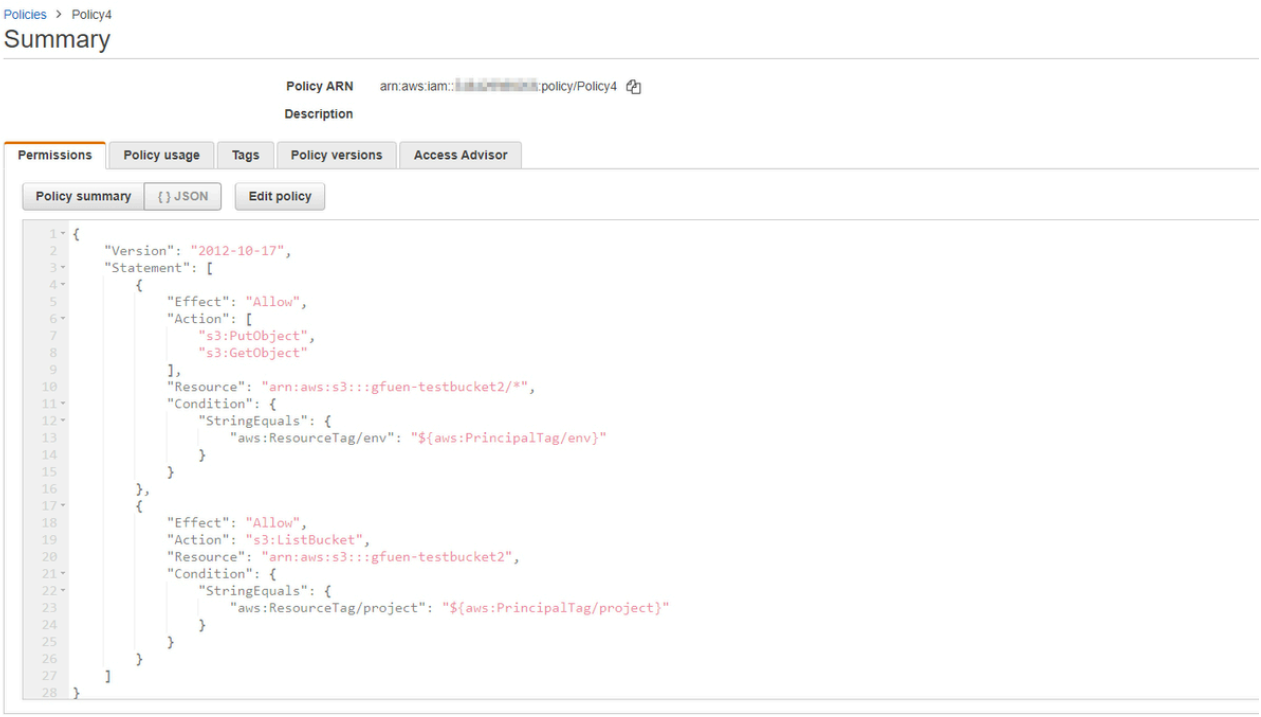


Create an S3 IAM Policy that can be attached to IAM User "greg" to access to all objects in S3 bucket "gfuen-testbucket2" based on a Resource tag for more granular secure access.

