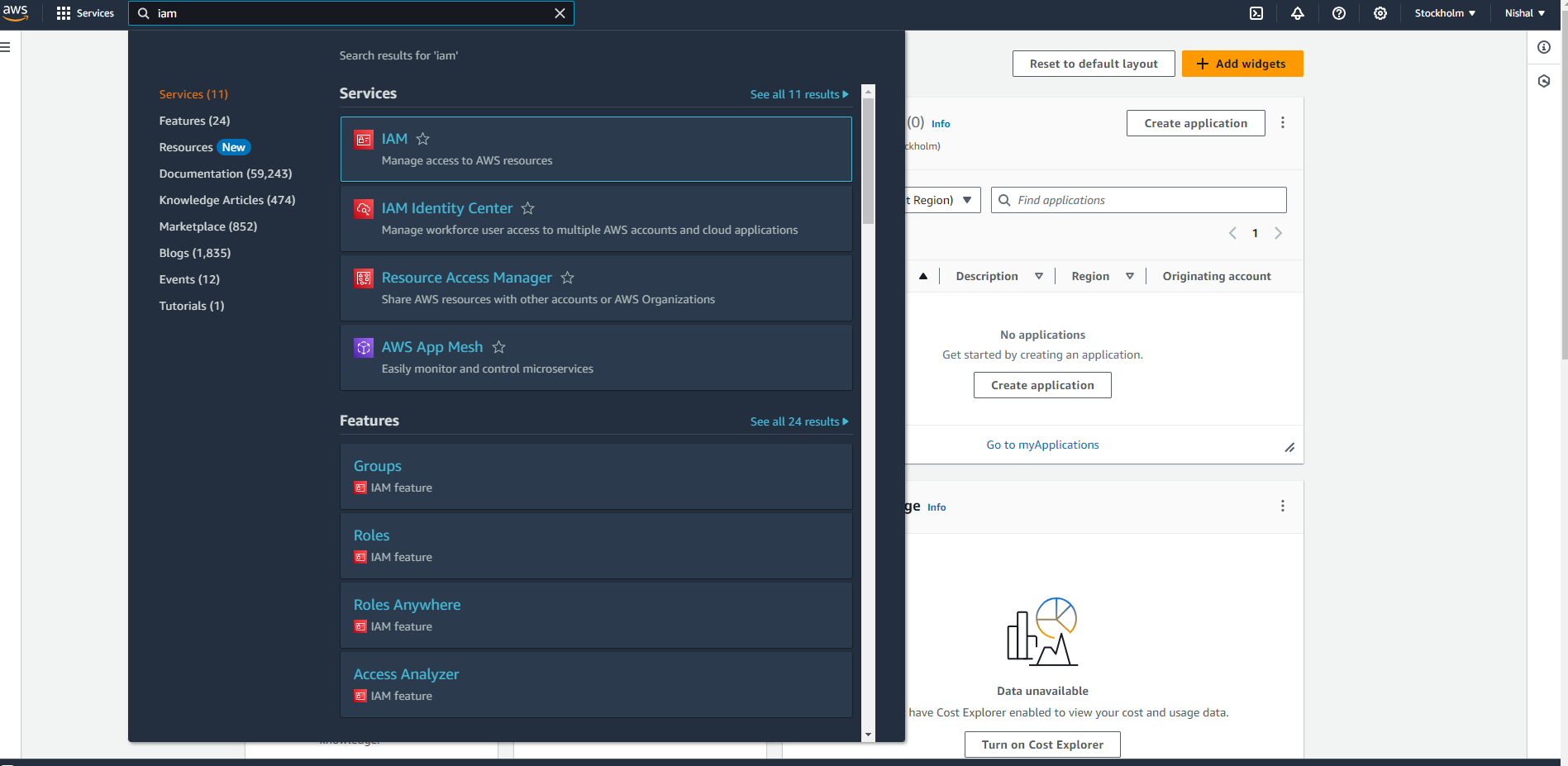
Name : Nishal Dbritto

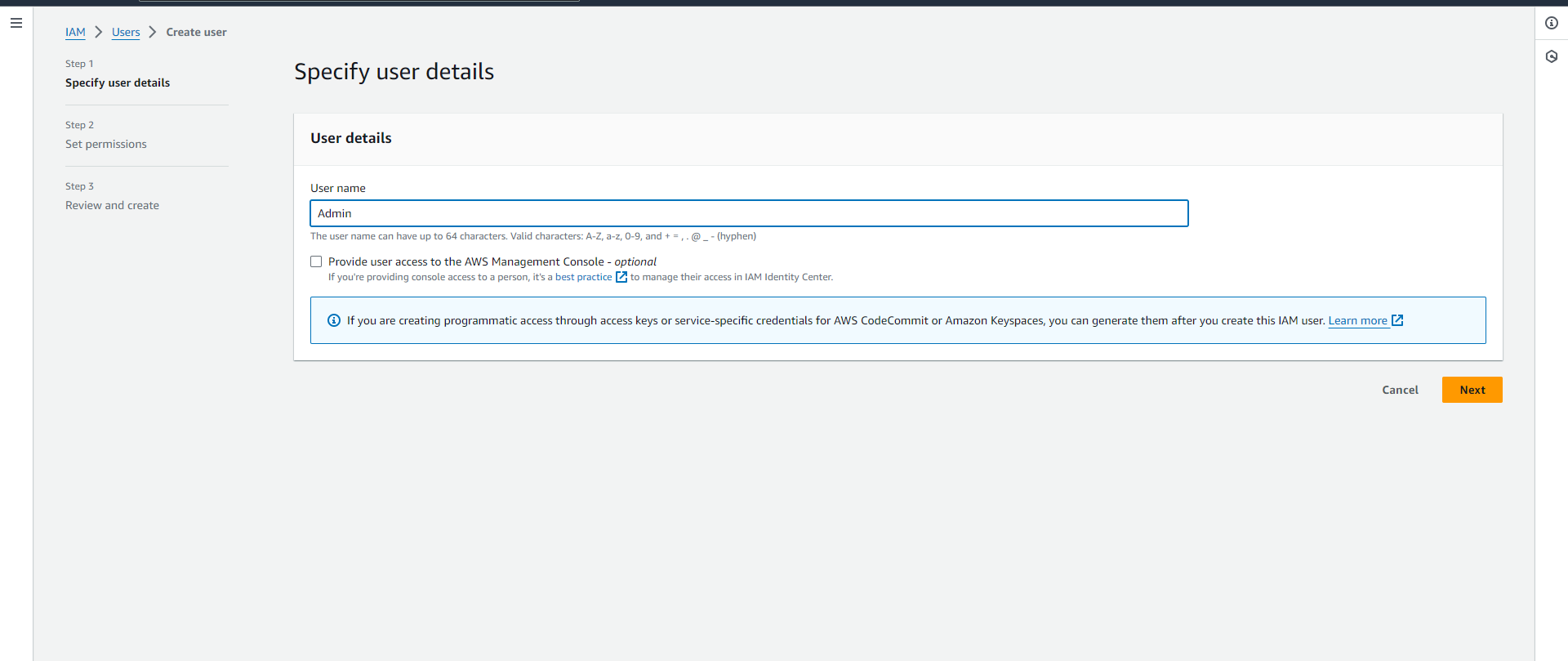
Roll No: A009

Sap Id : 86062300025

CC PRAC 3

