AWS Project Screenshots

Samuel Diaz

MIS 690: Cloud Computing

San Diego State University

November 01, 2024

Table of Contents

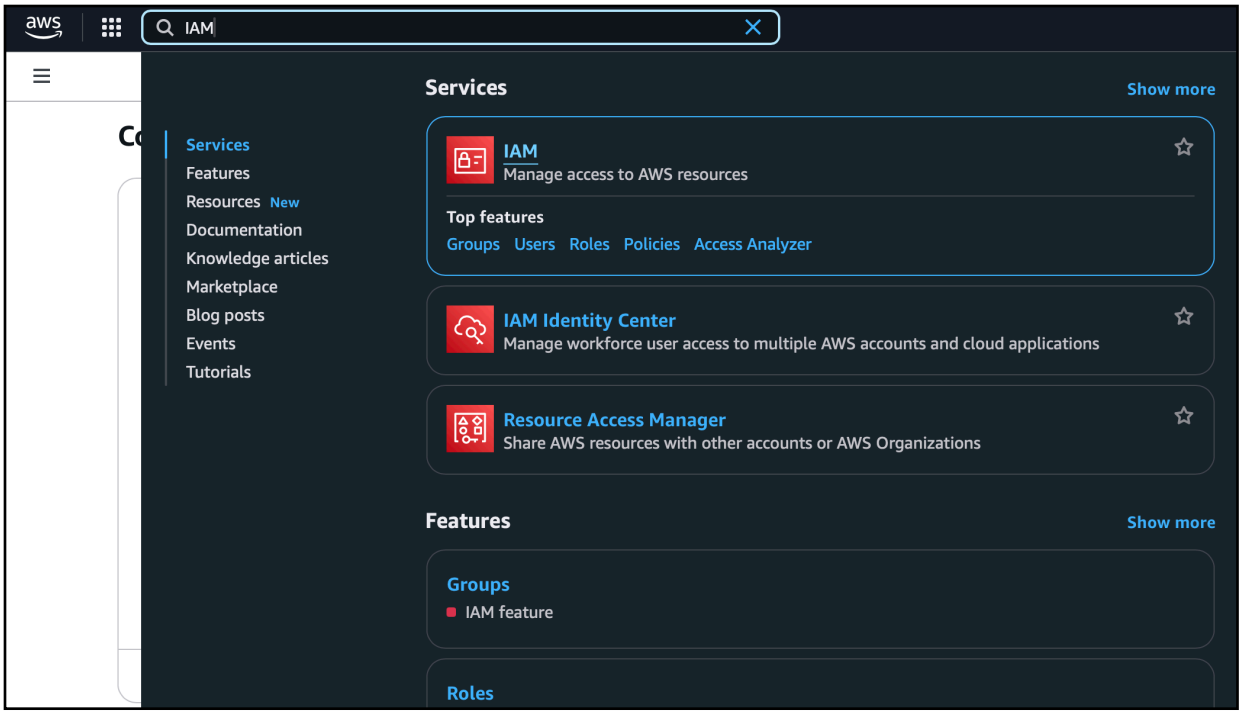
A. AWS IAM (Identity and Access Management)

i. Create ‘Admin’ User

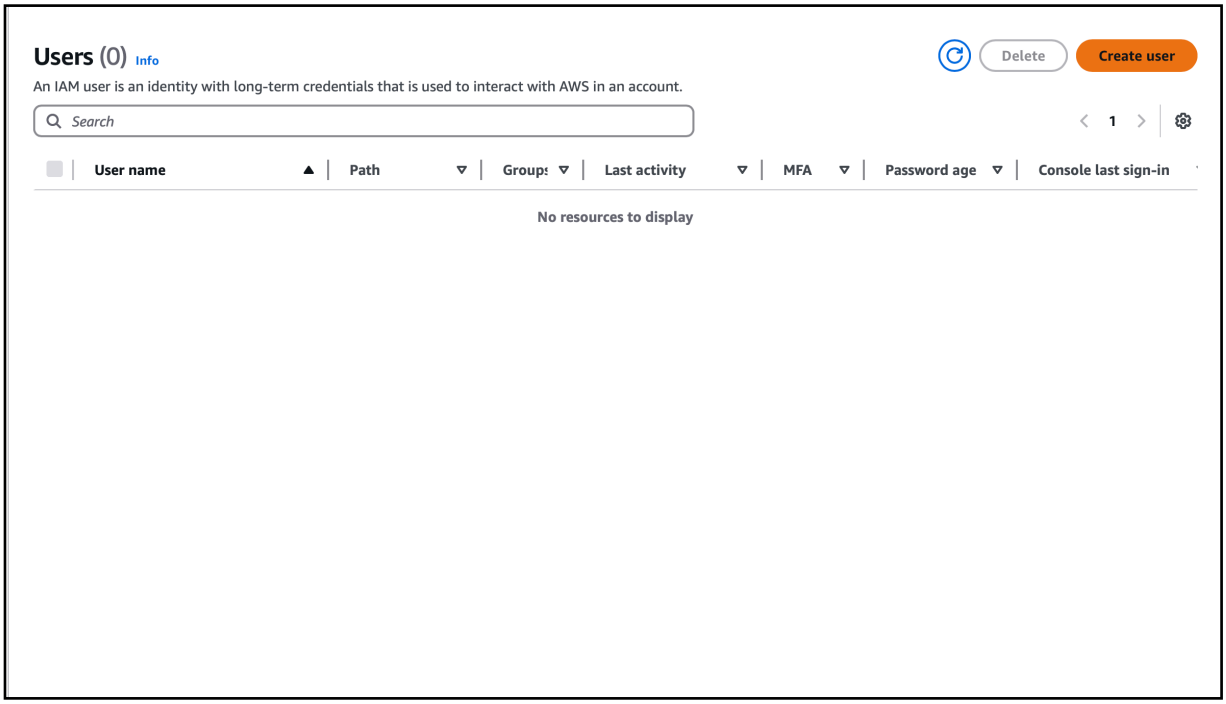
Reasoning: The root user has unrestricted access to all AWS services and resources in the account. If the credentials were compromised, the attacker would have complete control over all resources through my AWS account. These can be a breach of confidentiality and availability, leading to significant financial or data loss. Therefore, it is best to restrict the use of the root account for only initial setup and critical administrative tasks.

Steps:

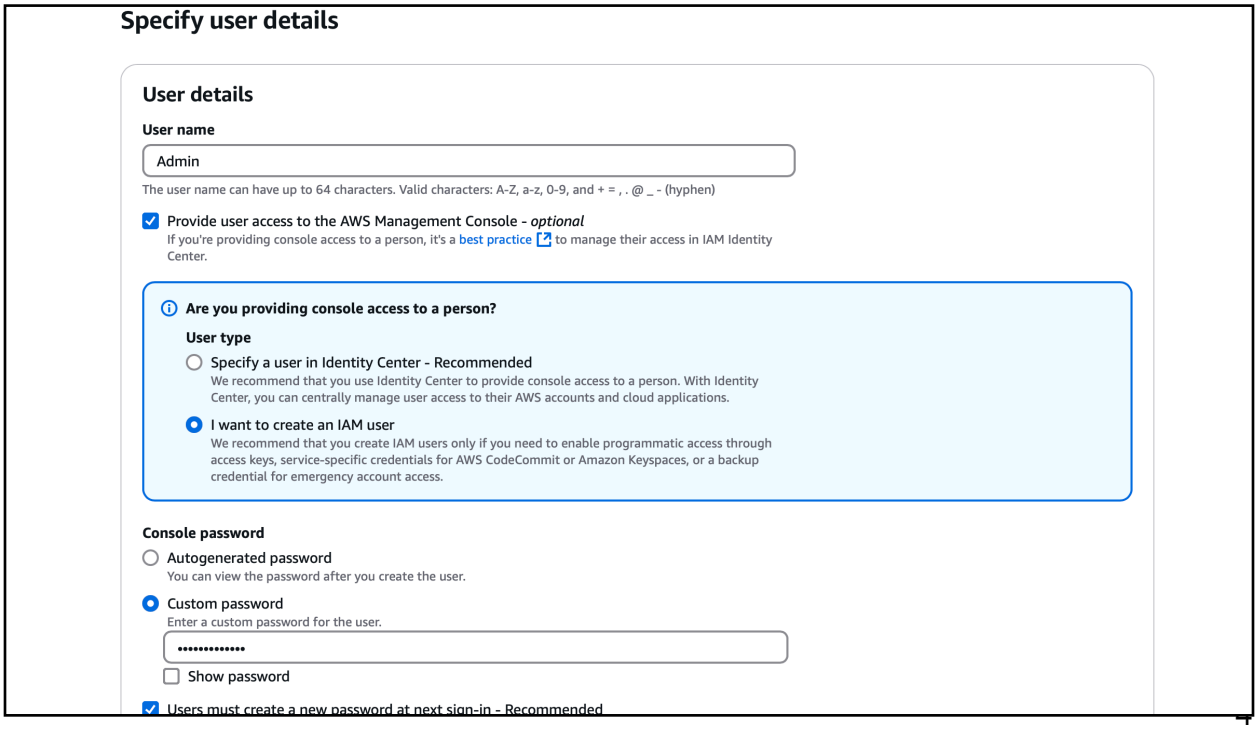
1. AWS Homepage > Search bar > type and select ‘IAM’



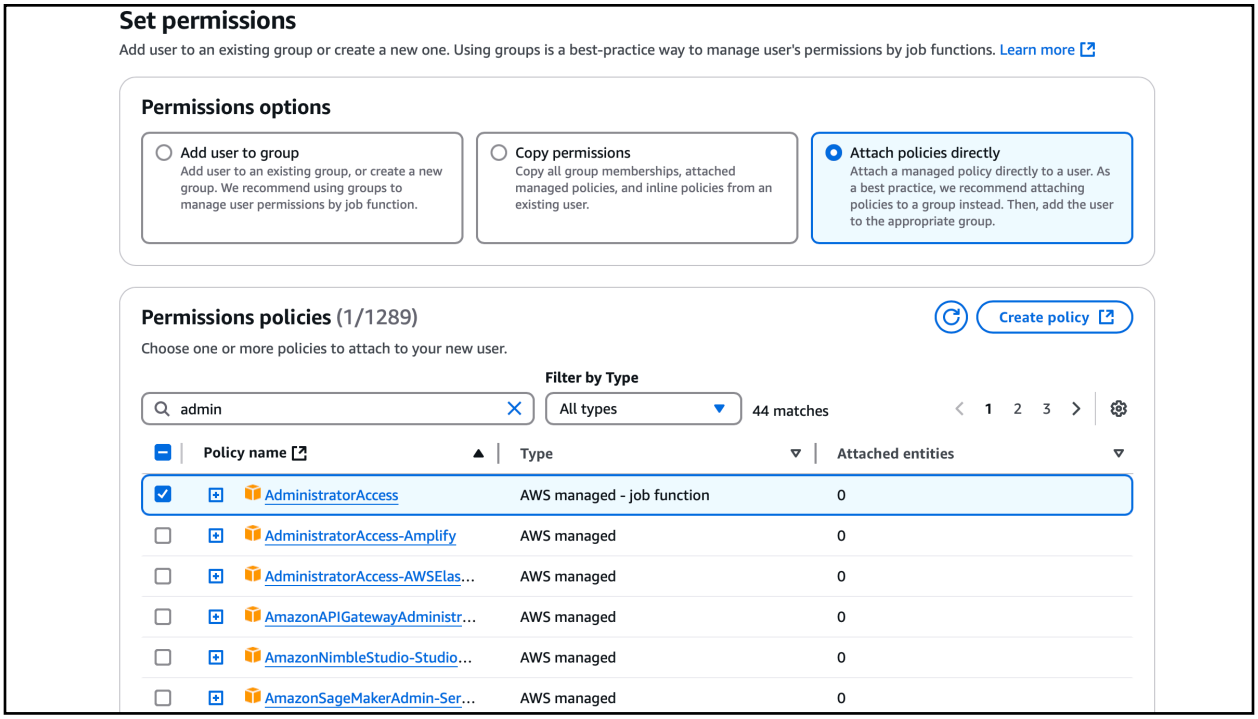
2. Access Management > Users > ‘Create User’



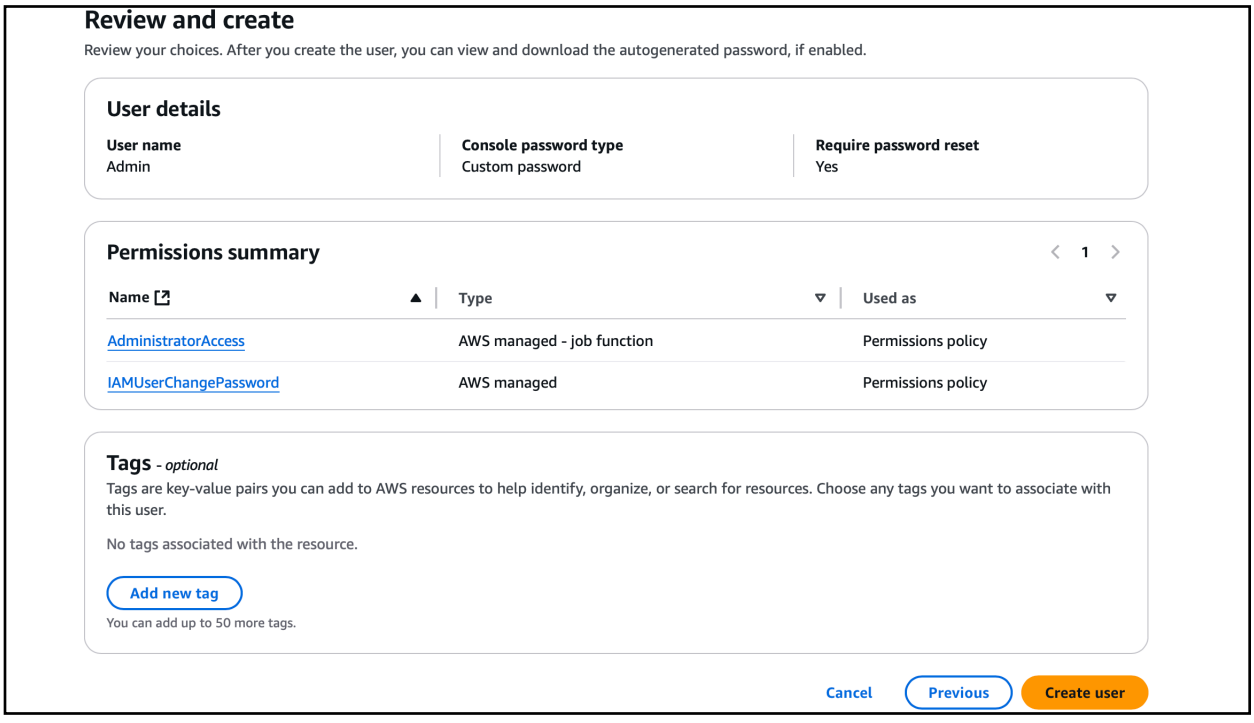
3. Specify user details > Username > ‘Admin’ > check ‘Provide user access to the AWS Management Console - optional’ > select ‘I want to create an IAM user’ > Console password > Custom password > type in temporary password > ‘Click next’



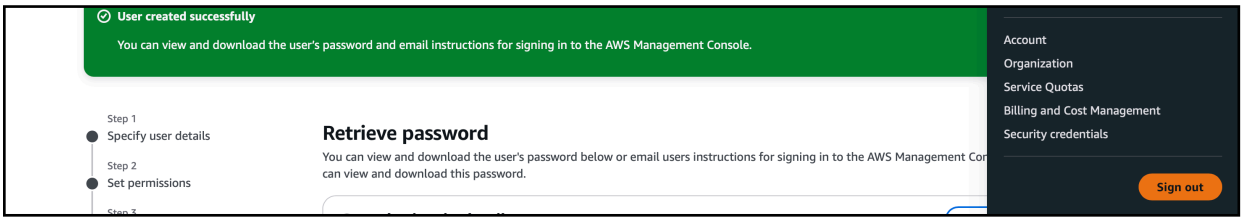
4. Set Permissions > 'Attach Policies Directly' > Permission policies > search ‘admin’ > select ‘AdministratorAccess’ > Click next



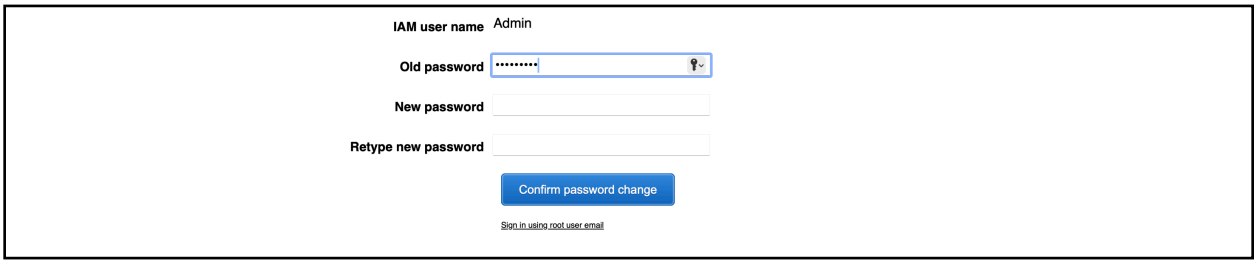
5. Review and verify details > Select ‘Create User’



6. On drop-in page > Copy the Console sign-in URL > From top right, select account dropdown > click ‘Sign out’



7. Paste the provided sign in link into a web browser > sign-in with the Admin credentials you created > You will prompted to change your password > Confirm password change