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": "${aws:PrincipalT
            }
        },
        {
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::gfuen-testbucket2",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/project": "${aws:Princi
            }
        }
    ]
}
```

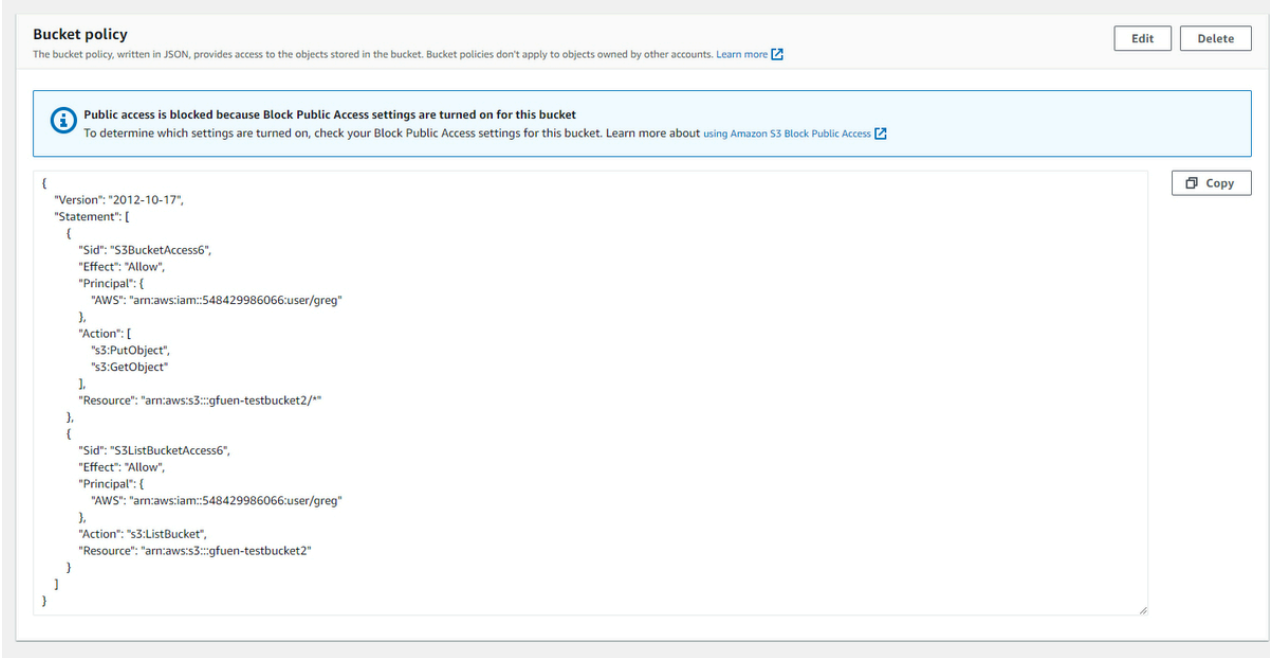


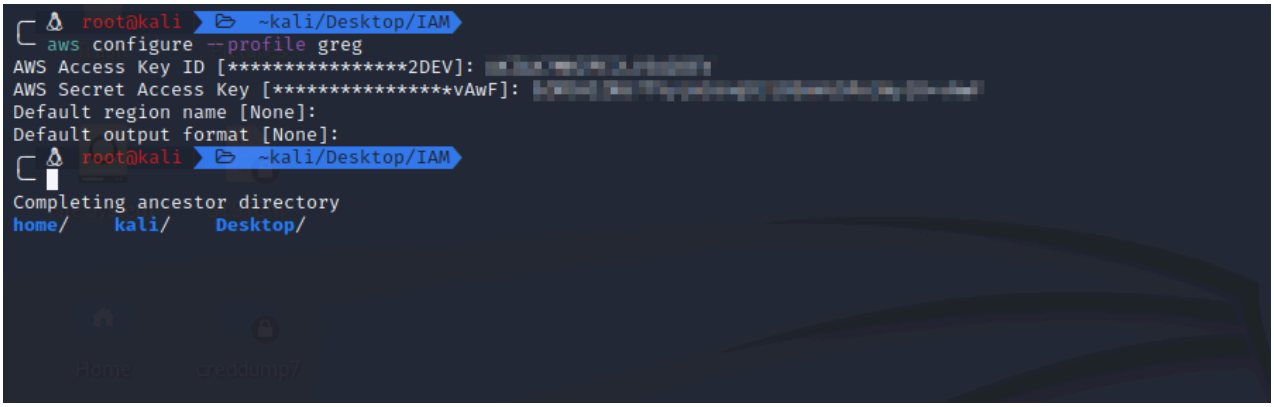


Create an S3 Resource Bucket Policy for IAM User "greg" to access to all objects in S3 bucket "gfuen-testbucket2". Navigate to the specified S3 bucket and under the permissions tab one can edit the S3 Resource policy for the selected S3 bucket to restrict access.

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"
        }
    ]
}
```




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