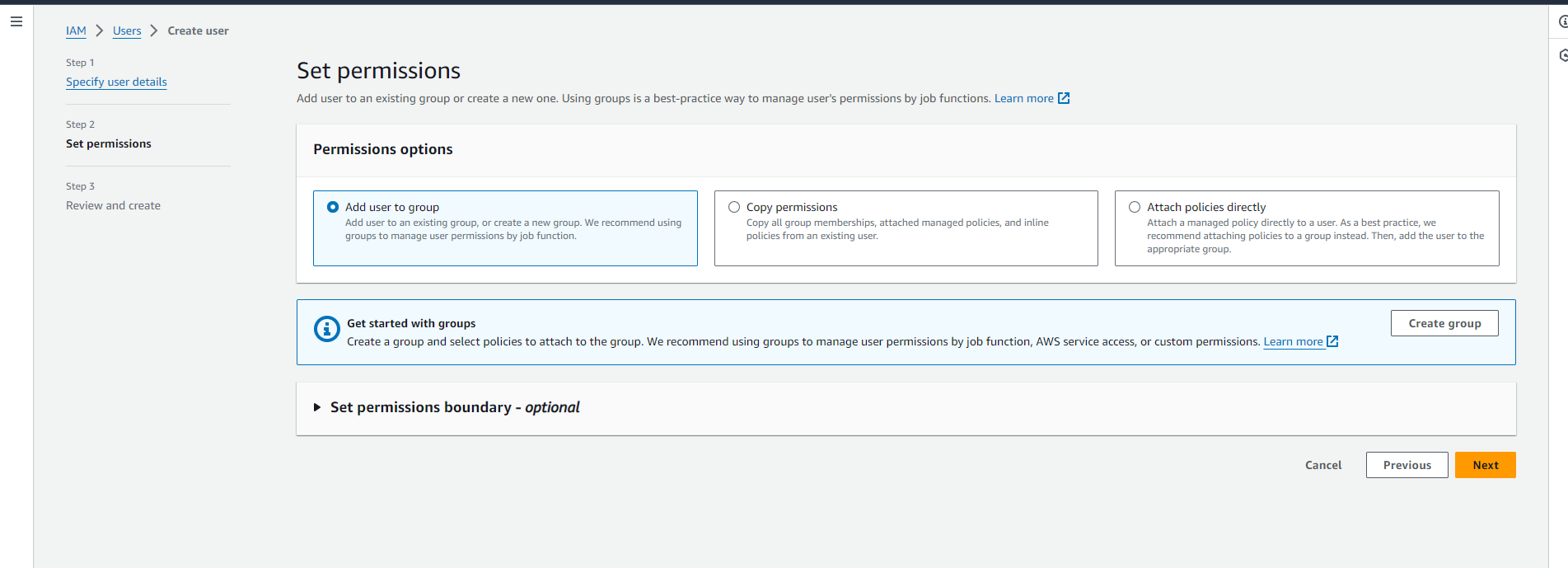


In search bar select iam

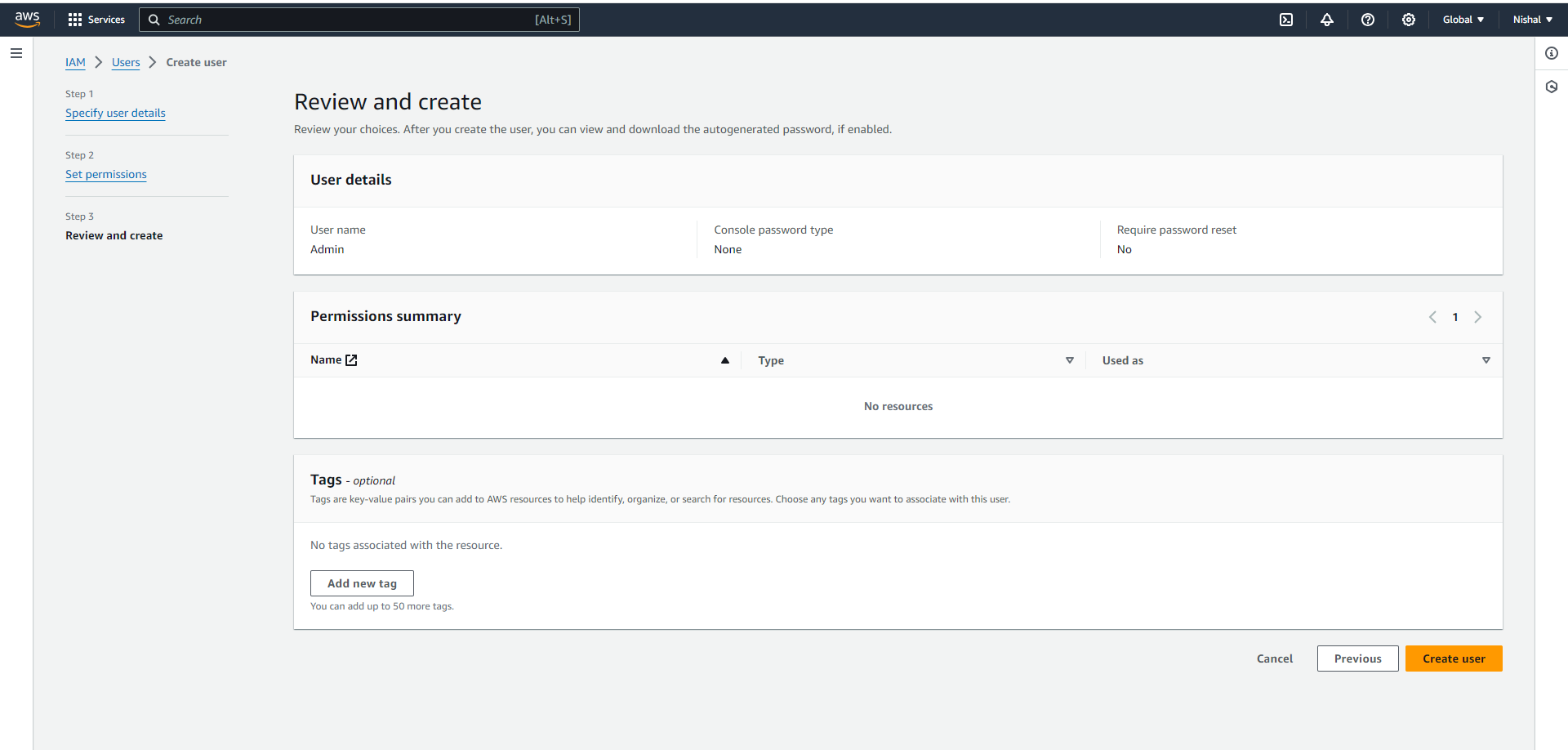


In iam dashboard navigate to user and click add user