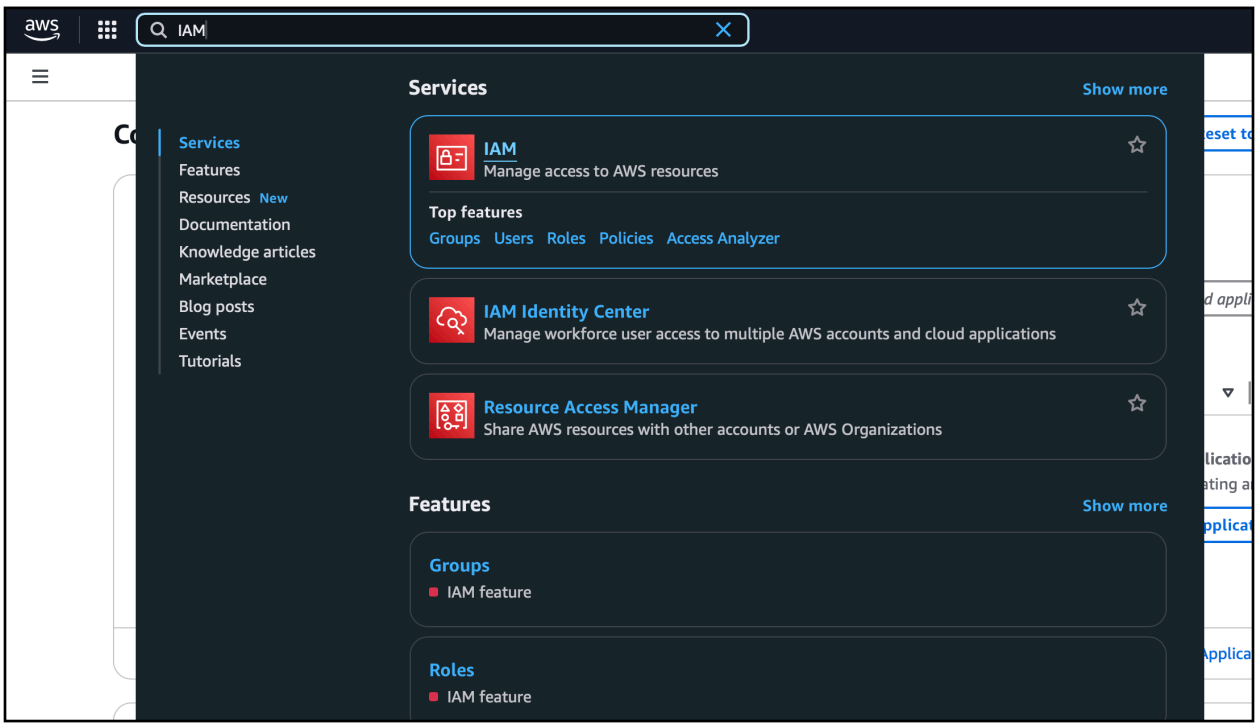


8. Successfully created and logged into Admin account.

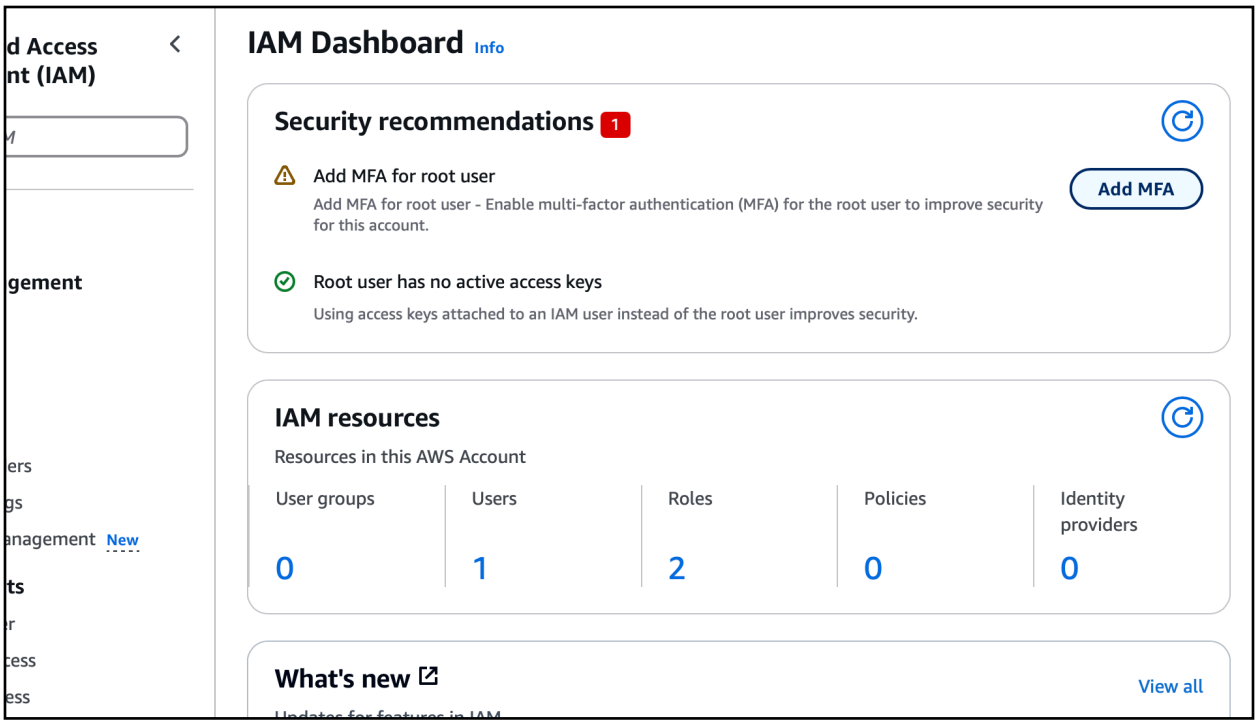
ii. Setup MFA For Root User

Reasoning: MFA adds an additional layer of protection to accounts by requiring both ‘something you know’, your password and ‘something you are’, the passkey (finger or face to unlock).

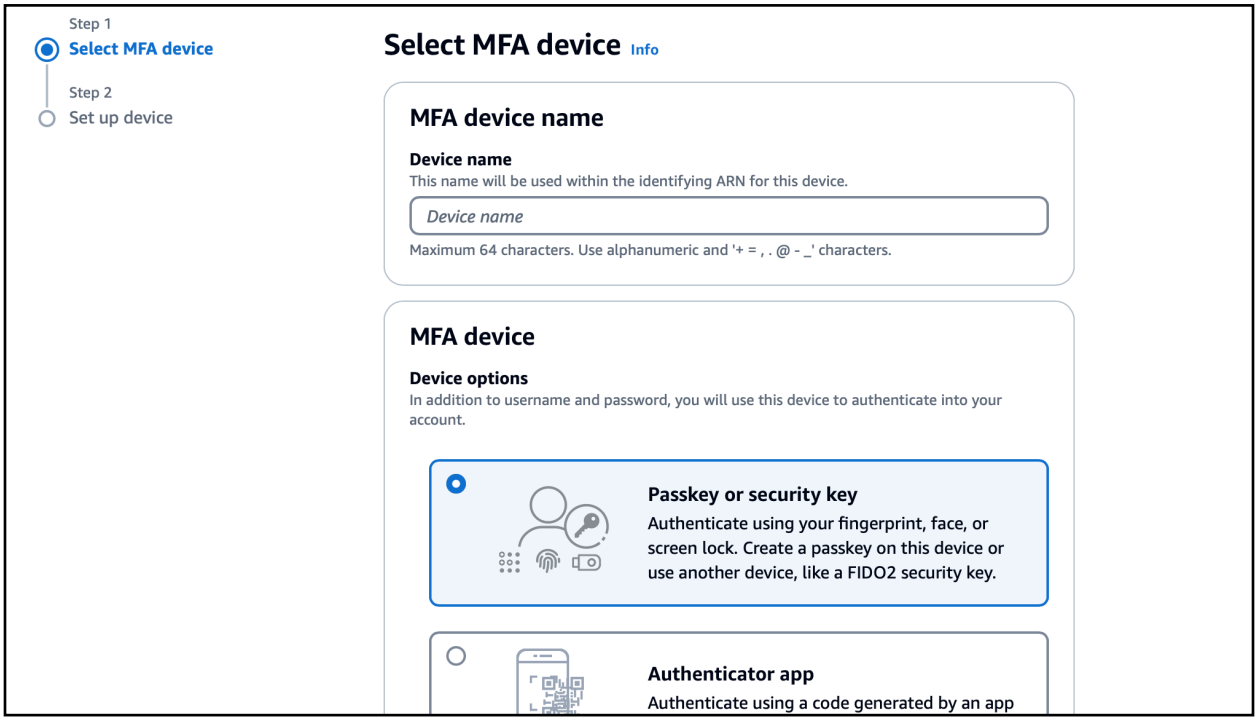
1. As root user > AWS Homepage > Search bar: type and select ‘IAM’



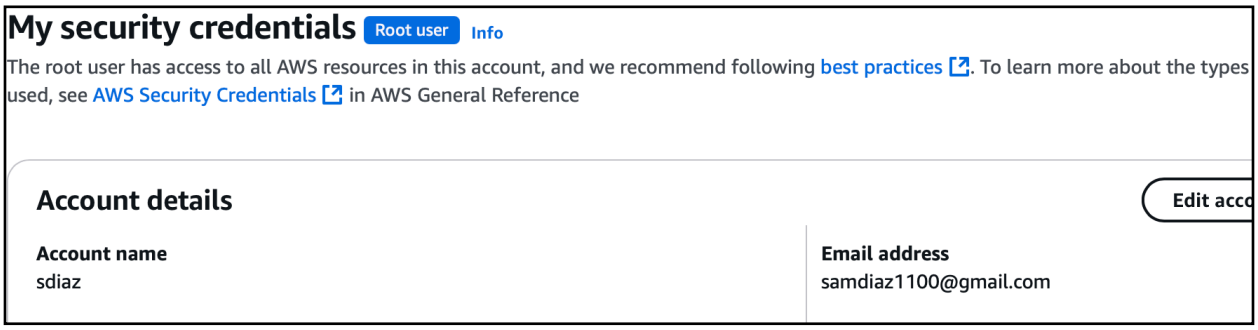
2. IAM Dashboard > Security Recommendations > ‘Add MFA’



3. Select MFA device > Device Name: type ’my_passkey’ > MFA device: select ‘Passkey or security key’ > Next

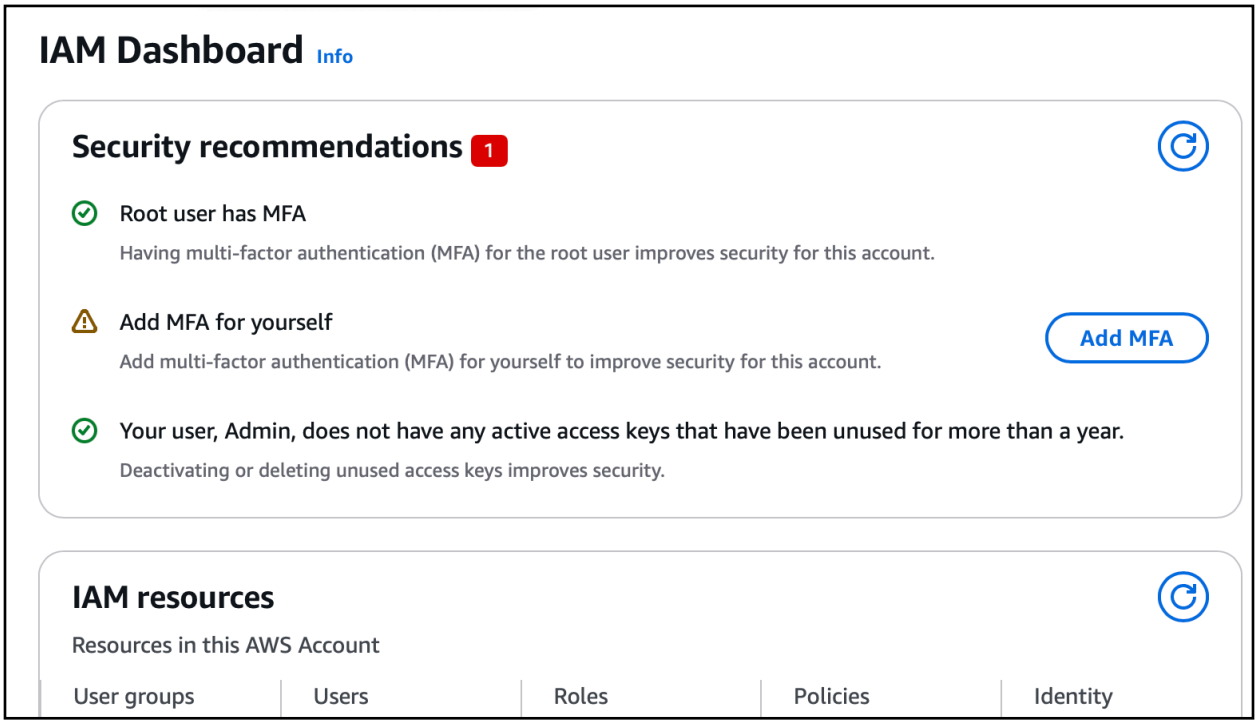


4. Follow the instructions (place fingerprint or use face unlock) > you have successfully setup MFA for root user.

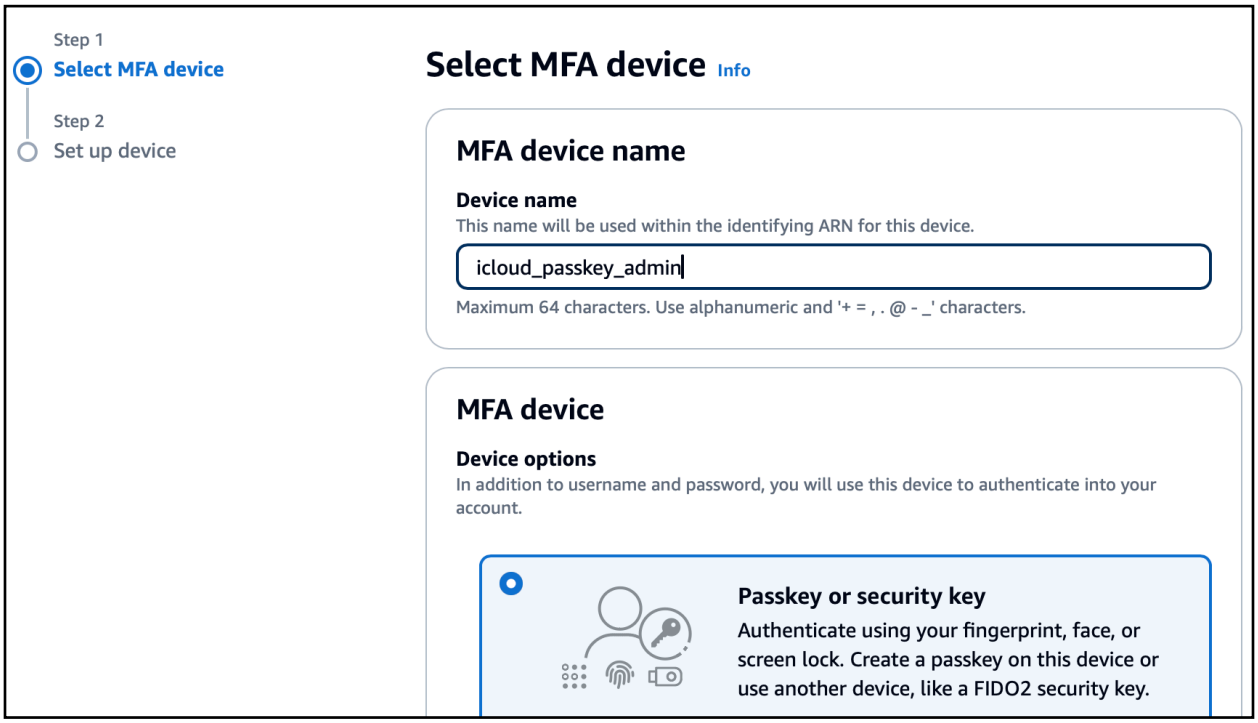


iii. Setup MFA for Admin Account

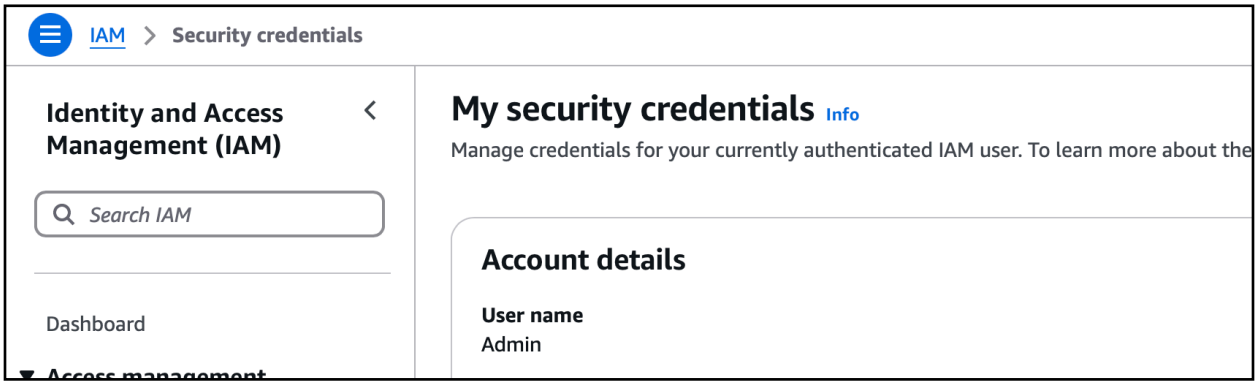
1. Similar to root user: IAM Dashboard > Security Recommendations > 'Add MFA'



2. Select MFA device > Device Name: type ‘my_passkey_admin’ > MFA device: select ‘Passkey or security key’ *(Note: ensure passkey is different from root user's)* > Next

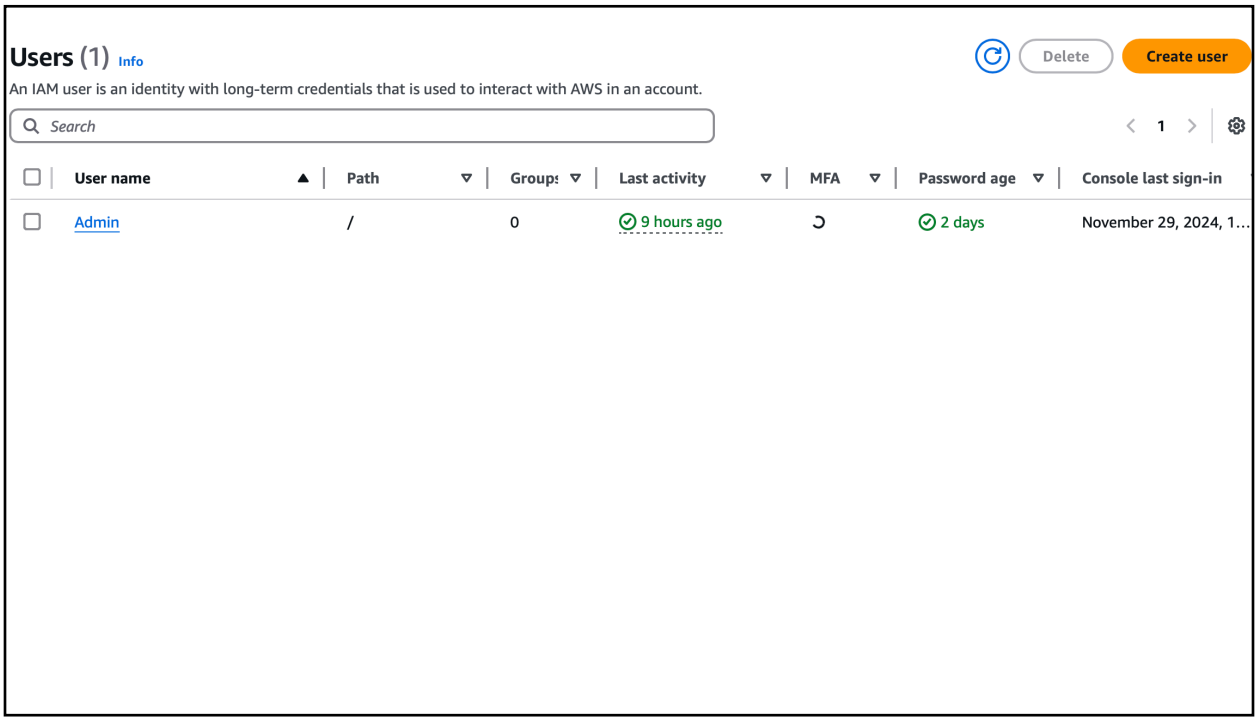


3. Follow the instructions (place fingerprint or use face unlock) > you have successfully setup MFA for admin user account.