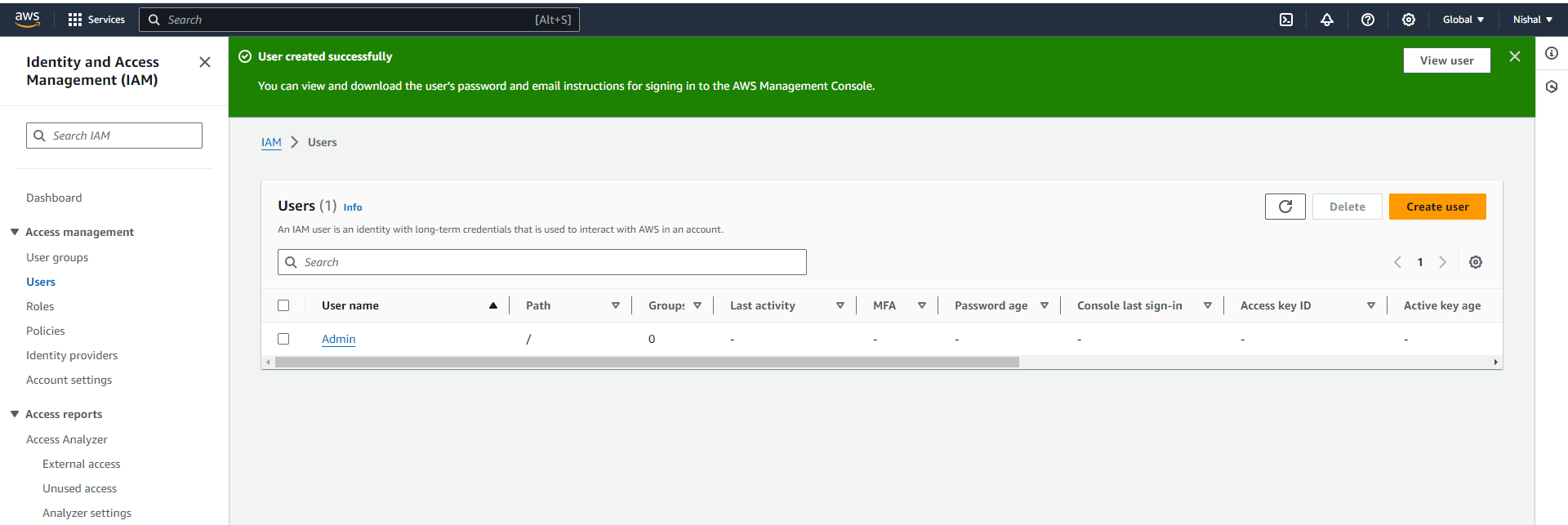
Specify the user name and next



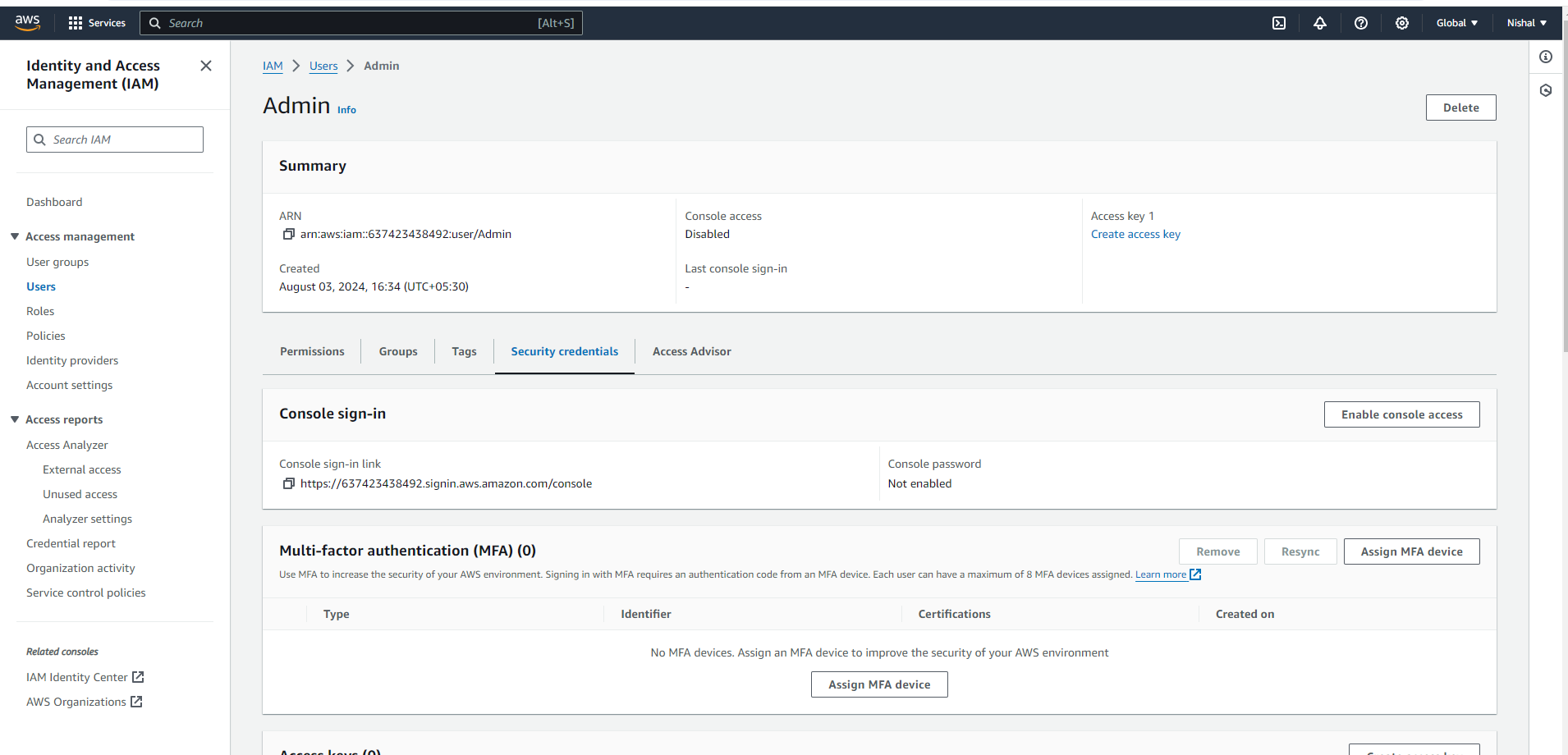
In set permissions select add user to group and click next