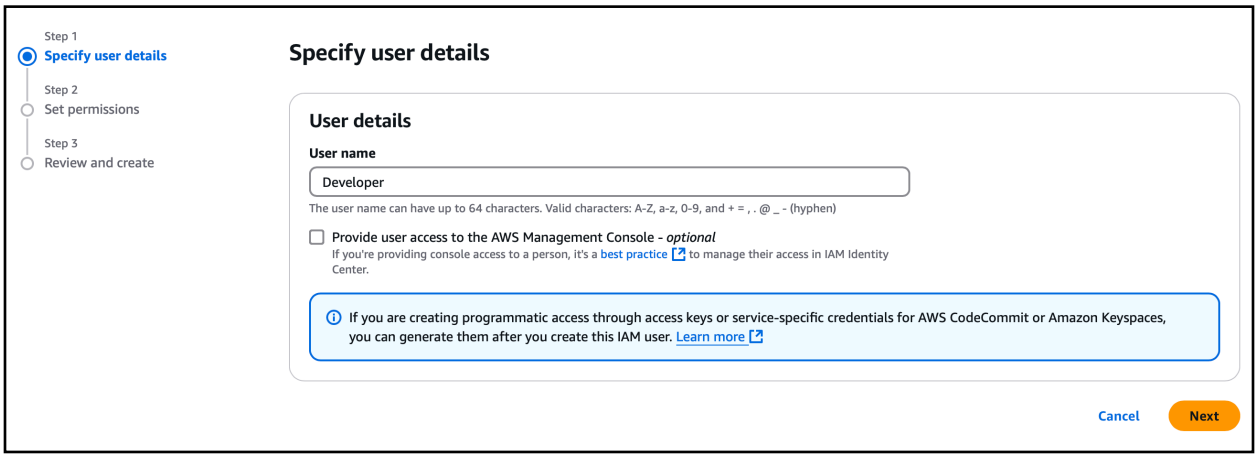


iv. Create ‘Developer’ User

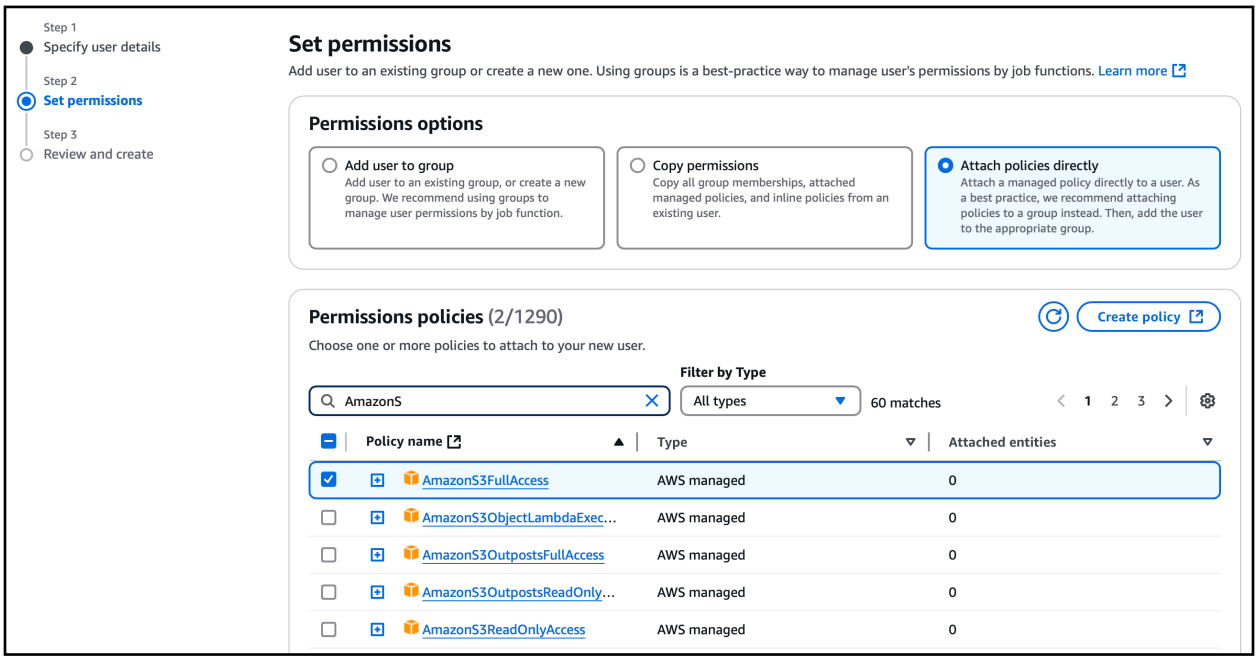
1. IAM > Access management > Users > Create user



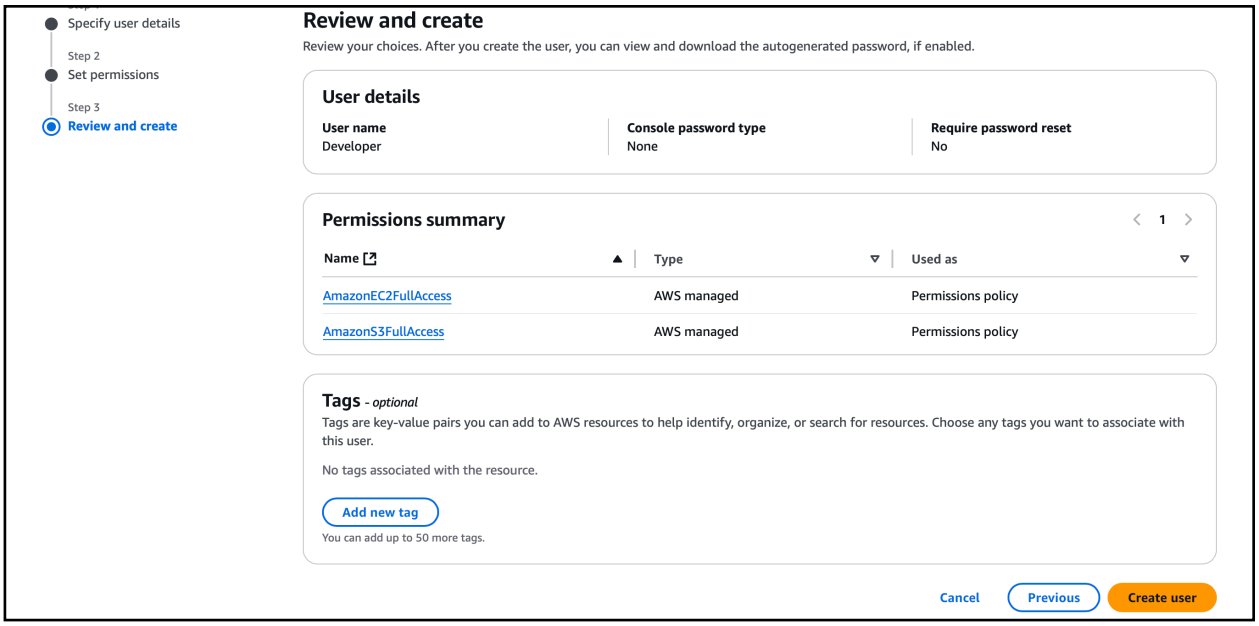
2. Users > Specify user details > User name: type ‘Developer' > Next



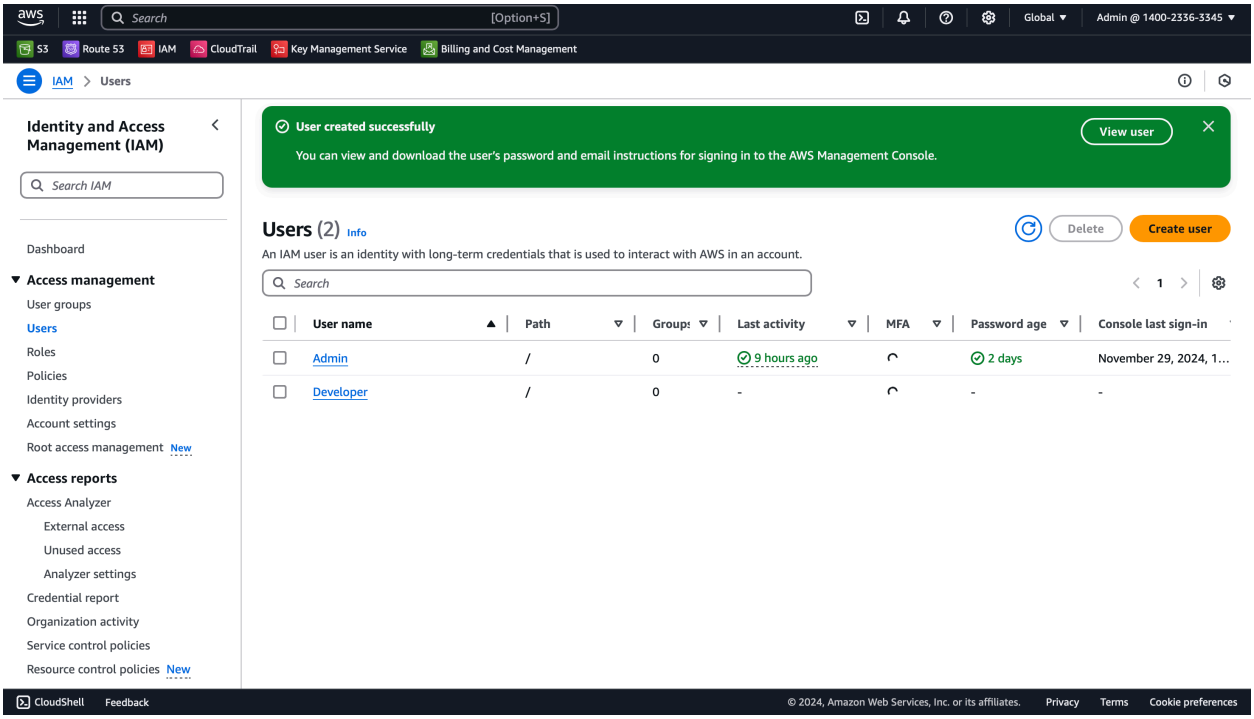
3. Set Permissions > Permission options > Attach policies directly > Permission Policies: select ‘AmazonS3FullAccess’ and ‘AmazonEC2FullAccess’ > Next



4. Review and Create > Verify permissions > Create User

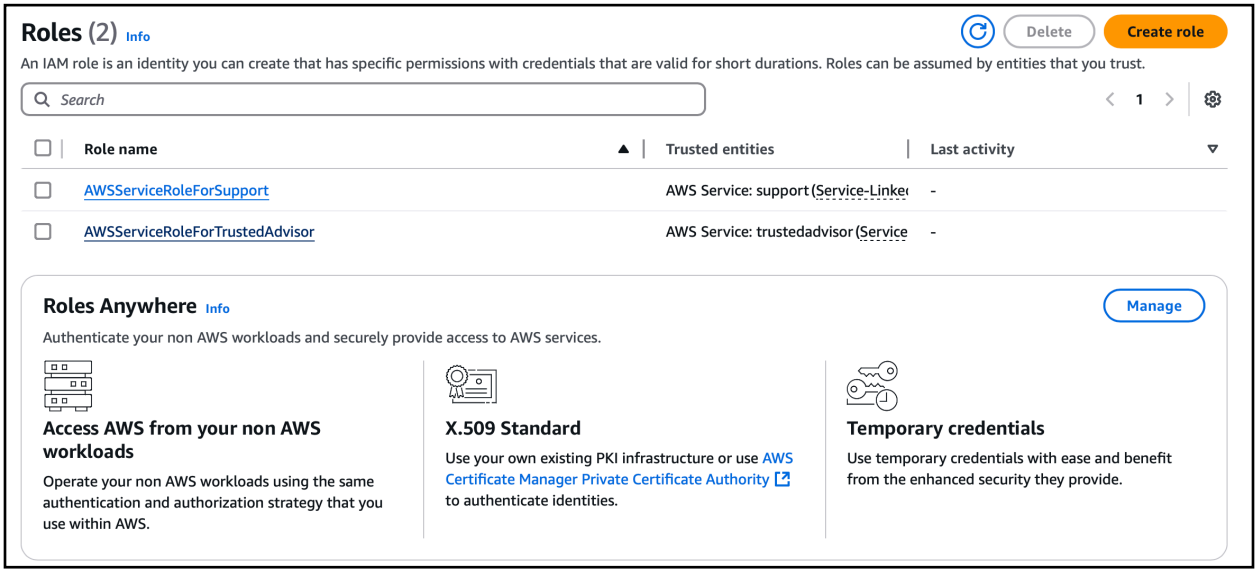


5. Successful creation of Developer user.

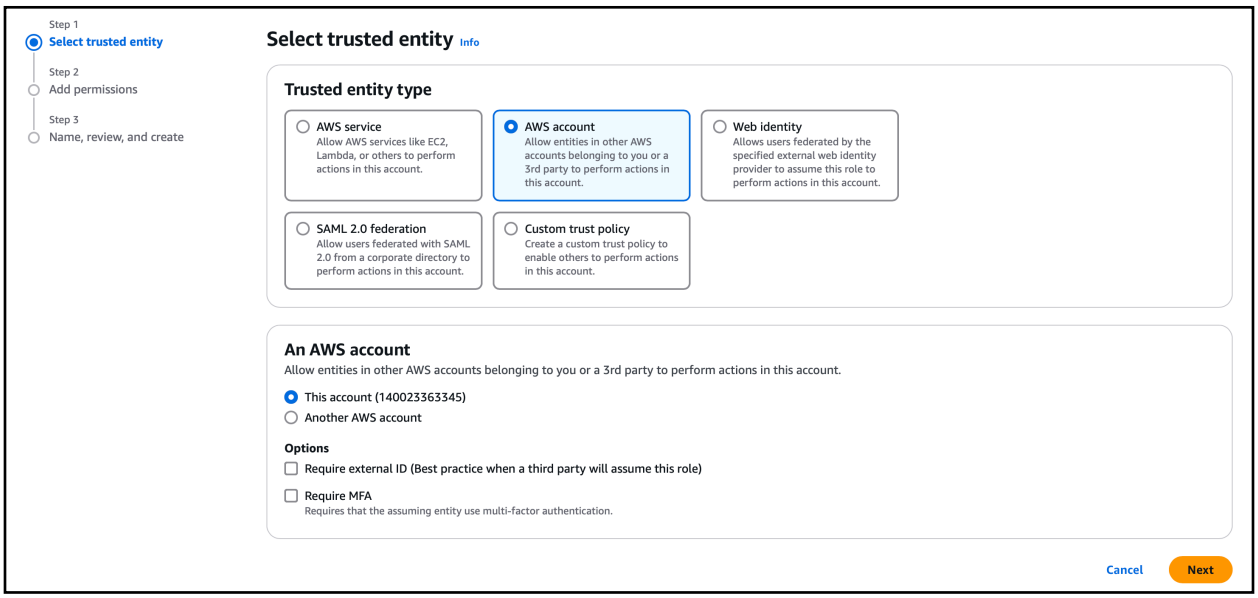


v. Create 'Auditor' Role

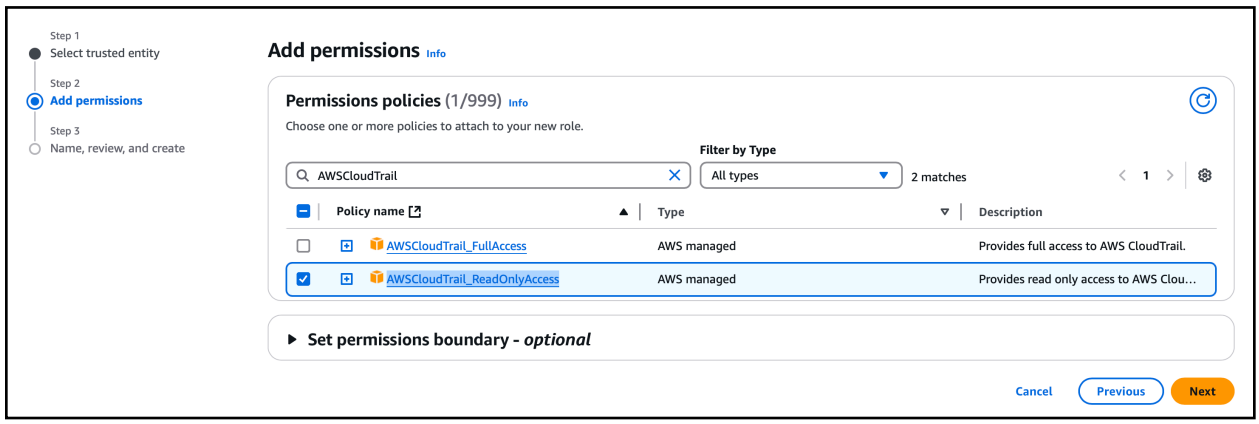
1. IAM > Access management > Users > Create role



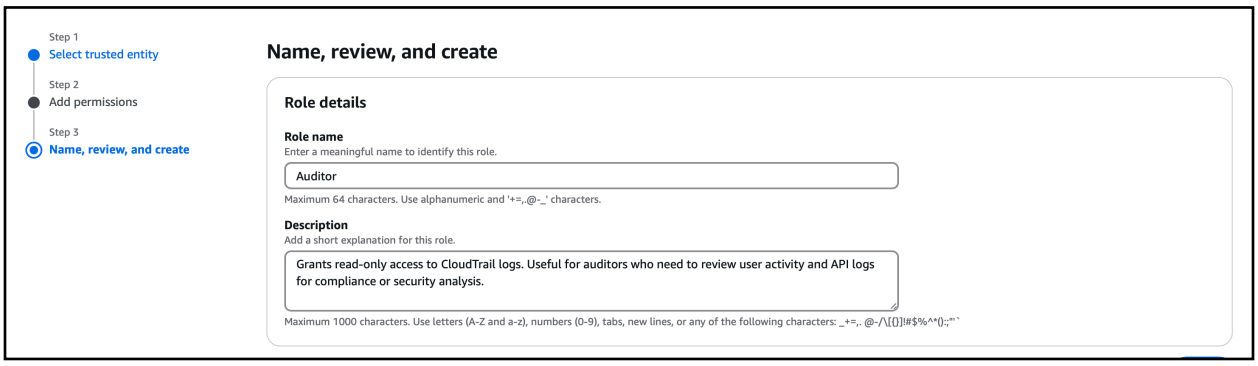
2. Select trusted entity > Trusted entity type: select ‘AWS account’ > An AWS Account > ‘This account’ > Next