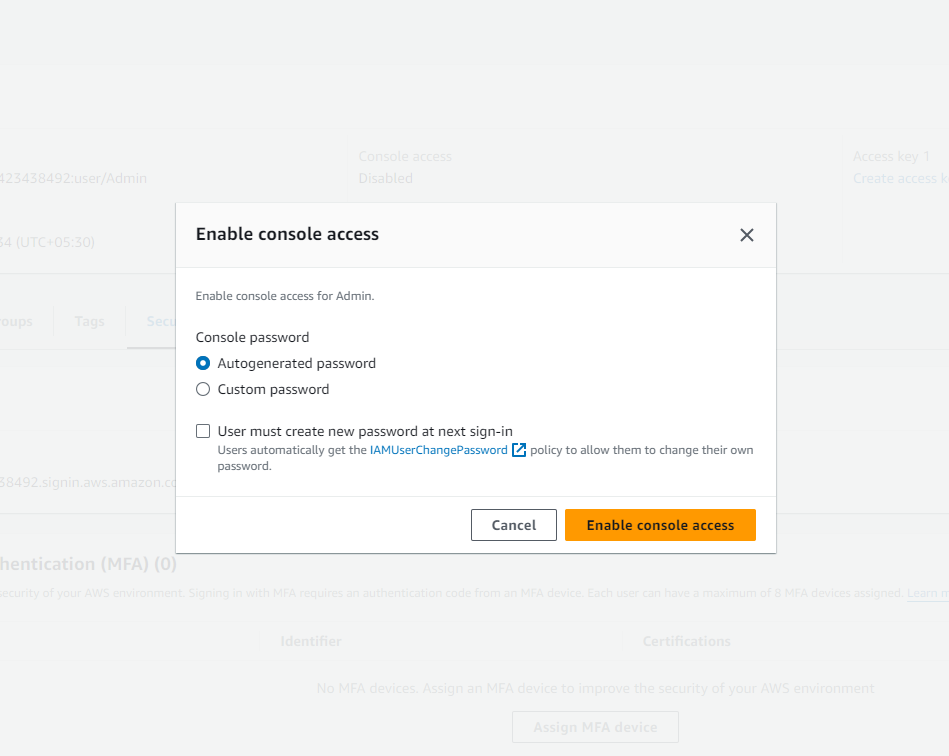
Click on create user

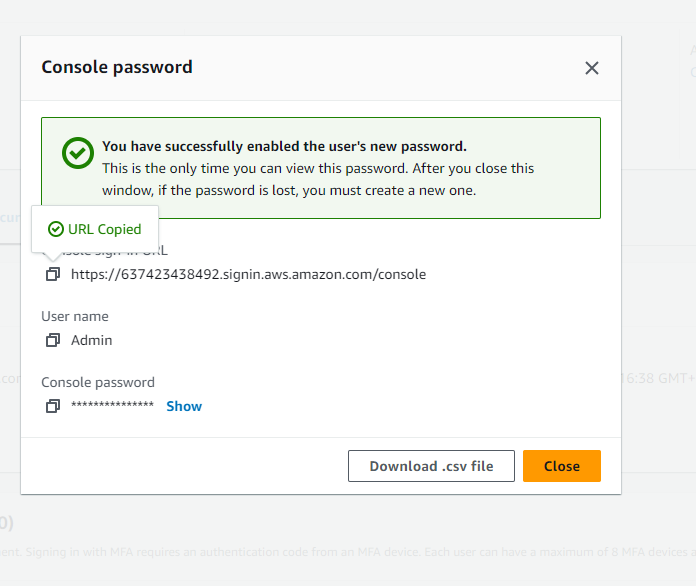


You have created the user

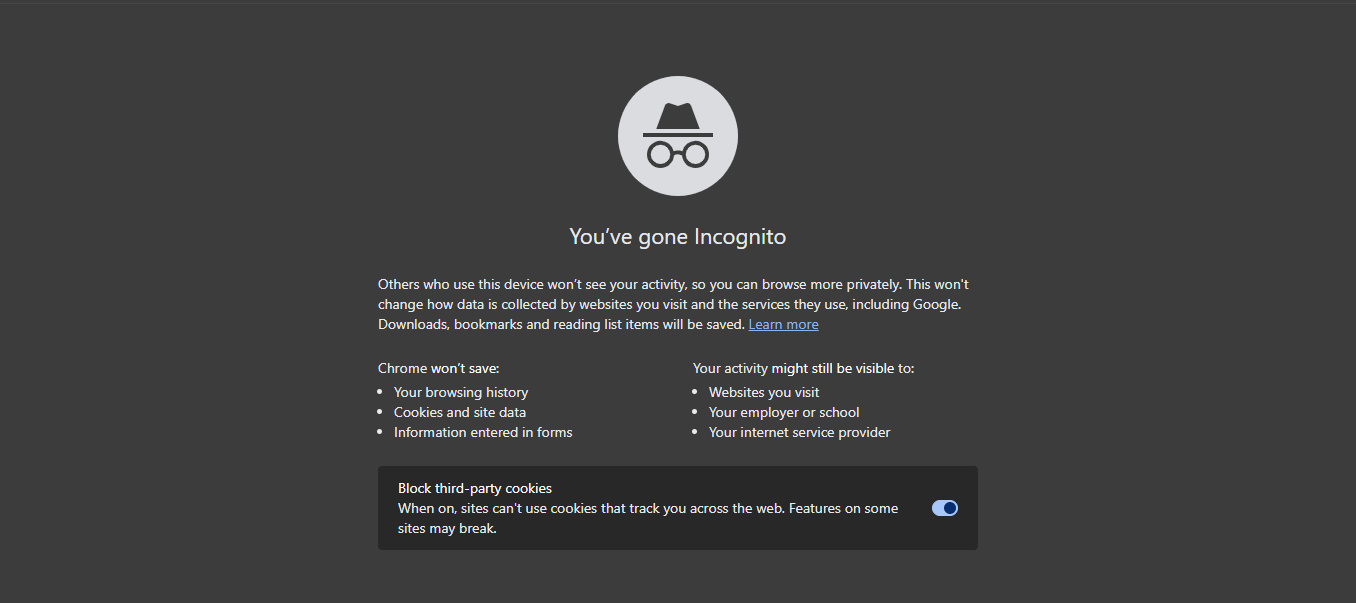


Click on user name created, go to security credentials and click on enable console access

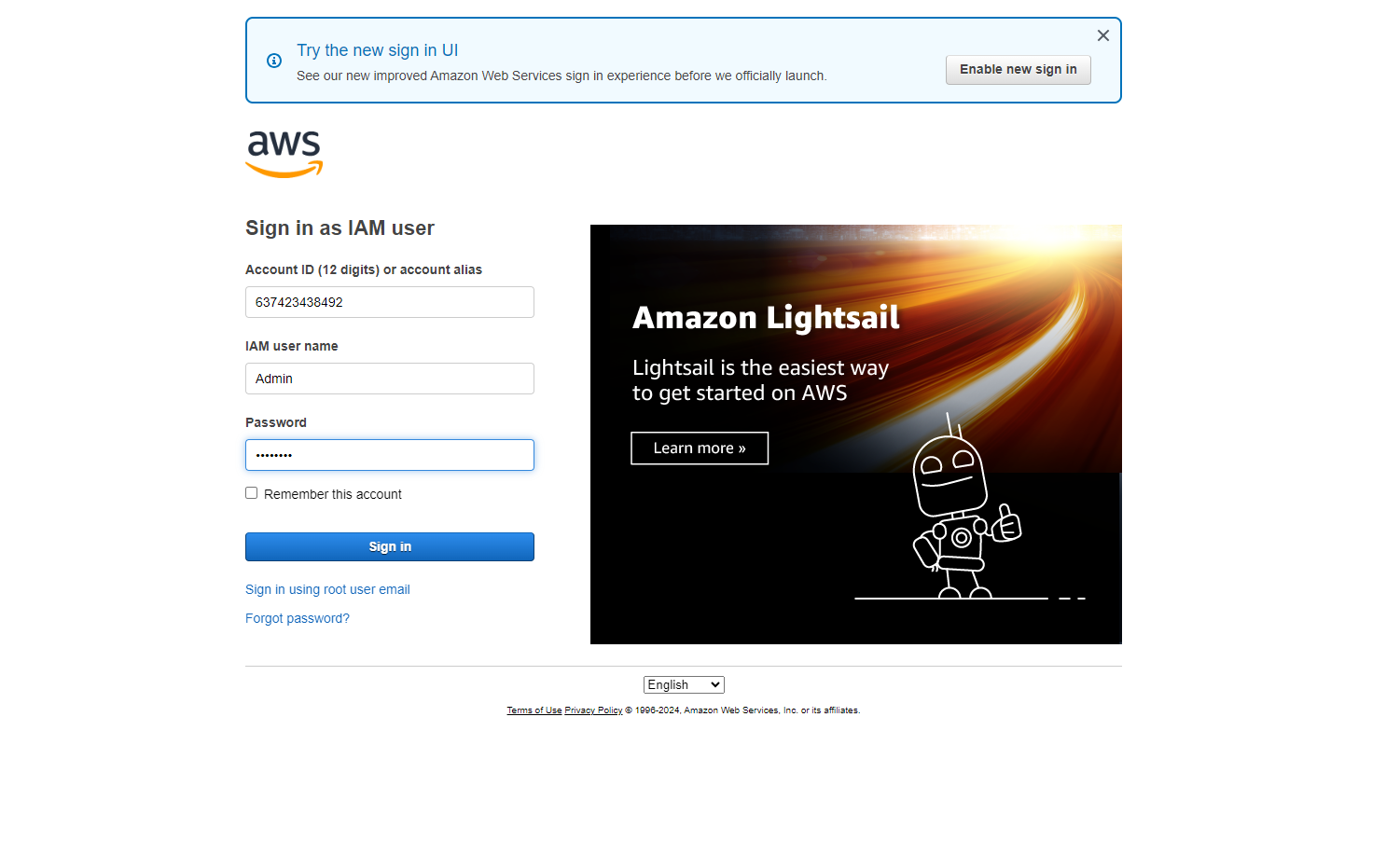




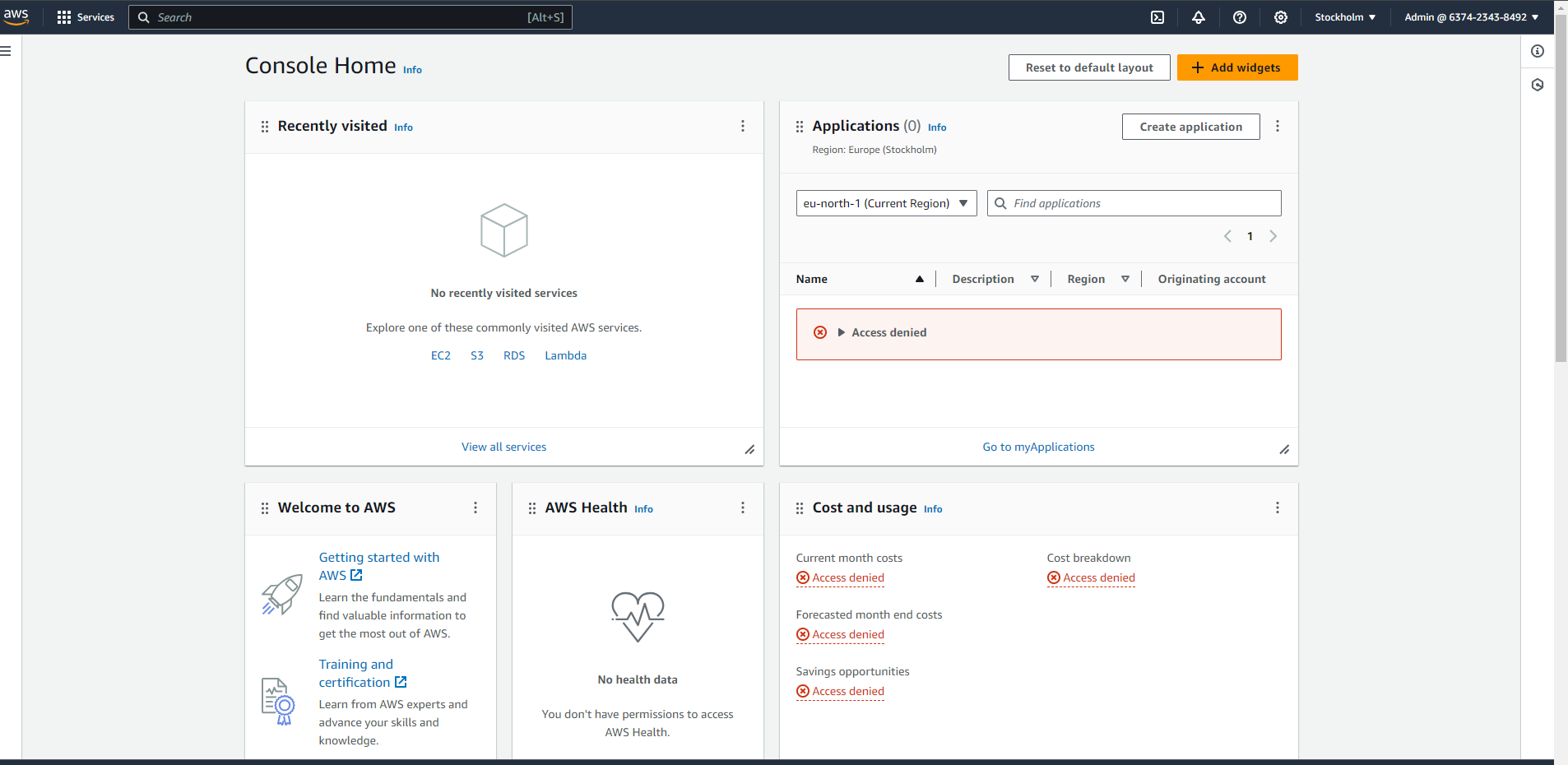
Copy url and click on download.csv file and then close