3. Add Permissions > Permission policies > ‘AWSCloudTrail_ReadOnlyAccess’

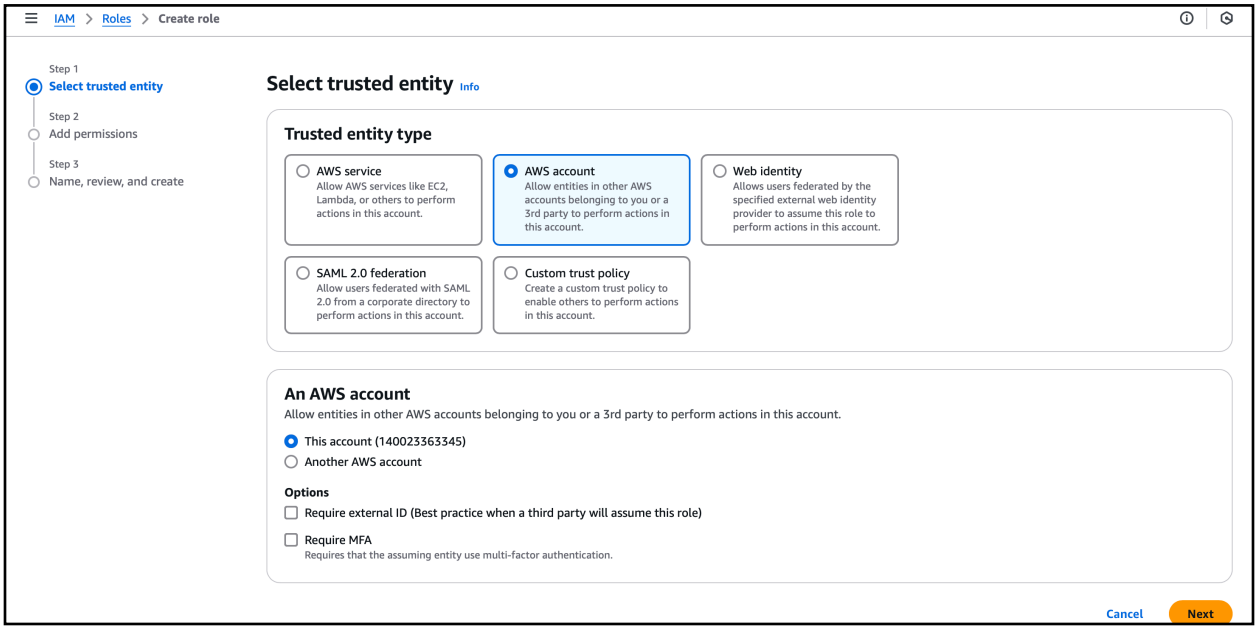


4. Role name > ‘Auditor’ > Description: type ‘Grants read-only access to CloudTrail logs. Useful for auditors who need to review user activity and API logs for compliance or security analysis.’ > Create role

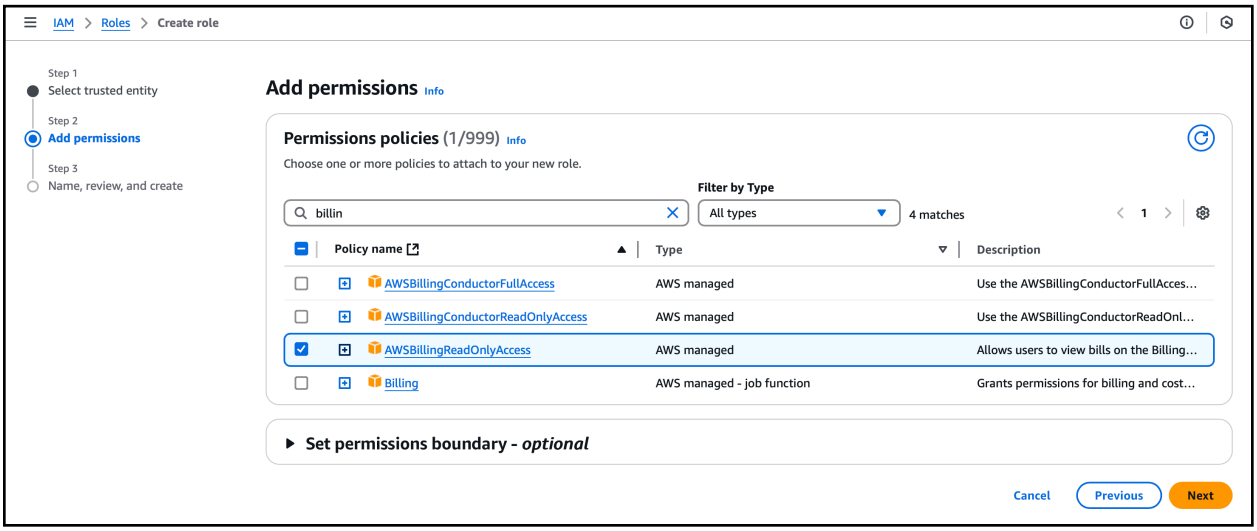


vi. Create 'ReadOnlyBilling' Role

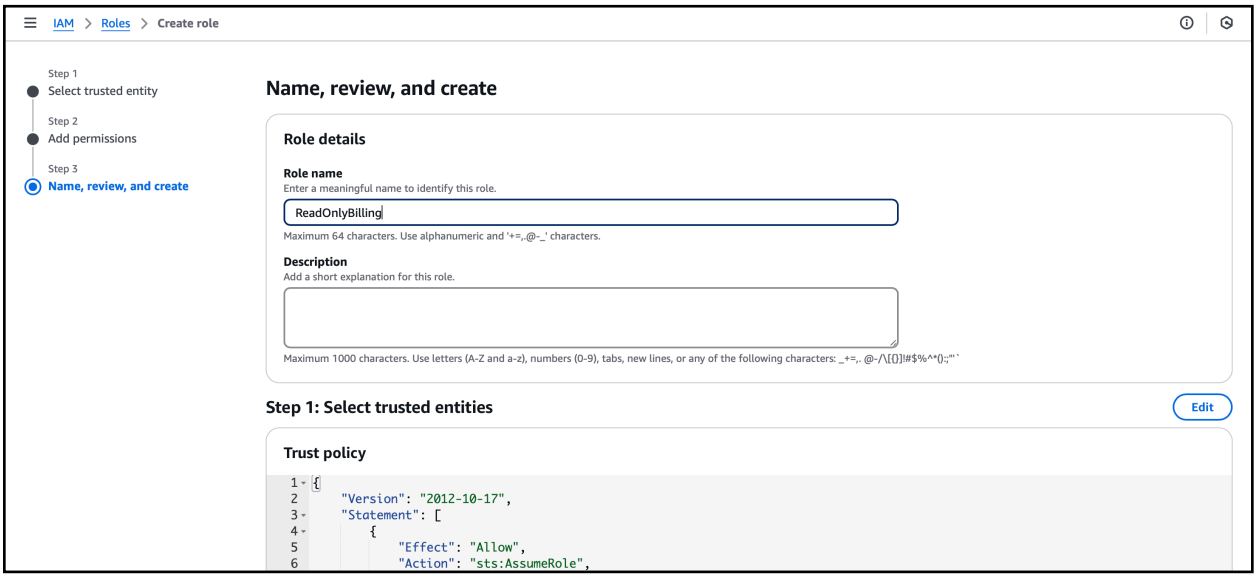
1. IAM > Access management > Users > Create role
2. Select trusted entity > Trusted entity type > 'AWS account' > An AWS Account > 'This account' > Next



3. Add Permissions > Permission policies: select 'AWSBillingReadOnlyAccess'



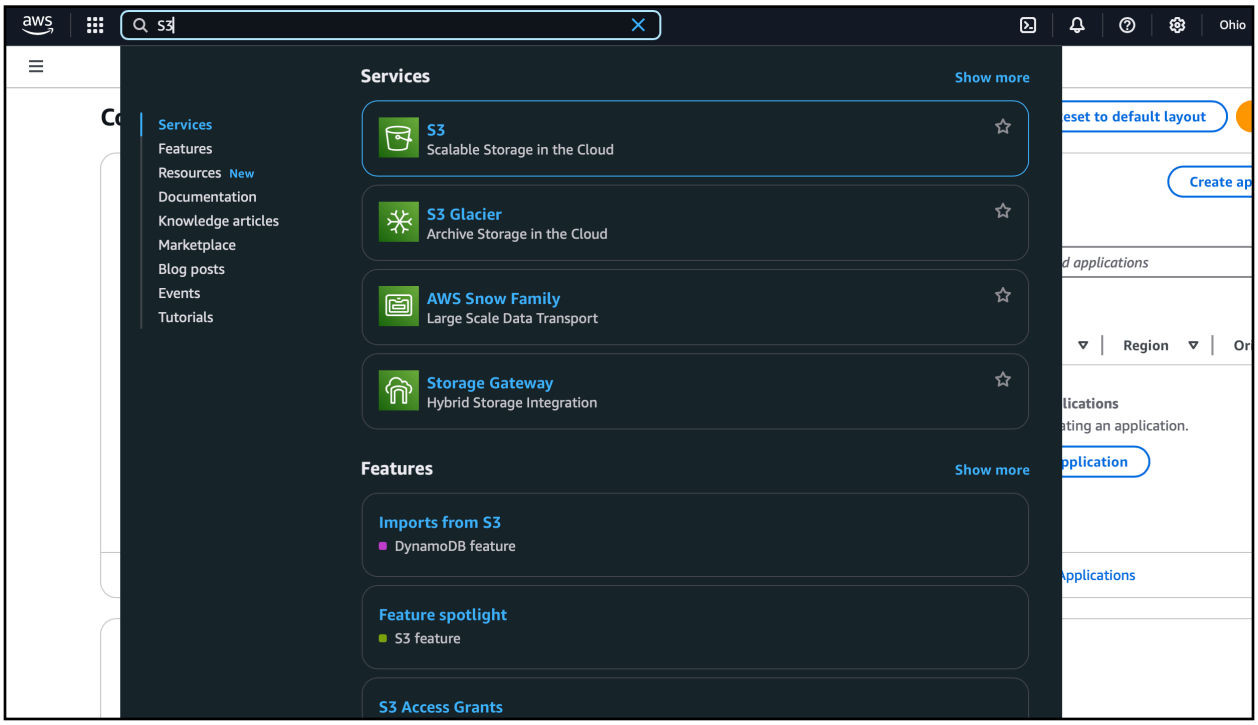
4. Role details > Role name: type ‘ReadOnlyBilling’ > Description: type ‘Grants read-only access to CloudTrail logs. Useful for auditors who need to review user activity and API logs for compliance or security analysis.’ > Create role



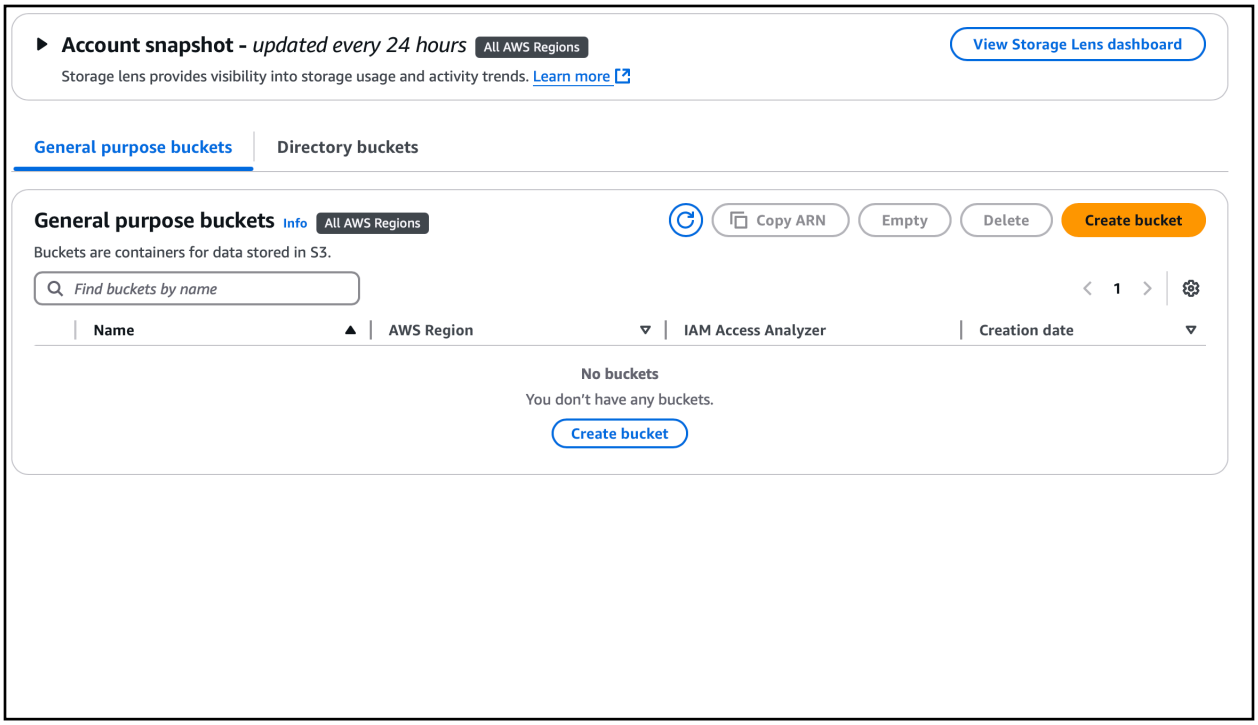
B. Amazon S3 (Simple Storage Service)

i. Host the Website for Public Access

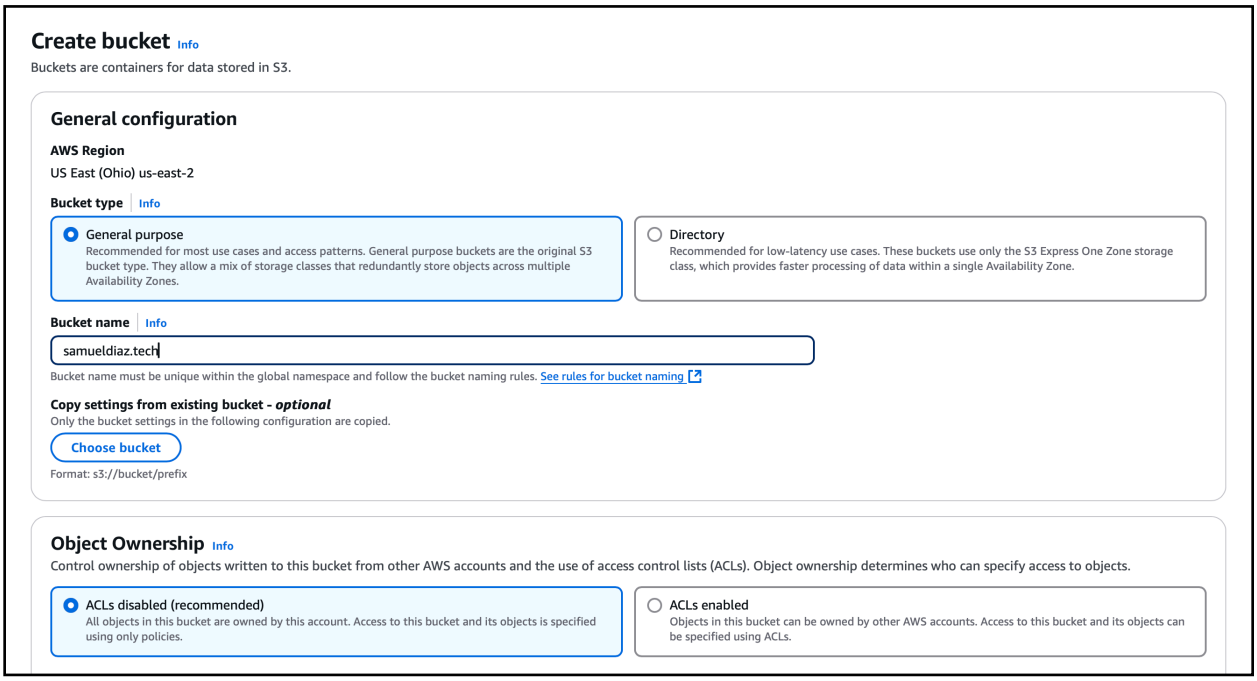
1. AWS Homepage > Search bar: type and select ‘S3’



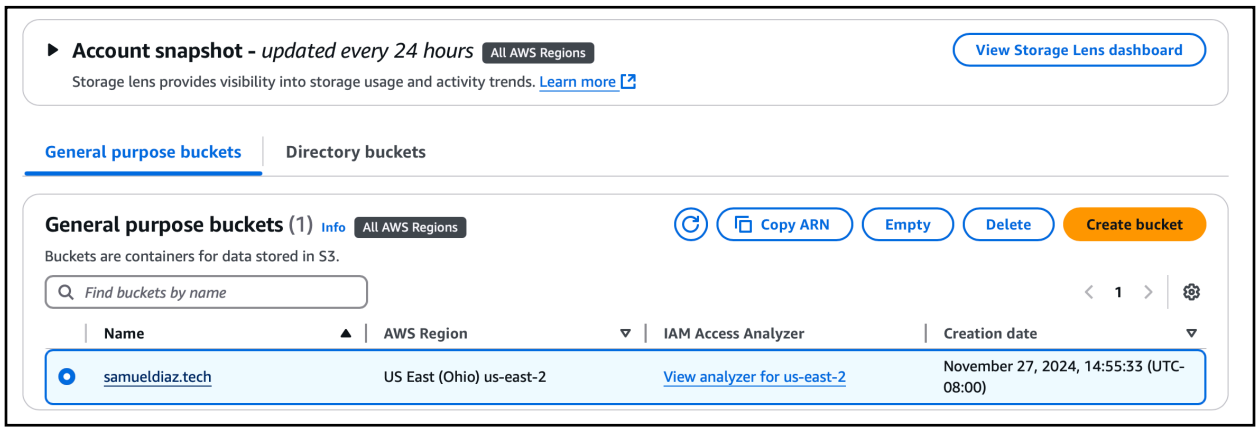
2. Amazon S3 > Buckets > ‘Create Bucket’