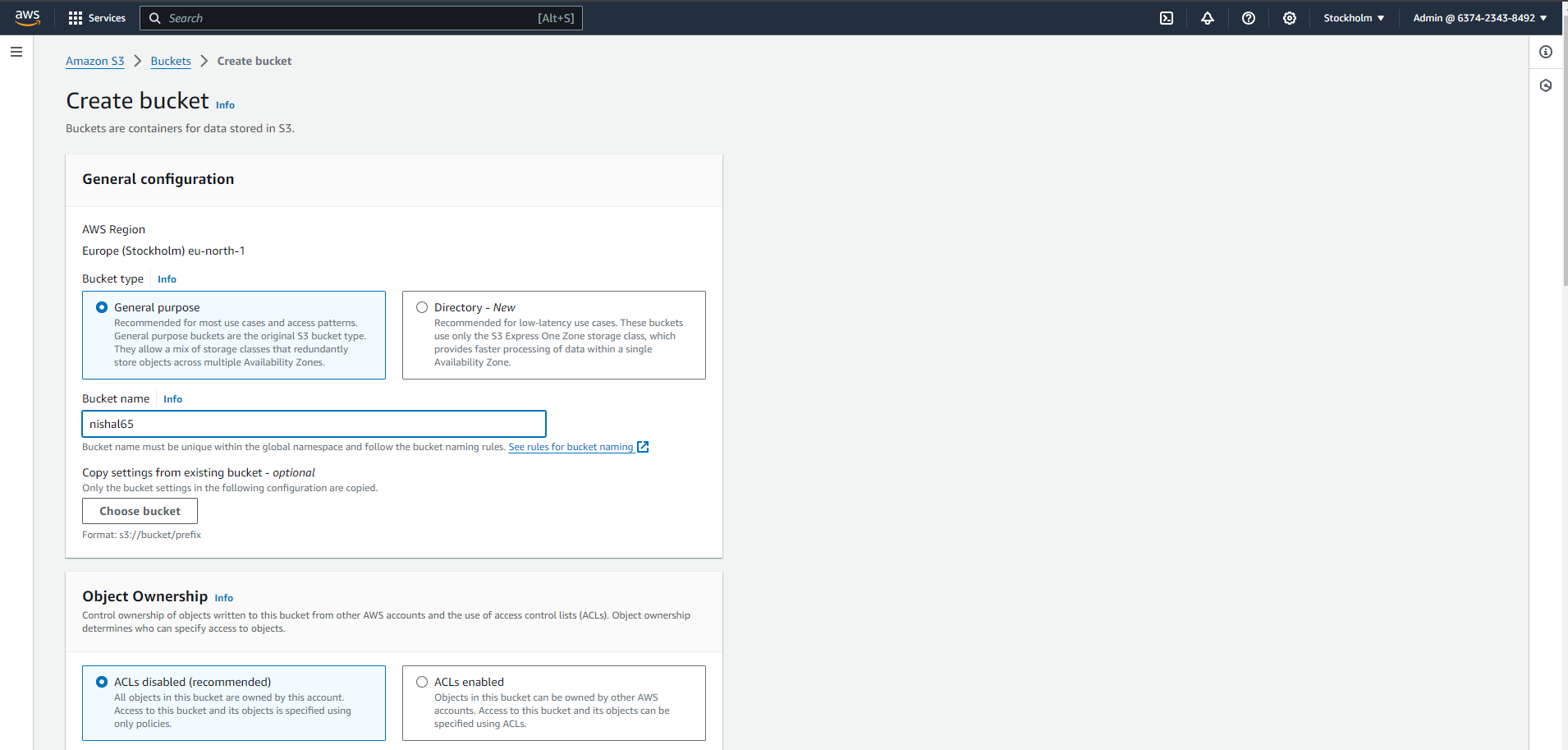
Paste url in incognito



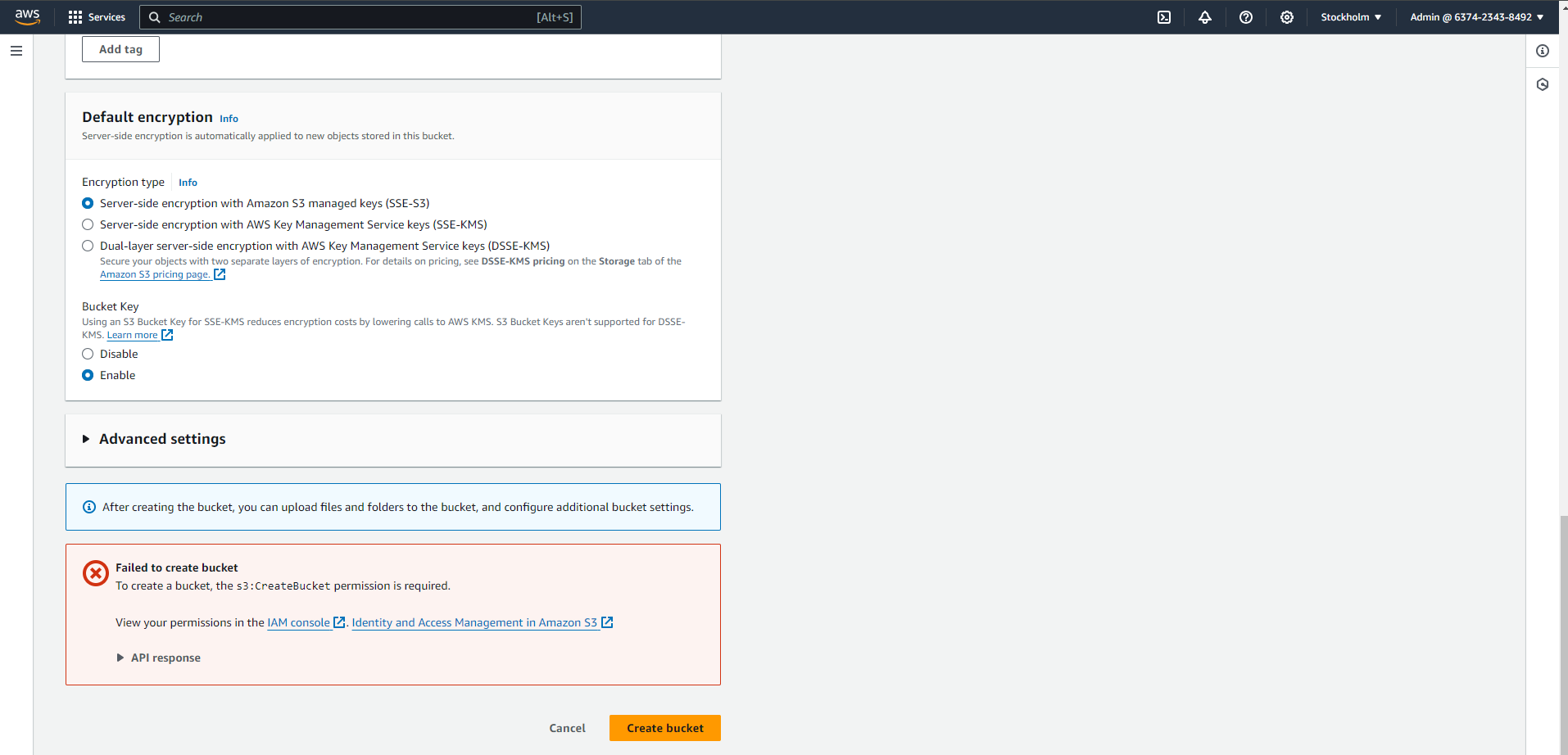
Sign in aws account with your user name and password downloaded earlier



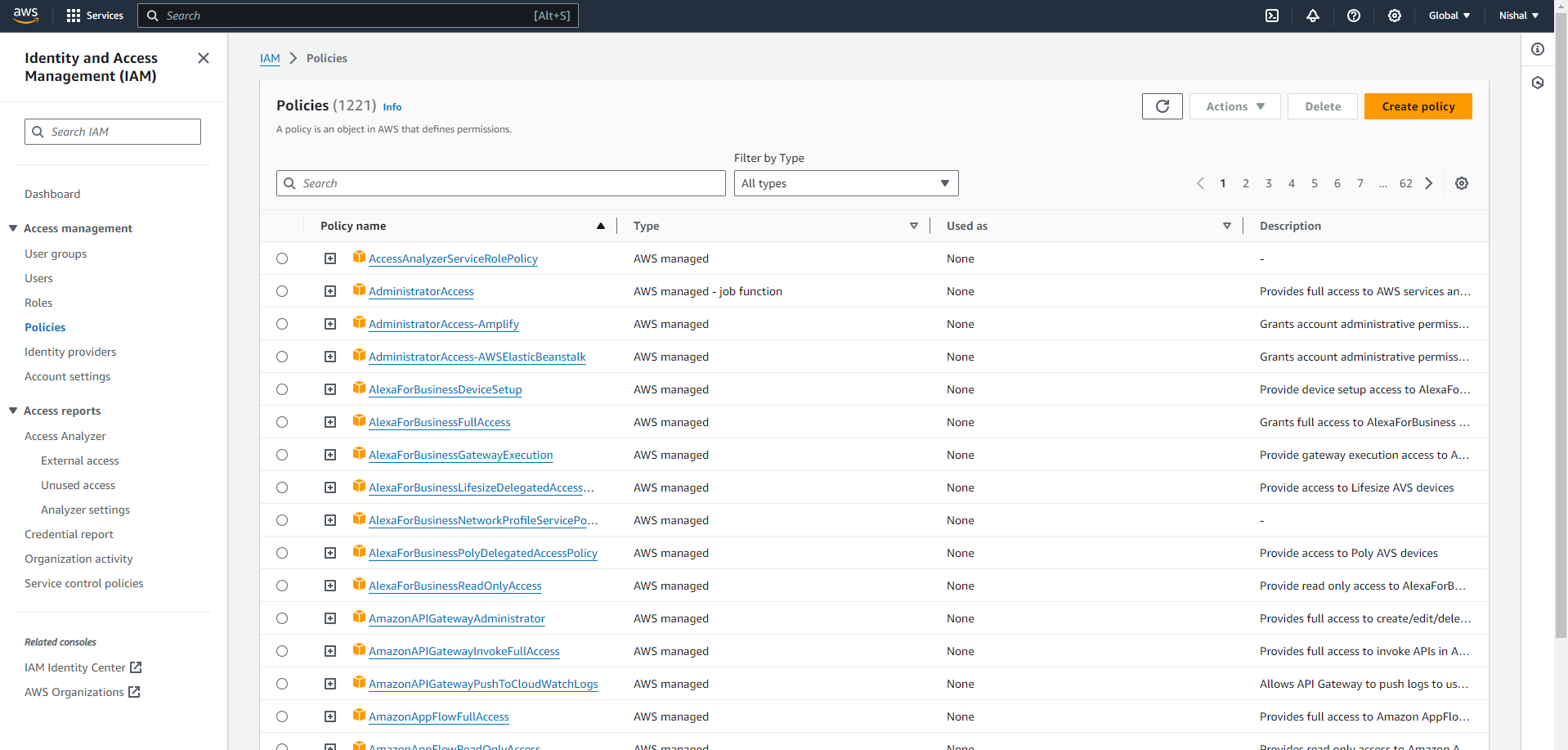
You will be directed to the above console and you can access any of the services , we select S3 service