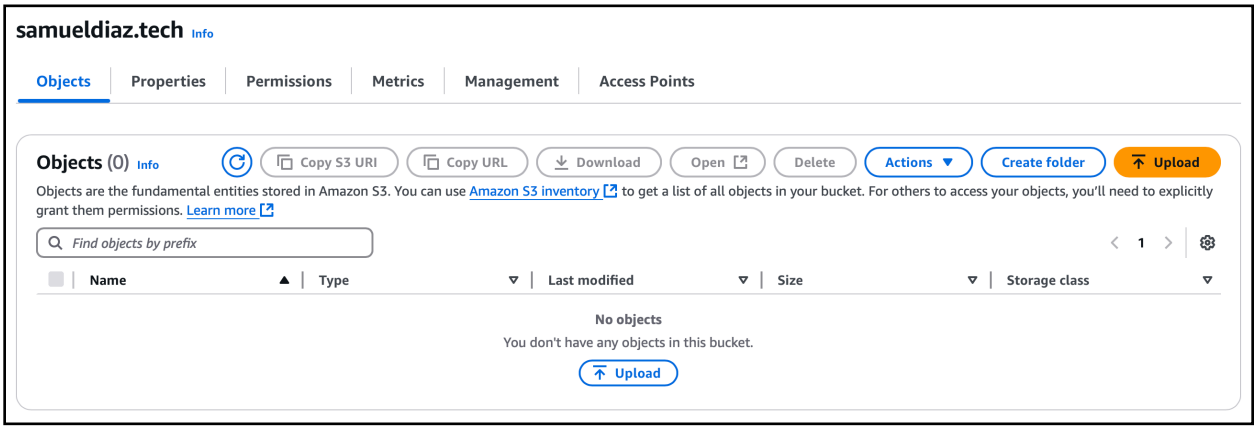
3. Region > Select closest region > Bucket name: ‘samueldiaz.tech' > scroll down > uncheck ‘Block Public Access settings for this bucket’ > acknowledge the warning > Click submit



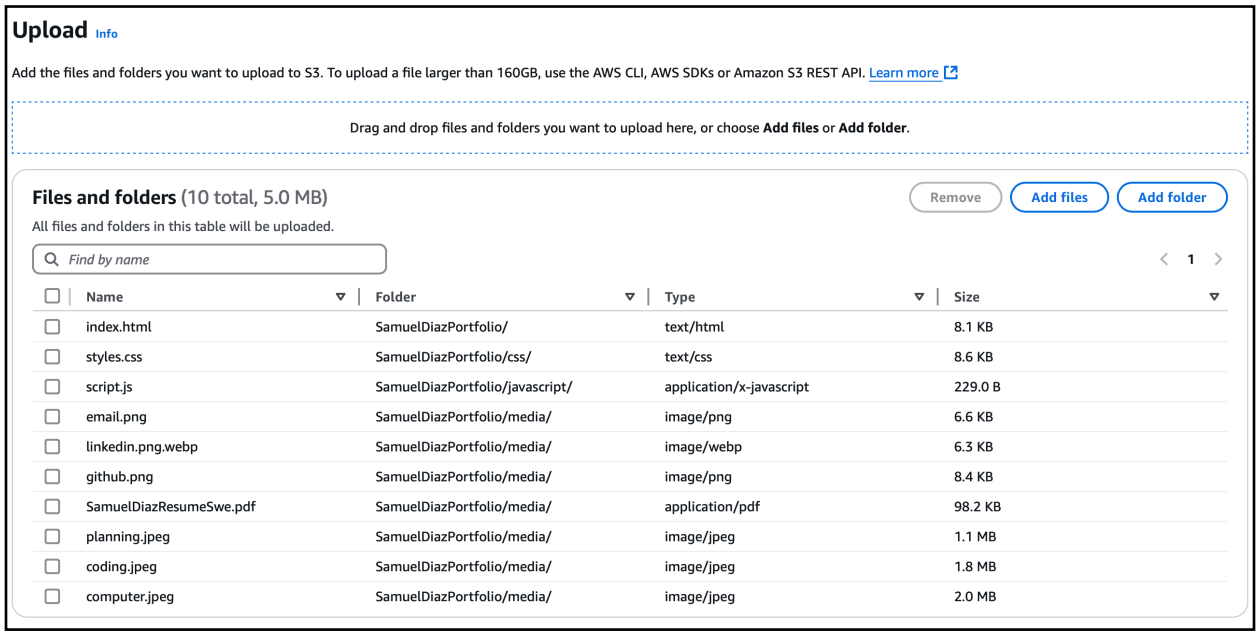
4. Amazon S3 > Buckets > ‘samueldiaz.tech'



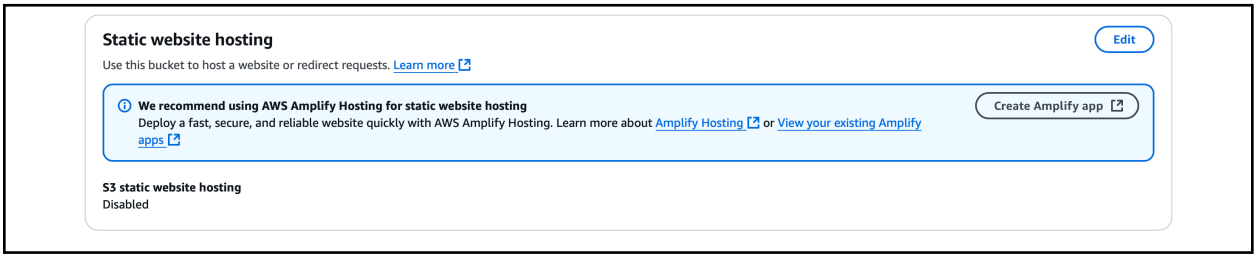
5. Objects > ‘Upload’



6. ‘Add folder’ > *Locate and select folder with web content* > ‘Upload’. The website data has been uploaded.

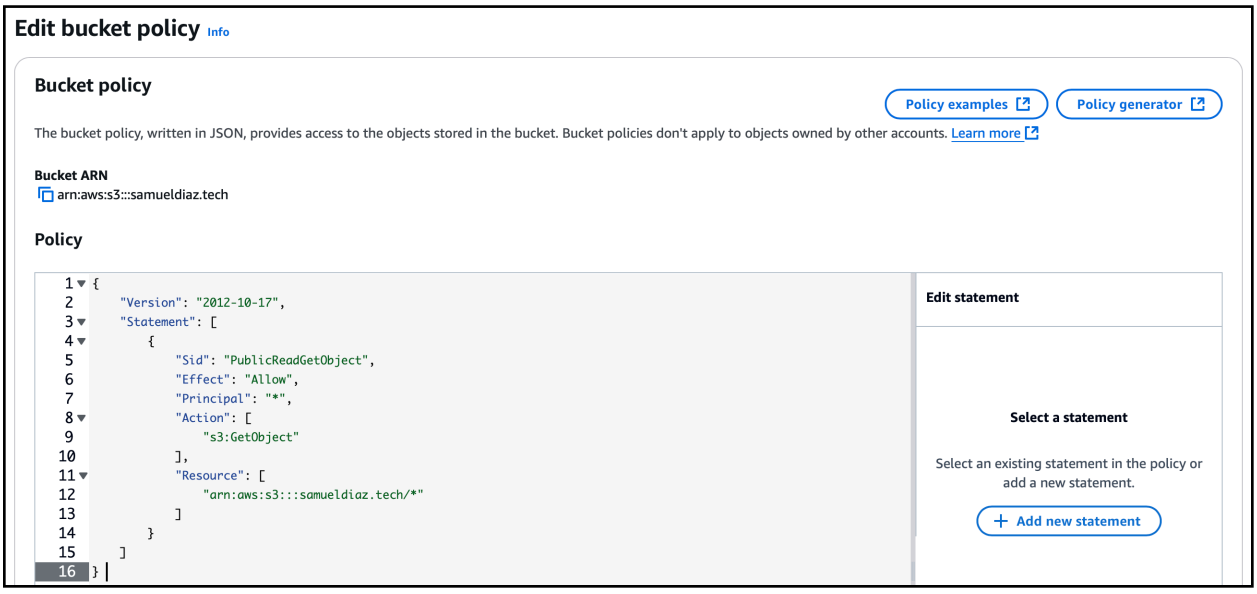


7. Amazon S3 > Buckets > 'samueldiaz.tech' > ‘Properties.’ > Static website hosting > ‘Edit’



8. Edit static website hosting > Static website hosting: select ‘Enable’ > Index Document: type ‘index.html’ > Save Changes

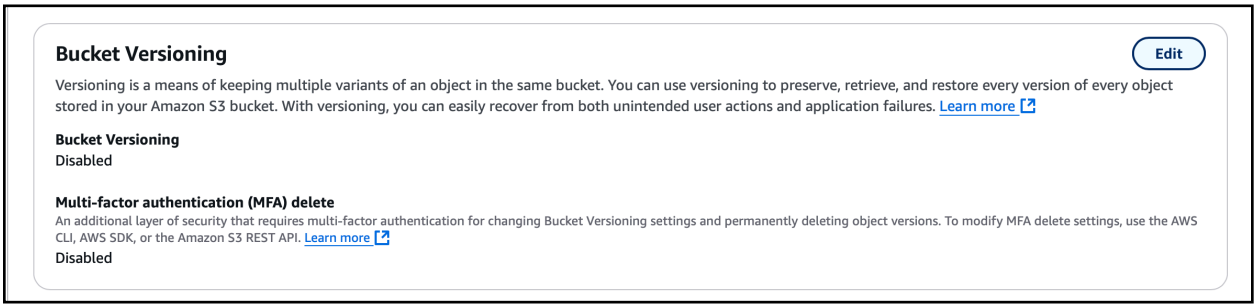
9. Buckets > 'samueldiaz.tech' > Permissions > Edit Bucket Policy > attach Policy (code below) > Save Changes



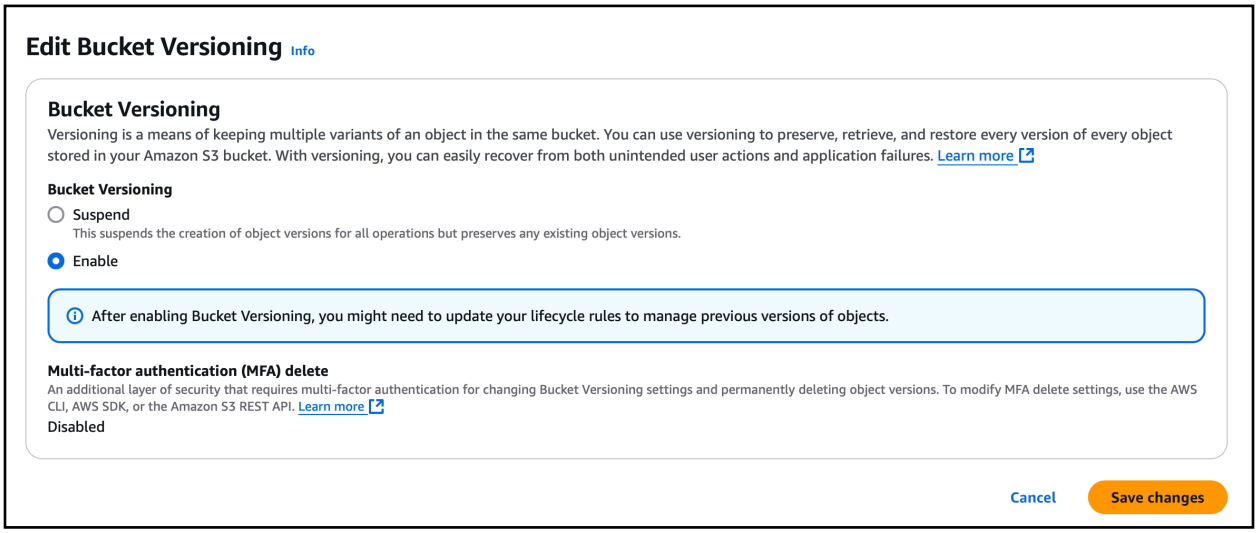
ii. Enable Bucket Versioning

Reasoning: Versioning has been activated for buckets to ensure that any modifications made are retrievable. Therefore, if modifications or updates cause issues, you are able backtrack to the state your website was previously on.

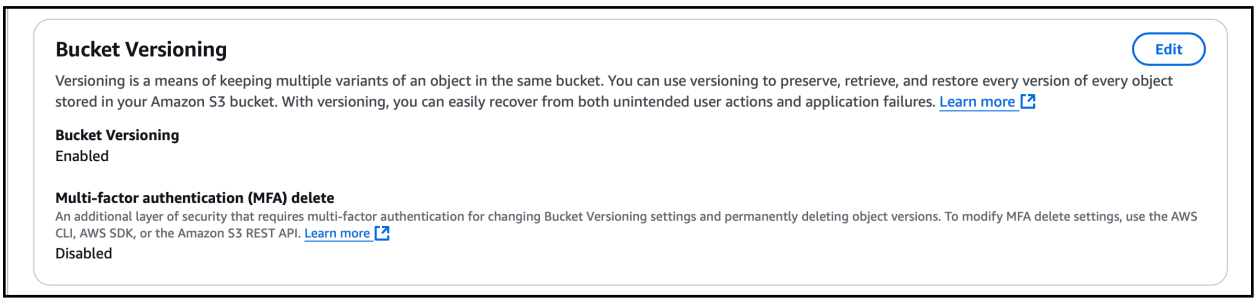
1. Amazon S3 > Buckets > 'samueldiaz.tech' > Properties > Bucket Versioning > Edit



2. Edit Bucket Versioning > Bucket Versioning > Enable > Save Changes

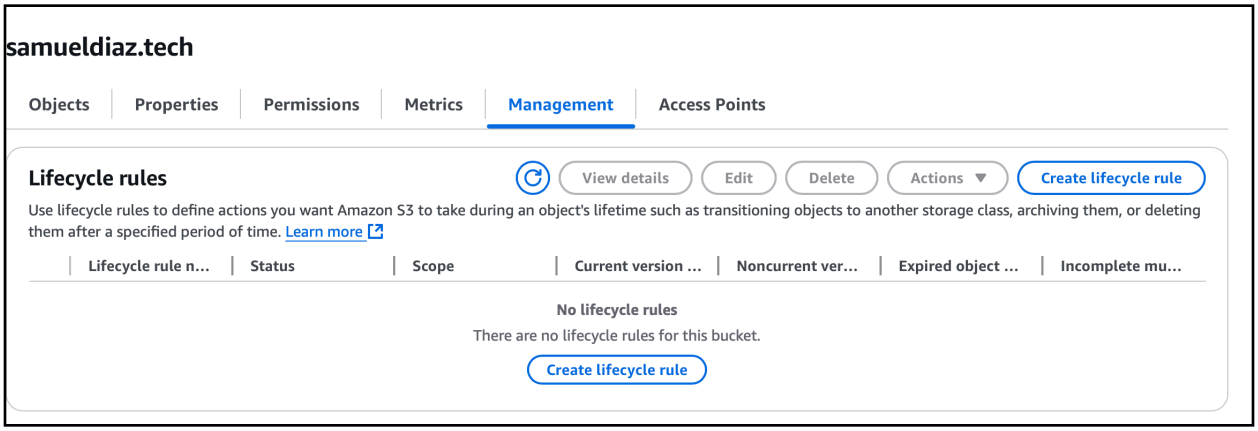


3. Bucket Versioning has now been enabled

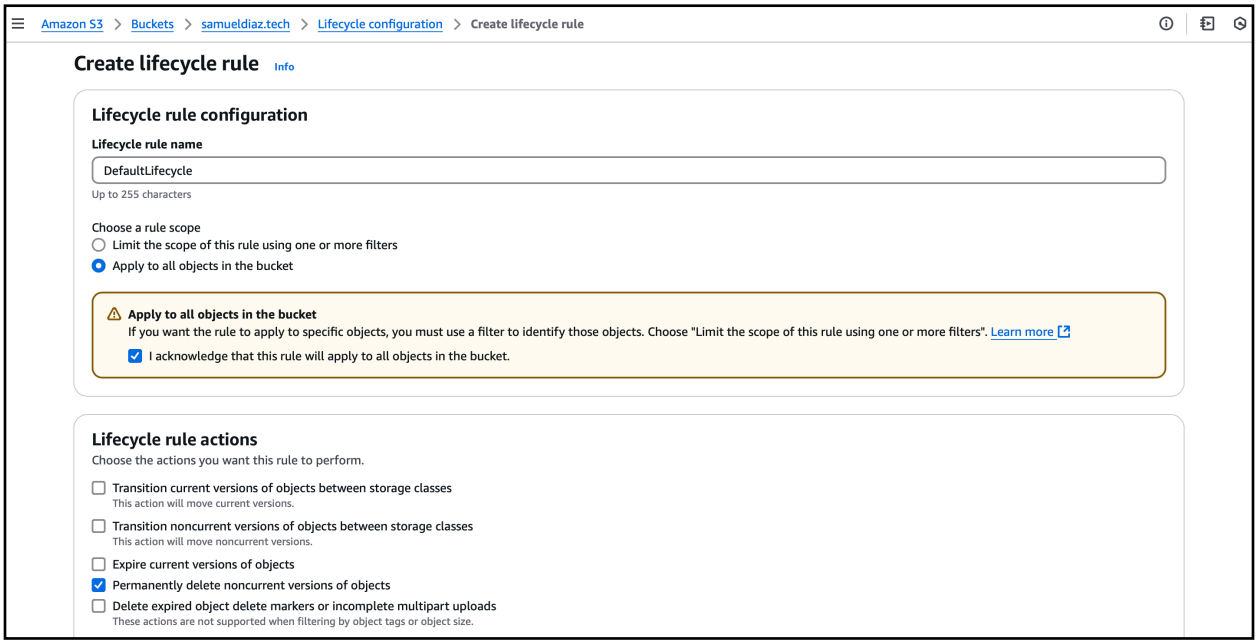


iii. Enable Lifecycle Policies

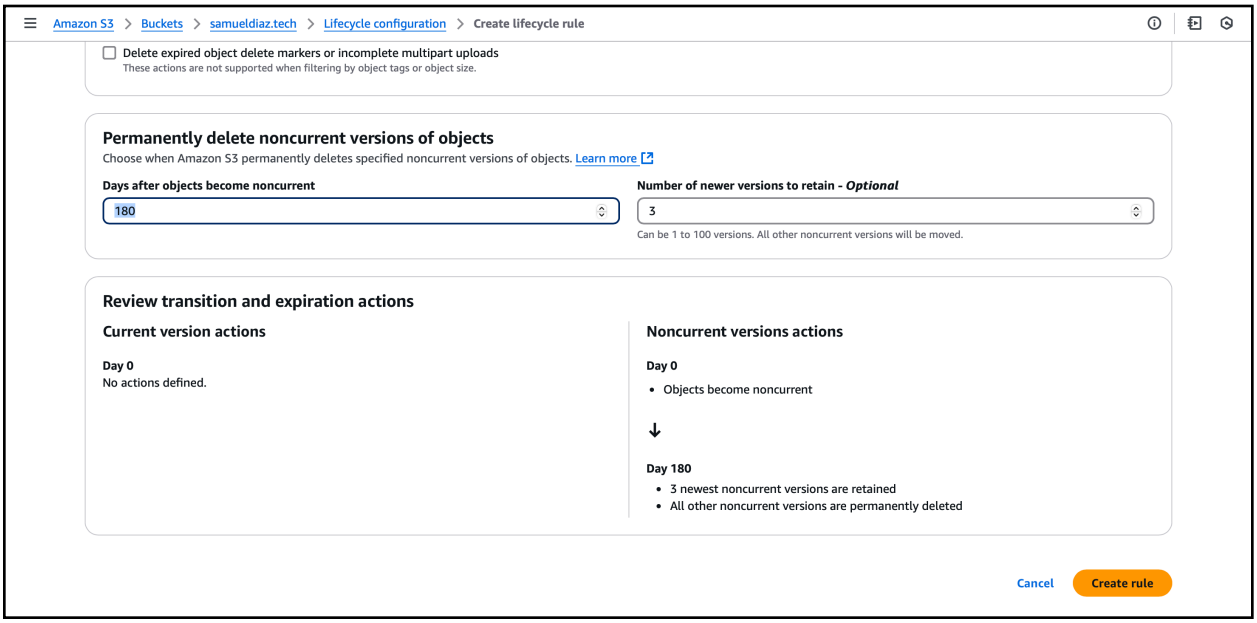
1. S3 > Buckets > 'samueldiaz.tech' > Management > Lifecycle rules > Create lifecycle rule



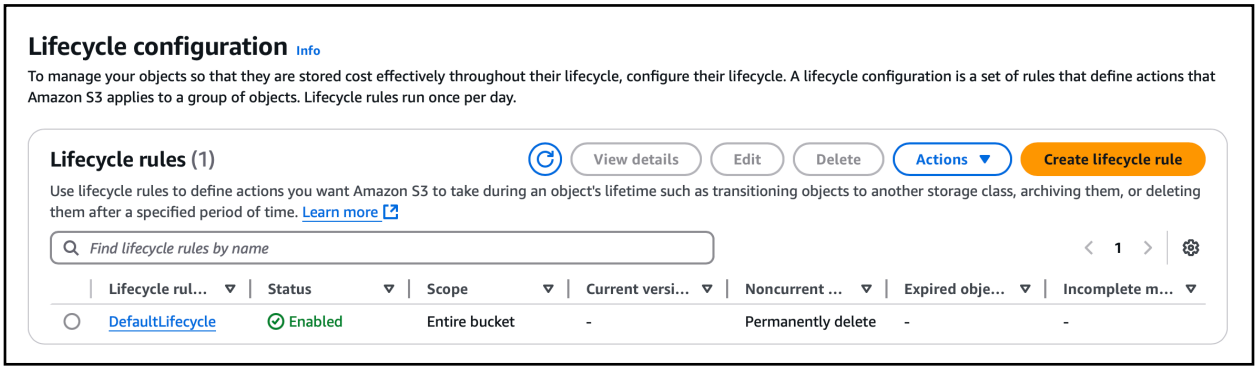
2. Create lifecycle rule > Lifecycle rule name: type 'DefaultLifecycle' > Chose a rule scope: select 'Apply to all objects in the bucket' > check 'I acknowledge that this rule…' > Lifecycle rule actions: select 'Permanently delete concurrent versions of objects'



3. Permanently delete non-current versions of objects > Days after objects become concurrent: type ‘180’ > Number of newer versions to retain - Optional: type ‘3’ > Create rule

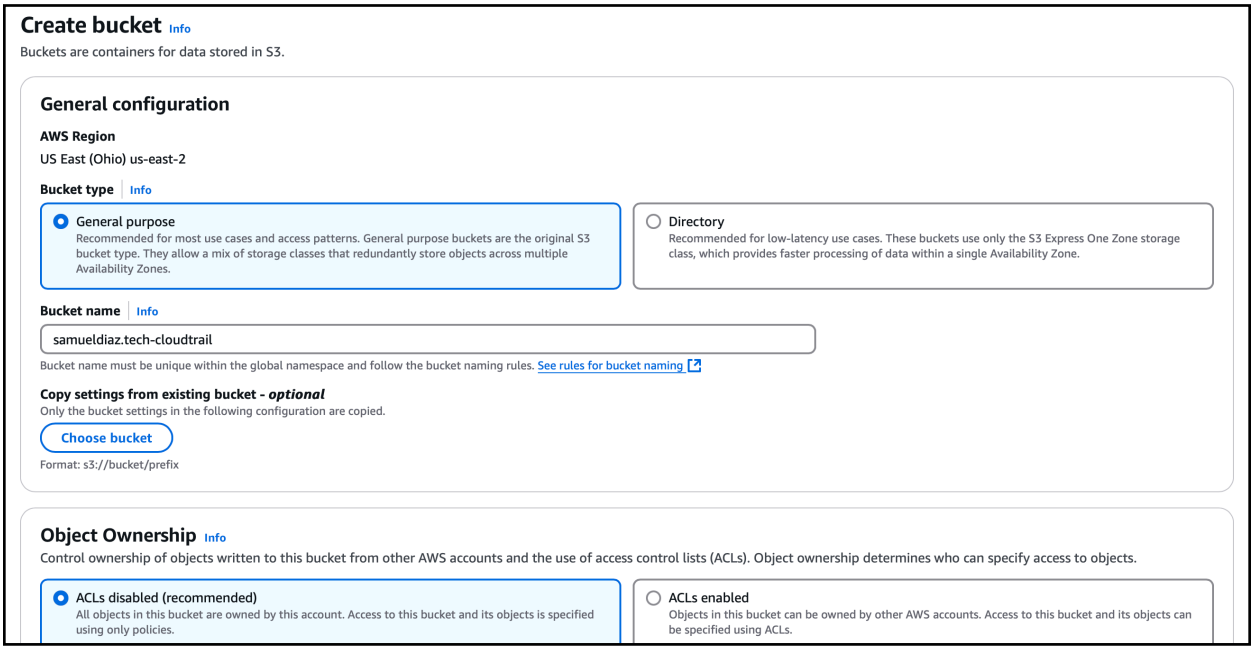


4. Lifecycle rule has been created permanently deleting obsolete versions after 6 months. The top 3 most current versions would be kept, no matter the time elapsed

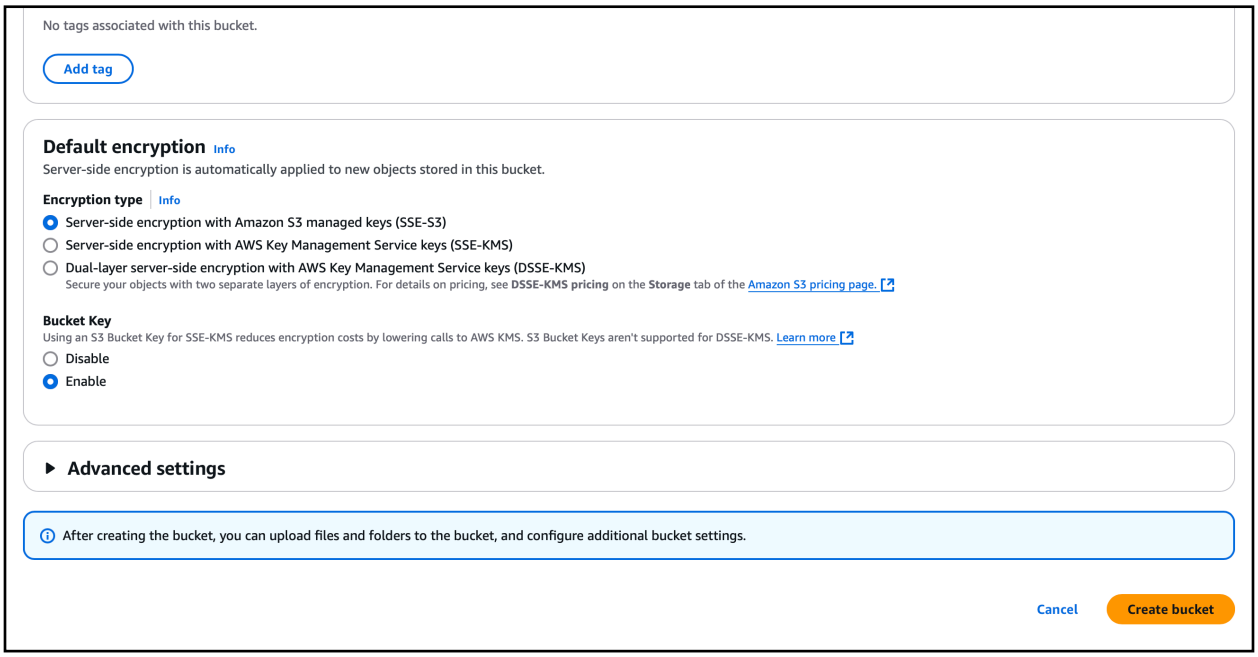


iv. Create Encrypted Bucket to Store CloudTrail Logs

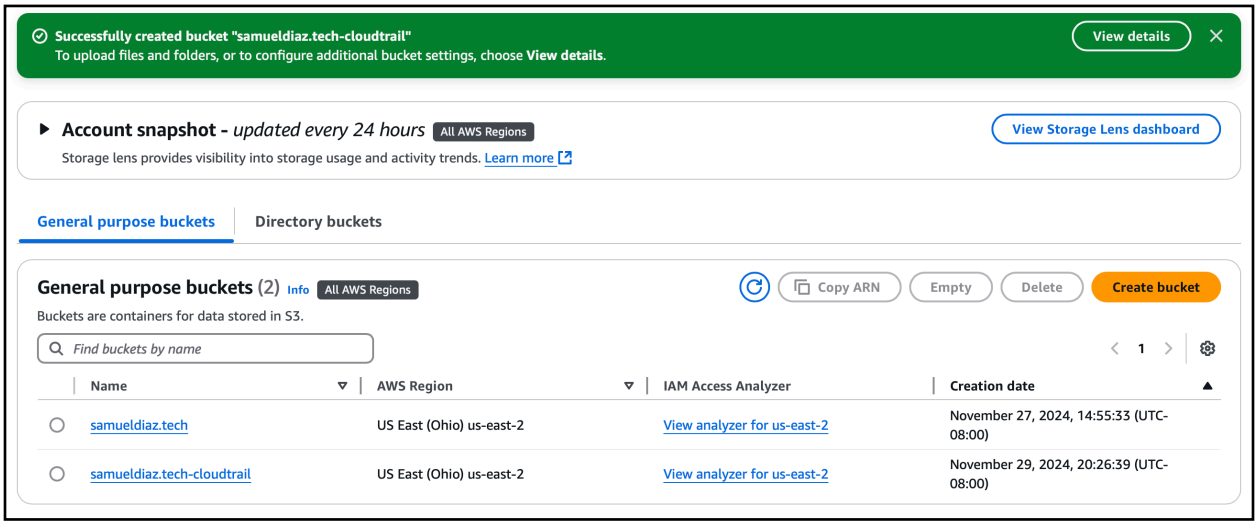
1. Amazon S3 > Buckets > Create bucket > General configuration > Bucket name: type ‘samueldiaz.tech-cloudtrail’



2. Block Public Access settings for this bucket: check ‘Block All Public Access’ > Default Encryption > Encryption Type: select ‘Server-side encryption with Amazon S3 managed keys (SSE-S3)’ > Create bucket



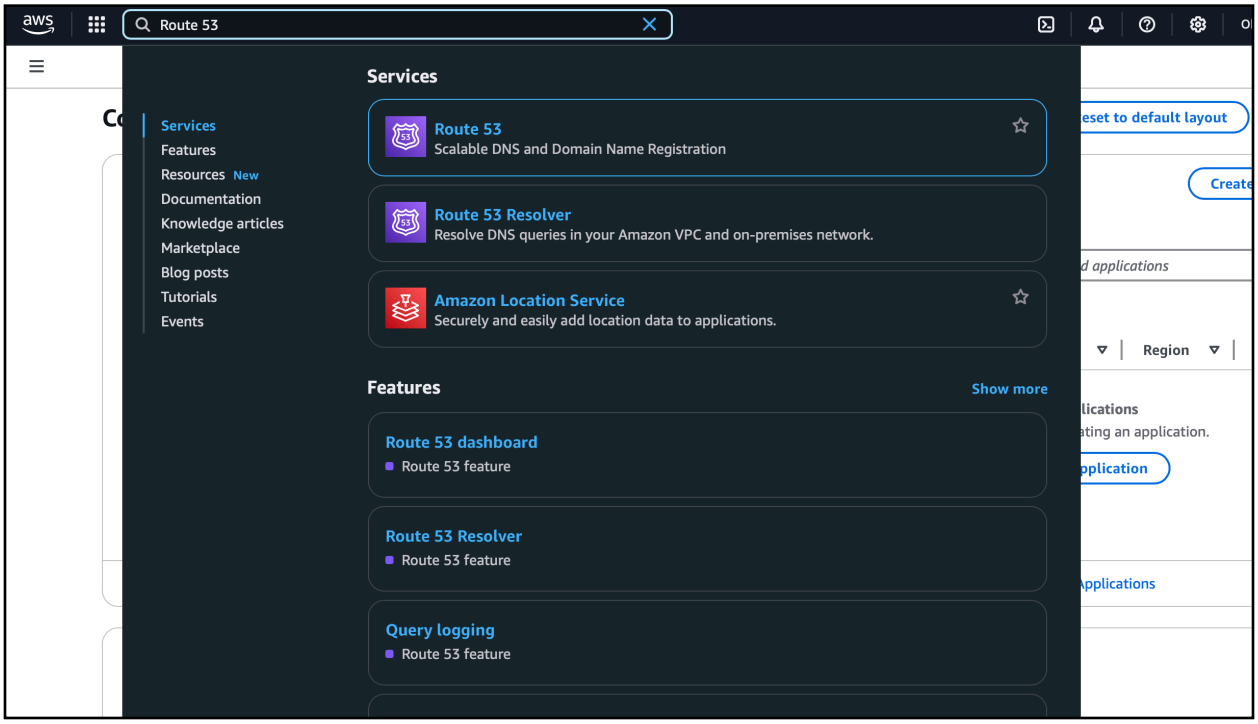
3. Verify creation of S3 bucket



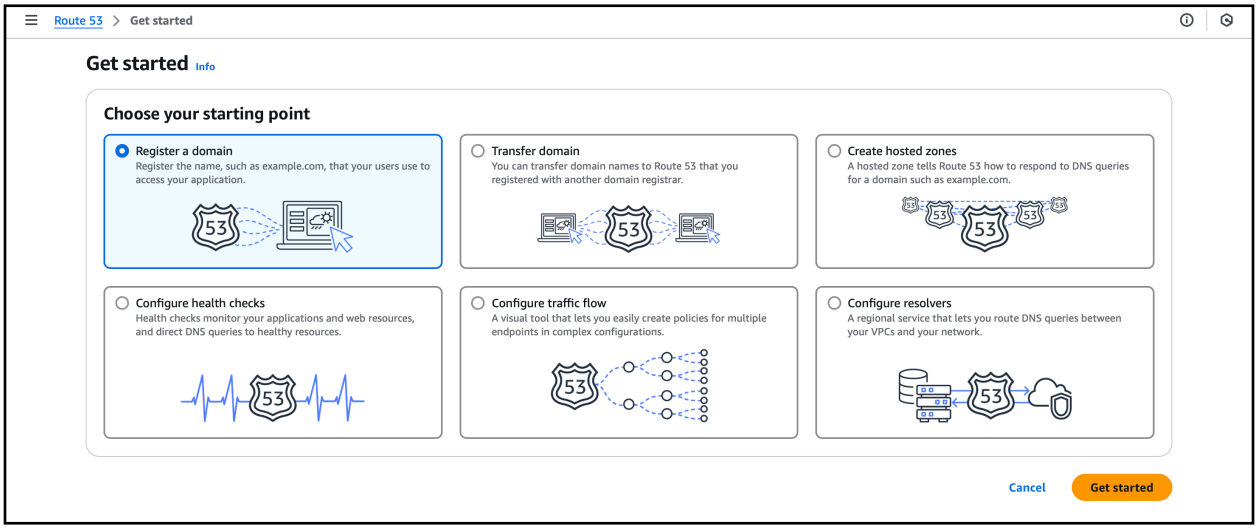
C. Route 53

i. Register Domain Name

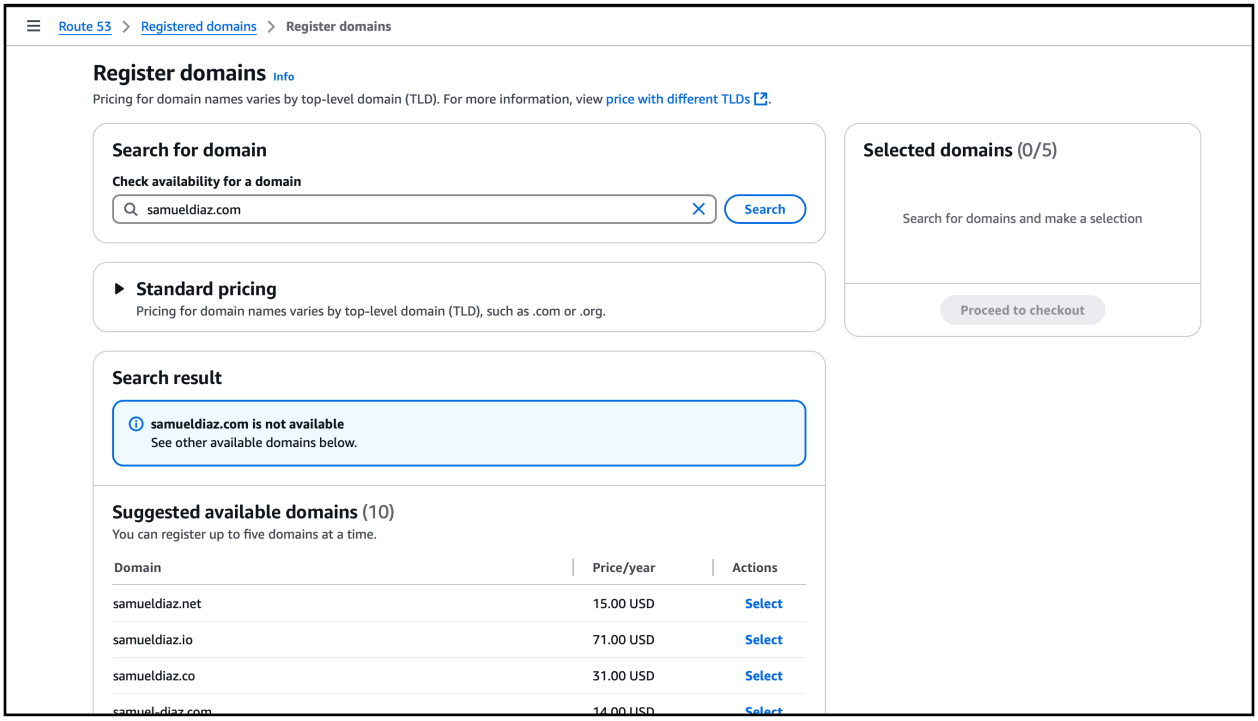
1. AWS Homepage > Search bar: type and select ‘Route 53’