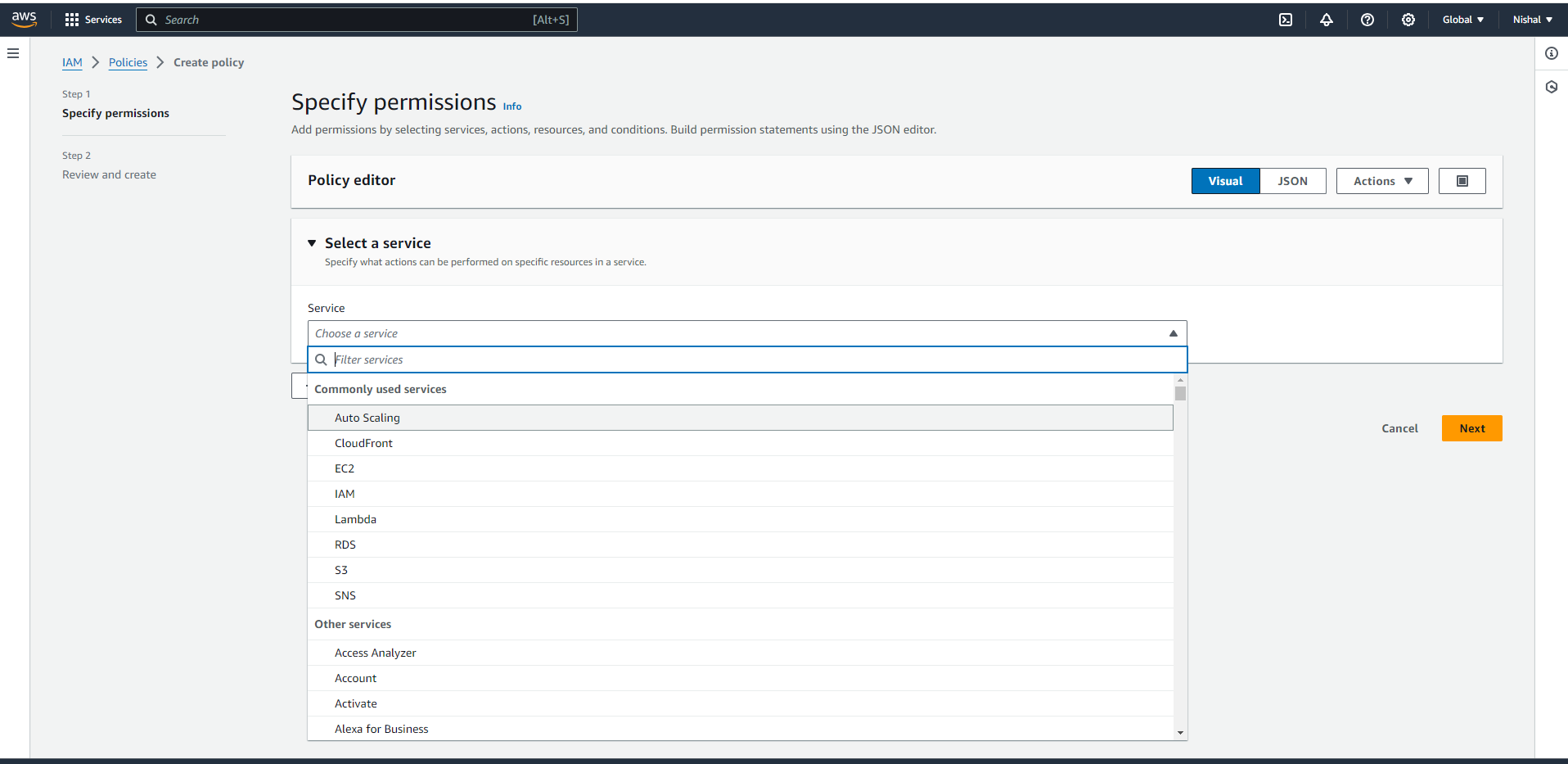
Enter the bucket name



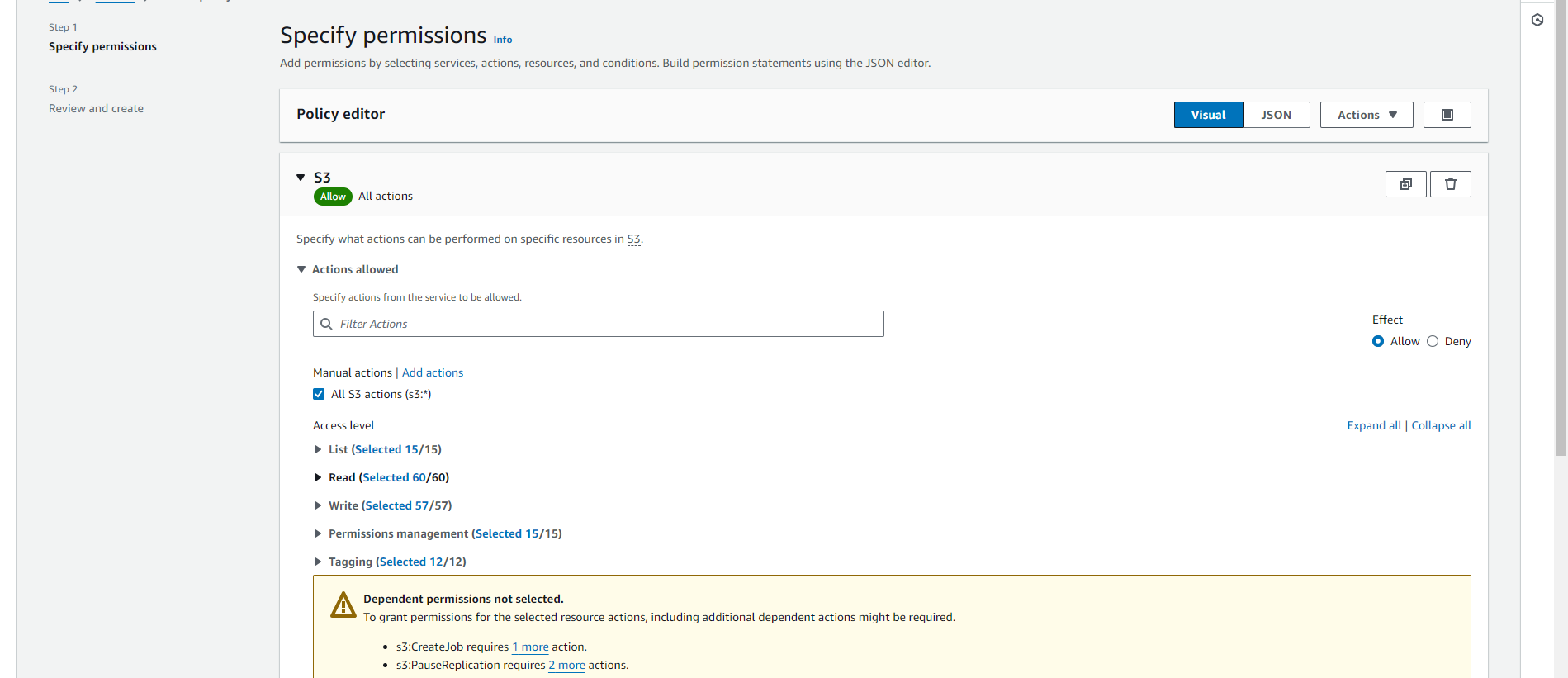
Select on Create bucket , but incognito mode fails to create a bucket