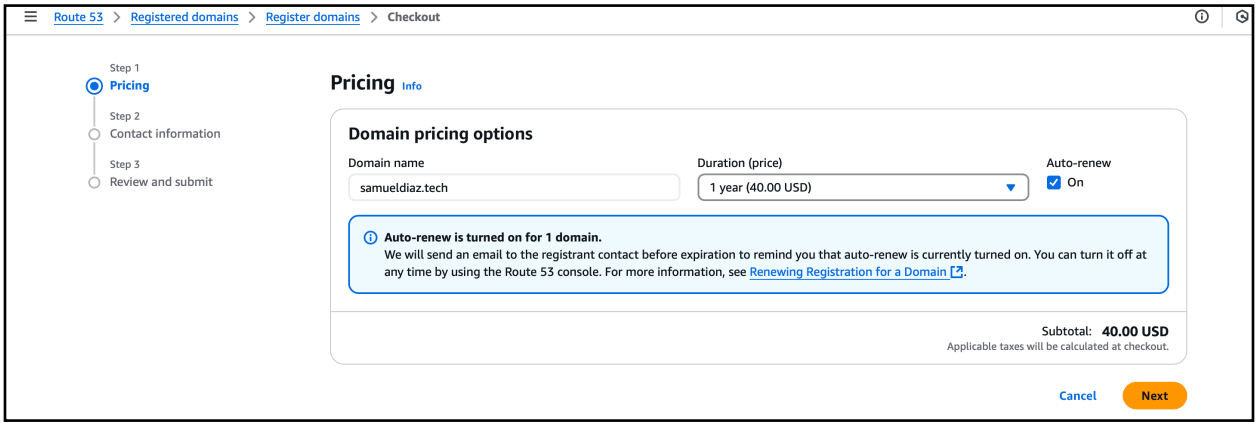
2. Get started > select Register a domain > click Get started



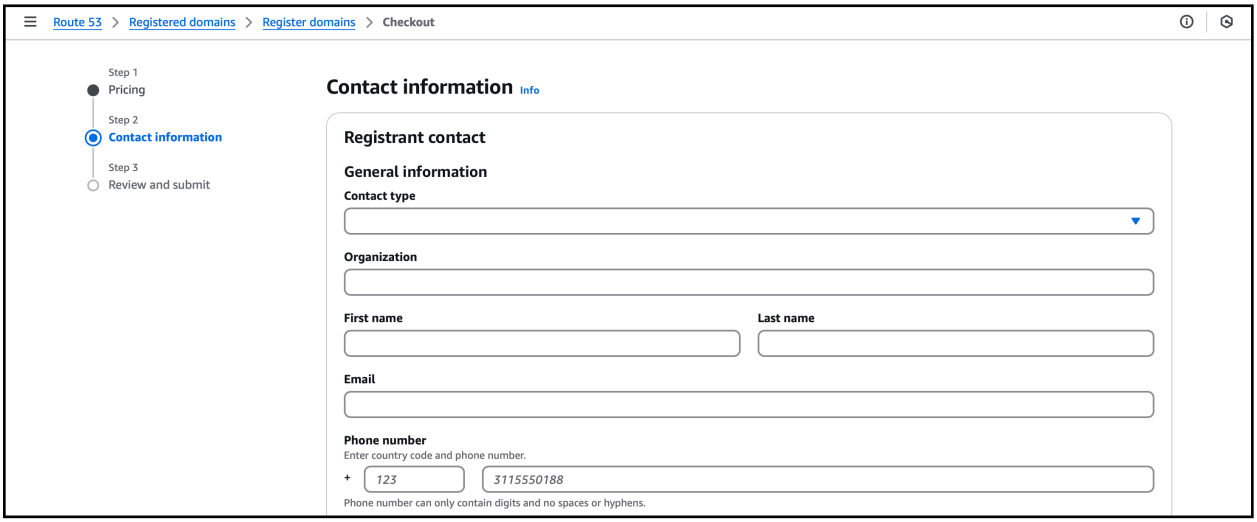
3. Register domain > Search for domain: type and select ‘samueldiaz.tech' > Proceed to checkout



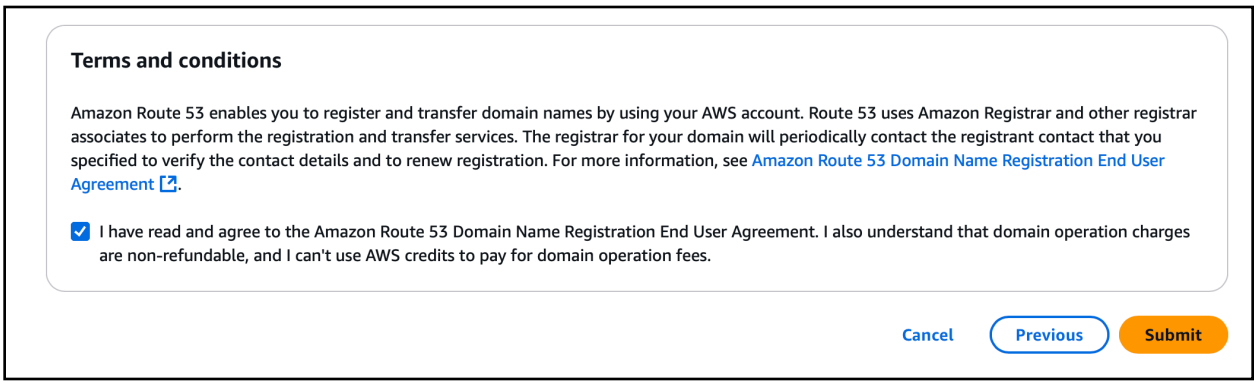
4. *Verify pricing* > Next



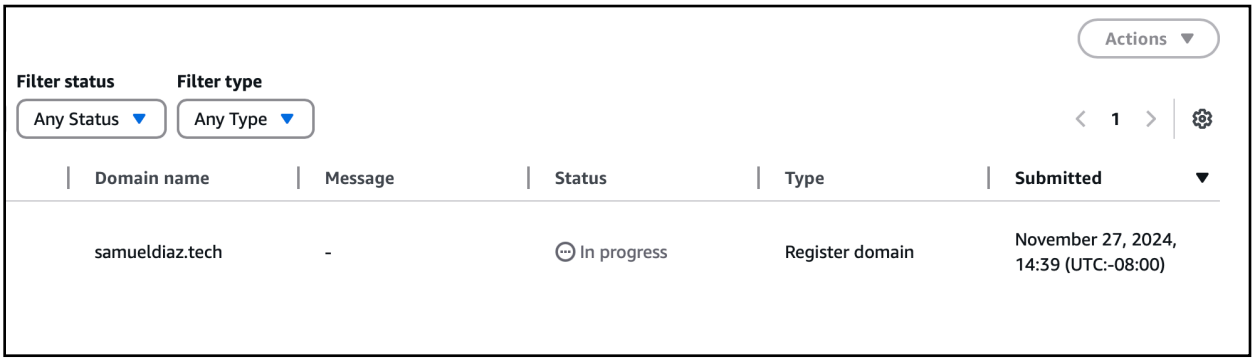
5. Verify contact information > Click Next



6. Terms and conditions > check 'I have read and agree…' > Click Submit

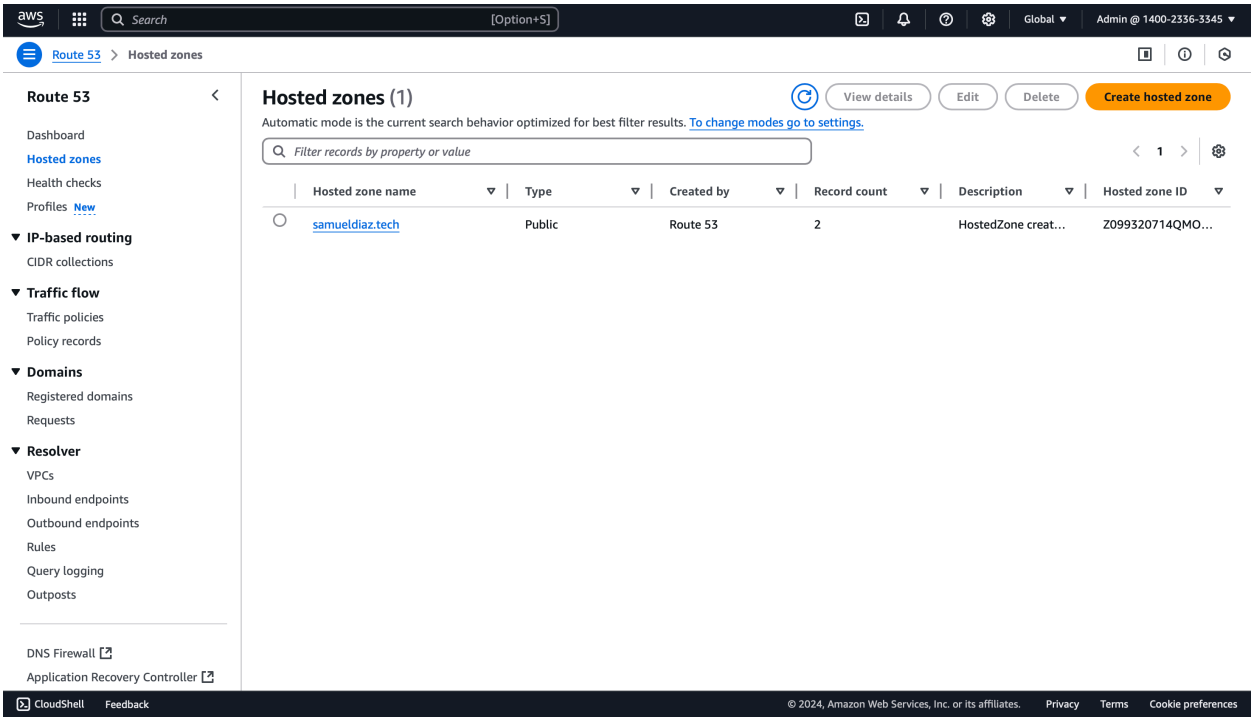


7. Your domain name is registered and ready to use.

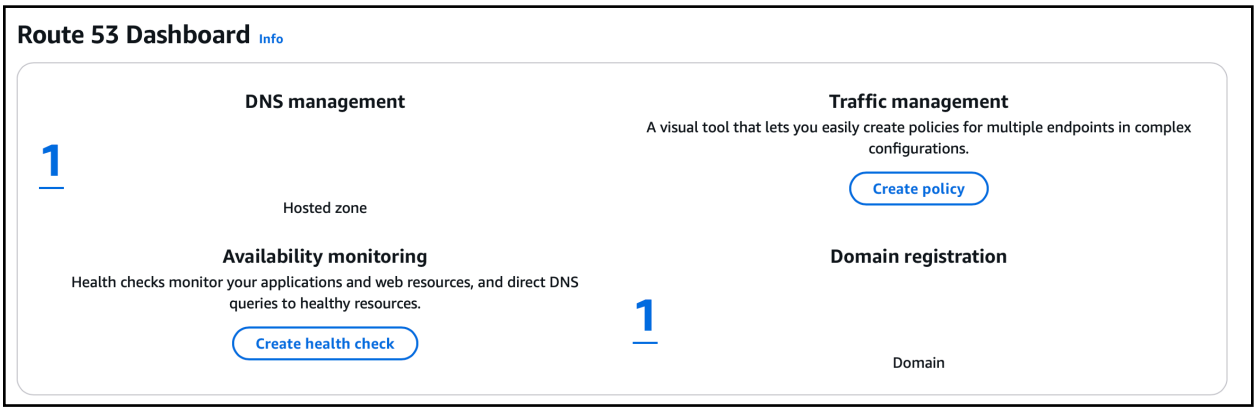


ii. Redirect custom URL (samueldiaz.tech) to S3 bucket

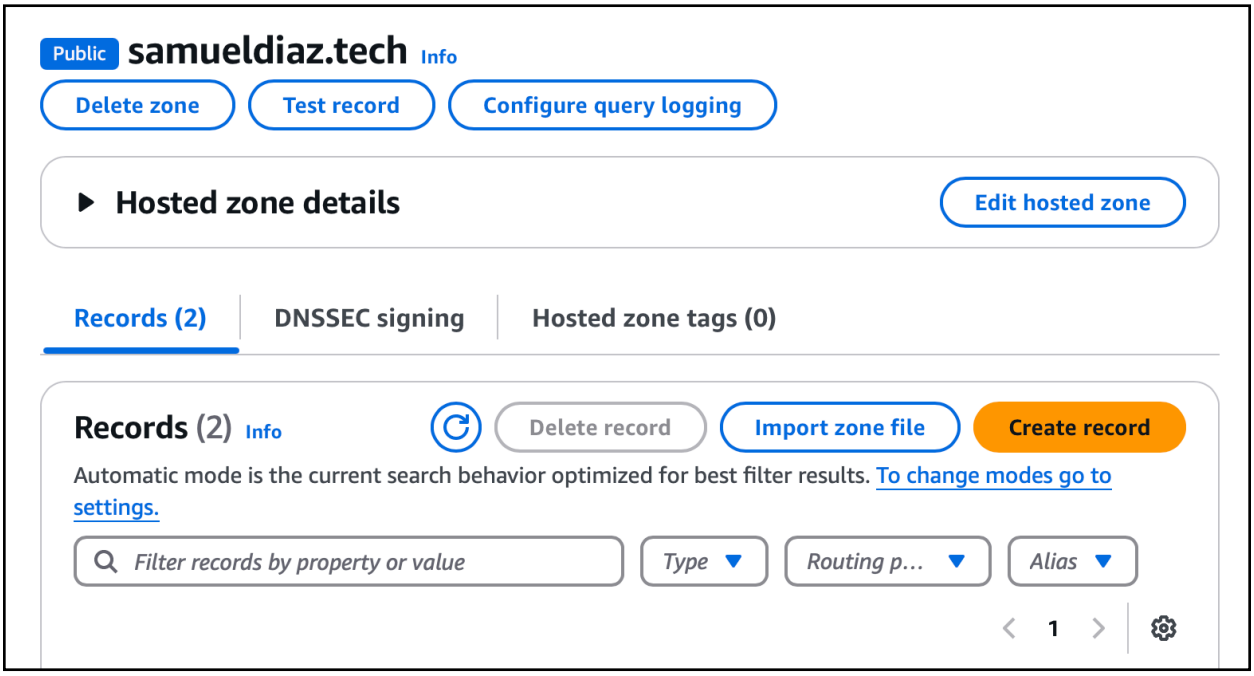
1.Route 53 > DNS Management > Hosted Zone



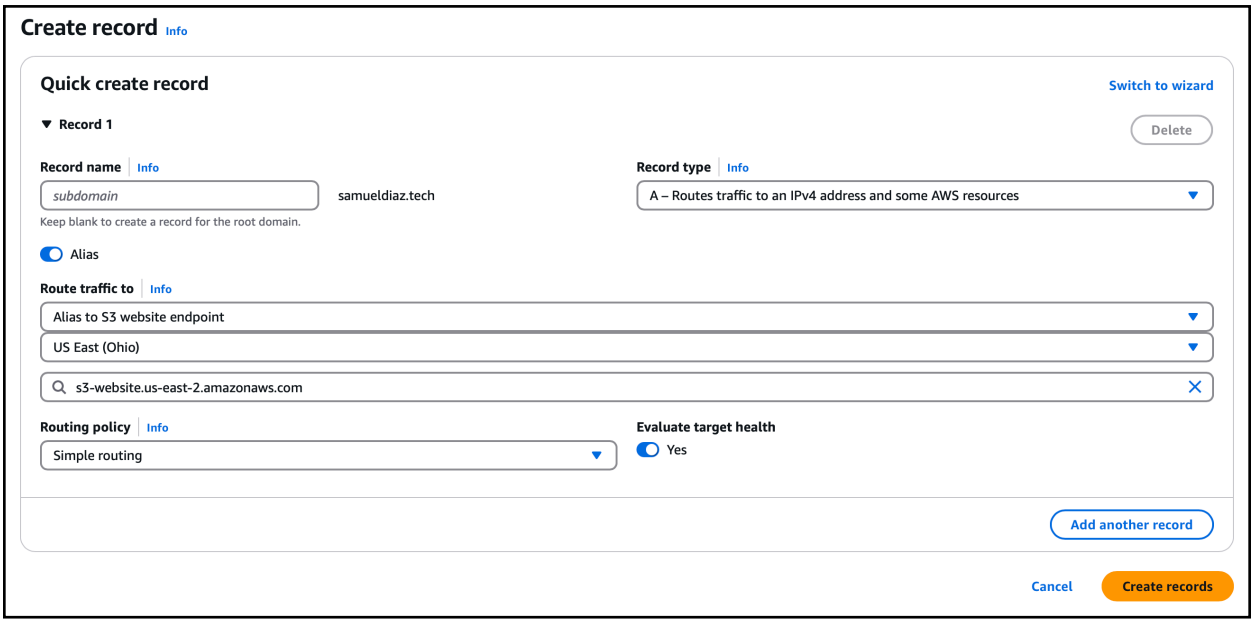
2. Hosted zones > select 'samueldiaz.tech'



3. Records > Create record



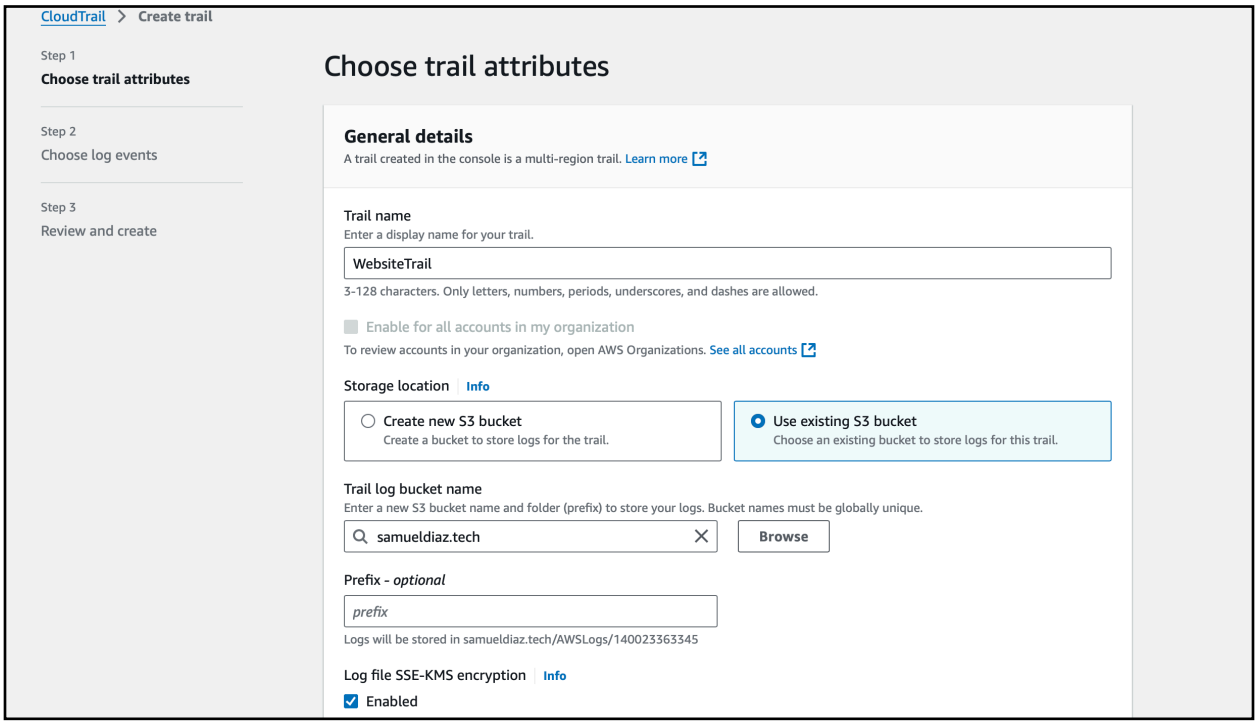
4. Enable Alias > Route Traffic to: select ‘Alias to S3 website endpoint’ > select region where the website was hosted > select S3 endpoint (the bucket holding the website data) > Create records.



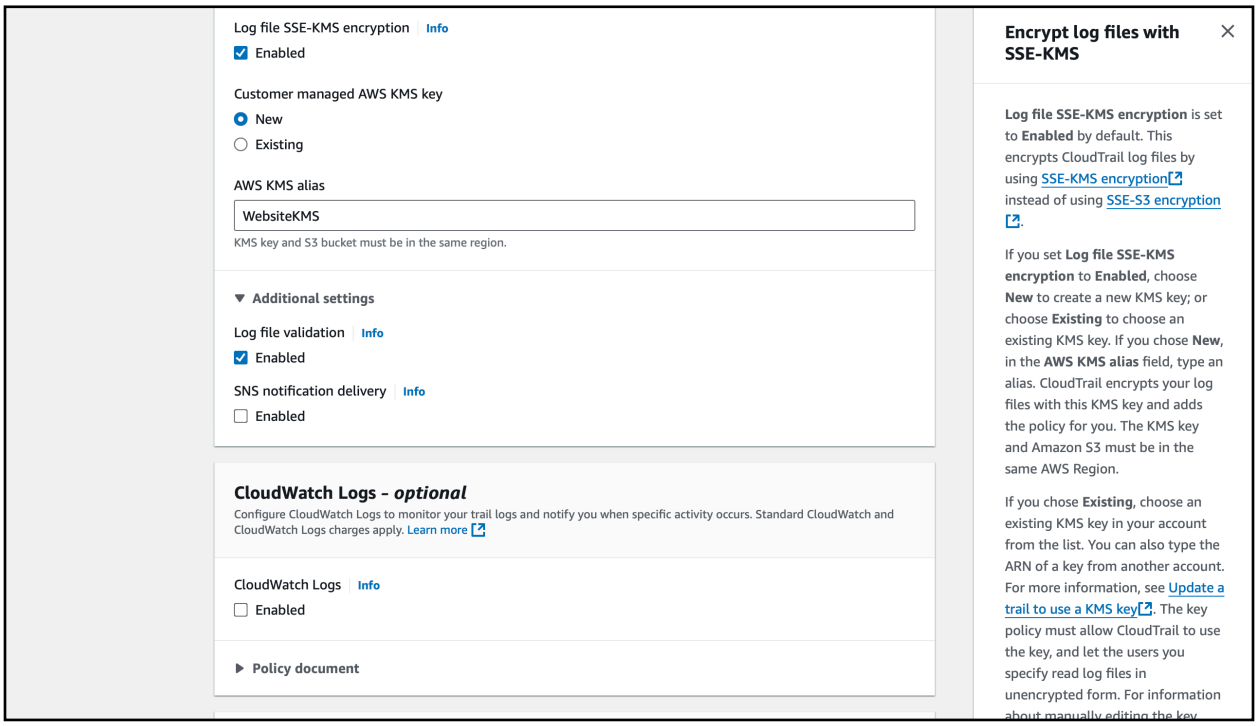
D. AWS CloudTrail

i. Enable Website Monitoring and Analytics (With Encryption)

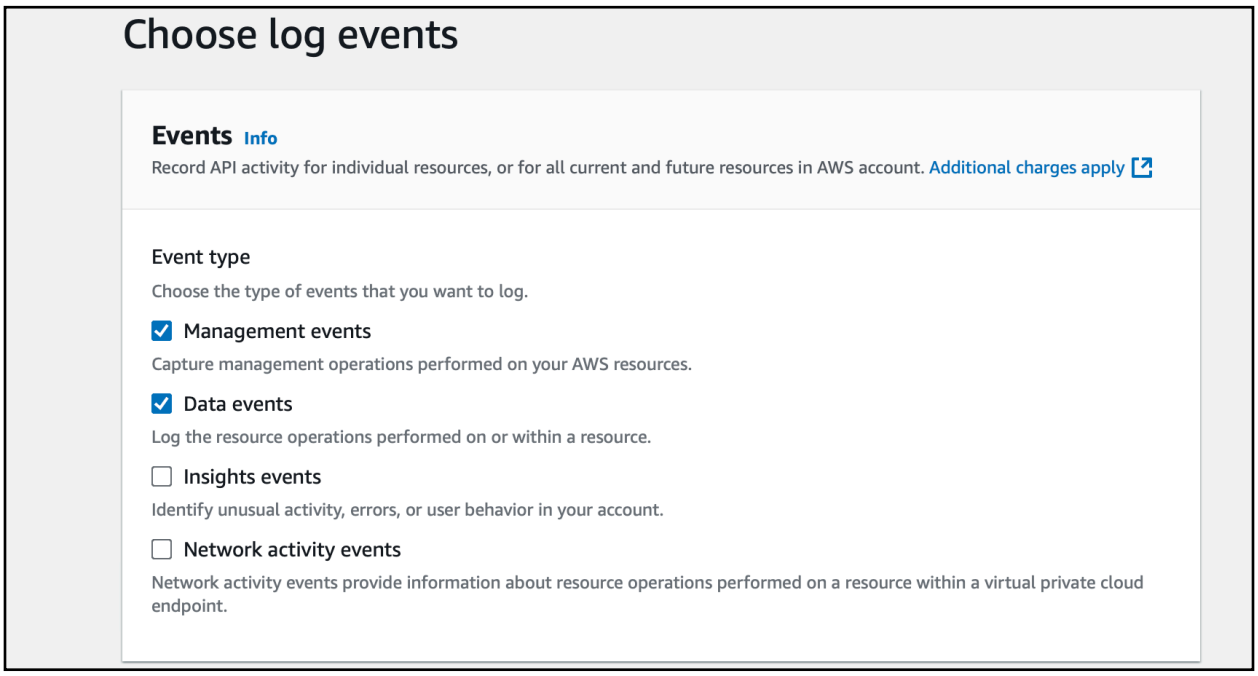
1. CloudTrail > Create a Trail > Trail name: type ‘WebsiteTrail’ > Storage location: select ‘Use existing bucket’ > Trail log bucket name: browse and select ‘samueldiaz.tech' > Log file SSE-KMS encryption: check ‘enabled’ >…



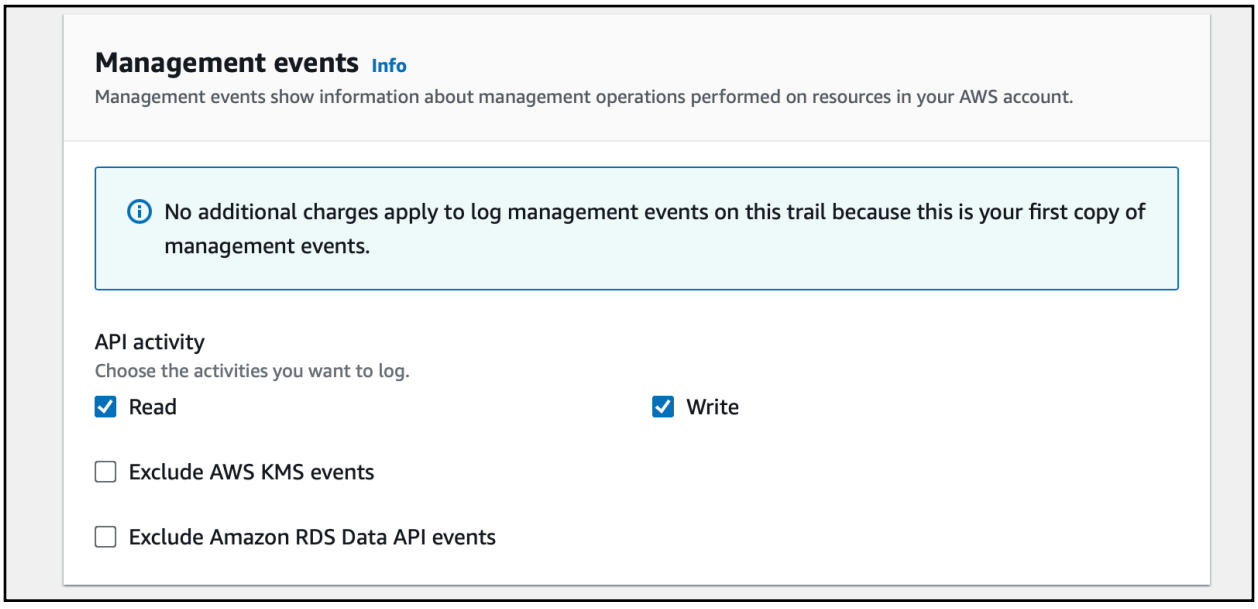
2. …> AWS KMS alias: type ‘WebsiteKMS’ > Log file validation: check ‘Enabled’ > select ‘Next’



3. Choose log events > Events: select ‘Management events’ and ‘Data events’



4. Management events: check ‘Read’ and ‘Write’ > Next > Data events > Resource Type: select ‘S3’ > Log selector template > select ‘Log all events’ > Next

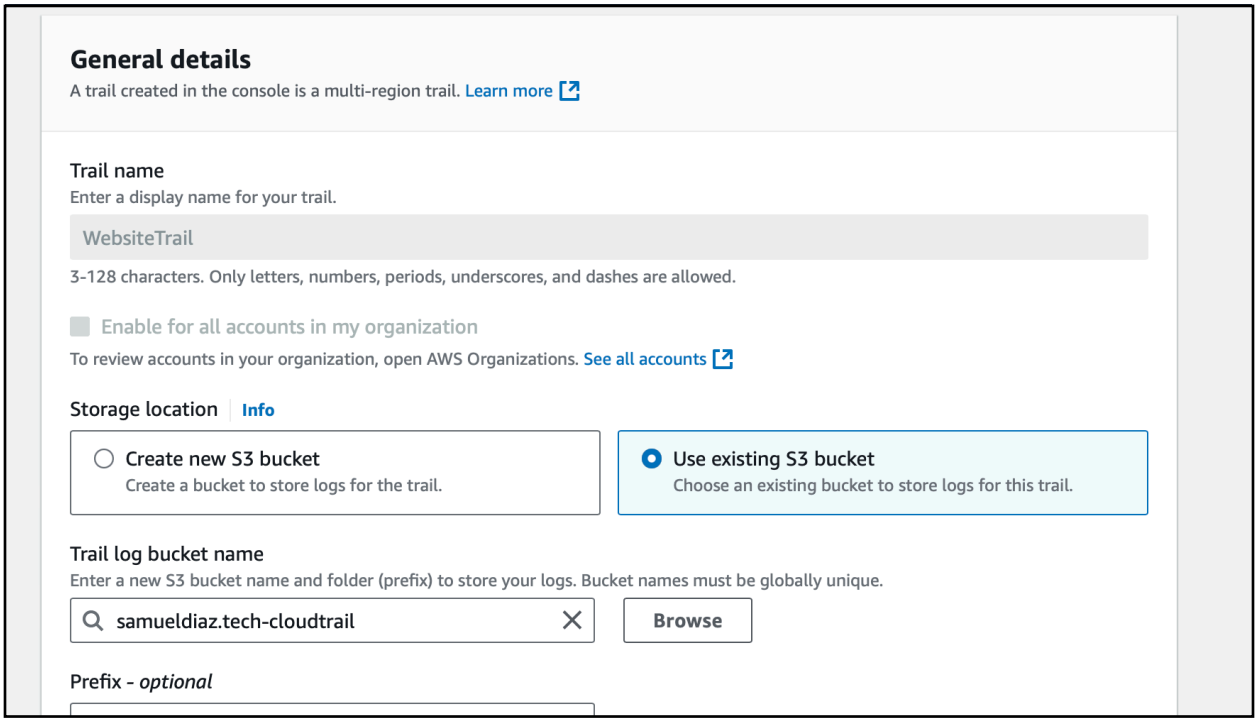


5. Review and Create > review options and click ‘Create Trail’

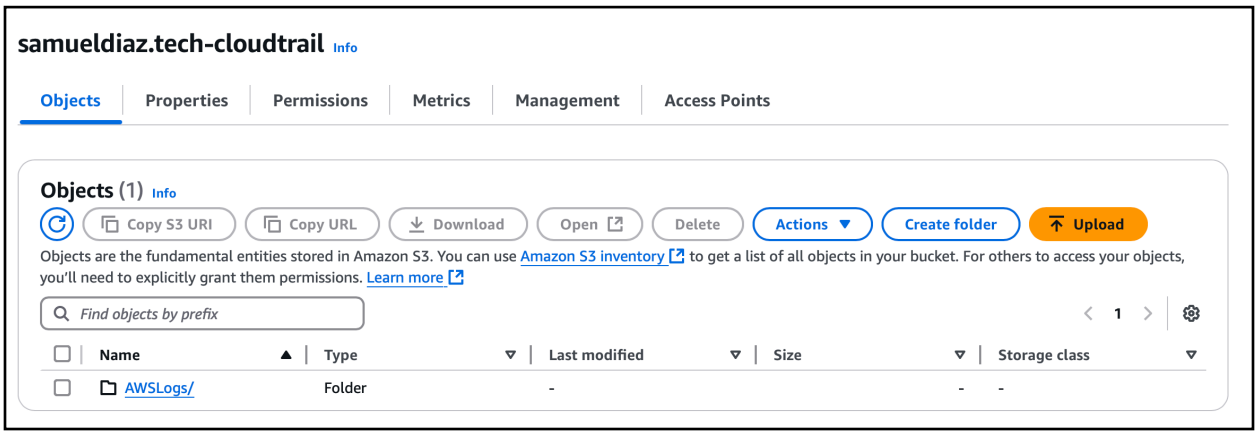
ii. Move CloudTrail to an Encrypted Bucket

1. CloudTrail > Trails > WebsiteTrail > General Details > Edit

2. General details > Storage location > Use existing S3 bucket > Browse > 'samueldiaz.tech-cloudtrail' > Choose > Save changes



3. Successfully moved CloudTrail logs to a private bucket ‘samueldiaz.tech-cloudtrail’ .

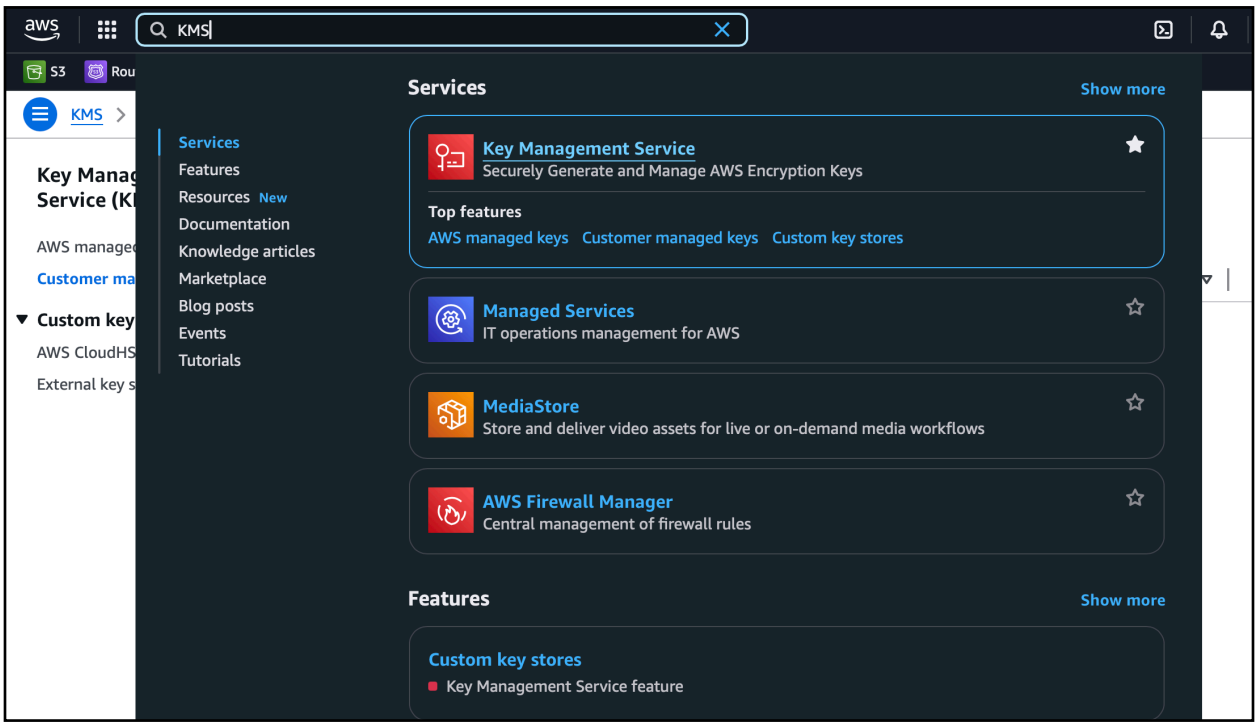


E. AWS Key Management Service (KMS)

i. Verify the Creation of the CloudTrail Key

Note: This step branches from the step 'Enable Website Monitoring and Analytics (with Encryption)' where you enabled CloudTrail with KMS.

1. AWS Homepage > Search bar: type and select 'Key Management Service'



2. Verify whether the key created in the previous step has been successfully generated.