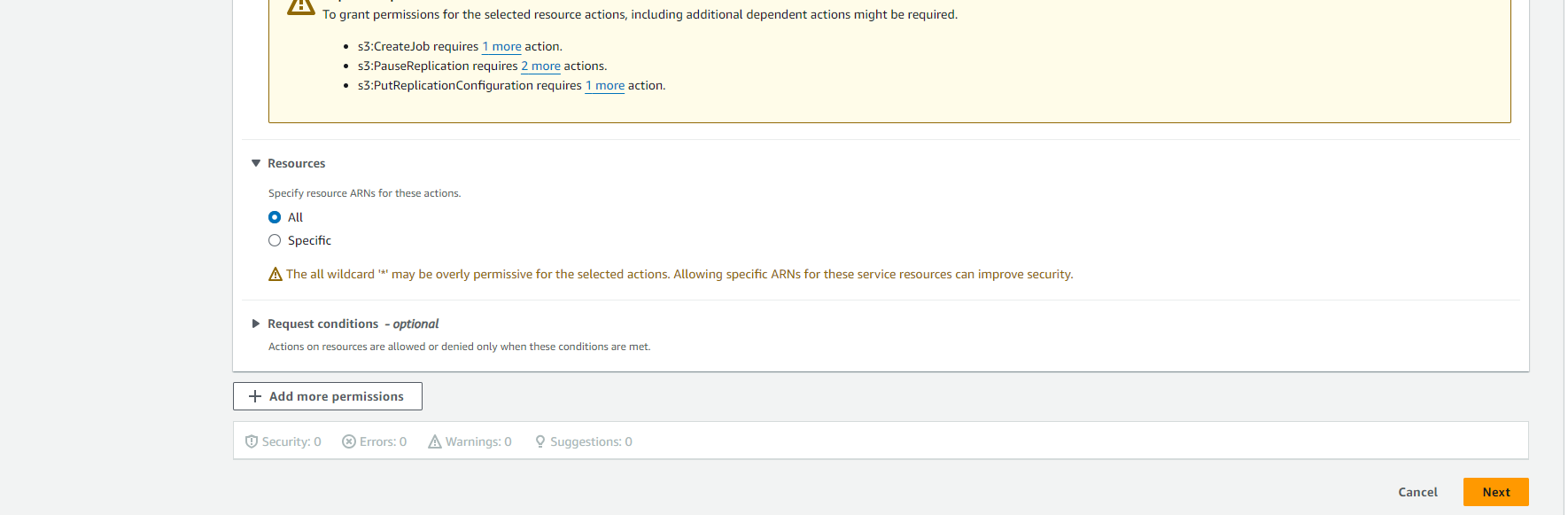
Now go to root user , select policies and click on create policy



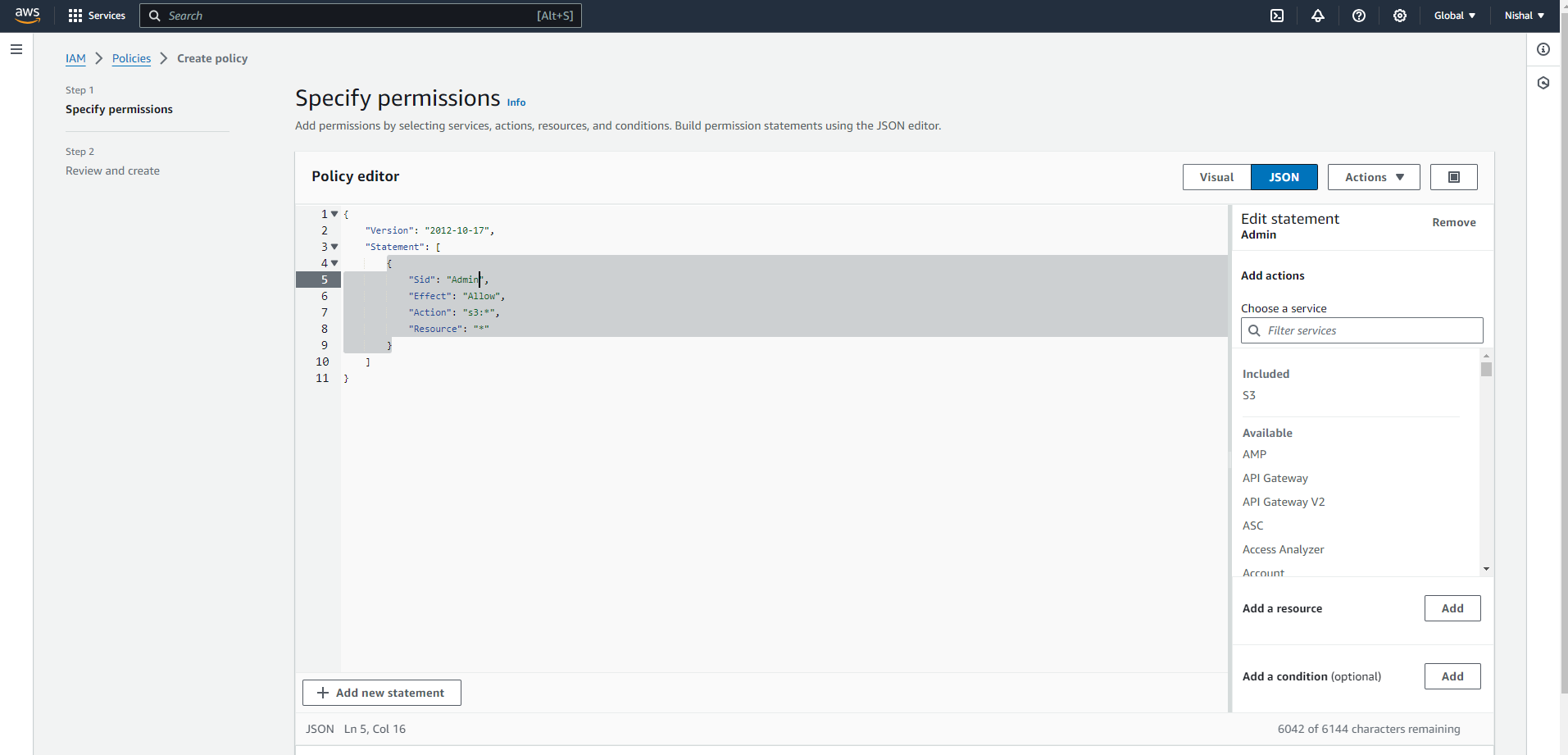
In specify permissions select a service – S3



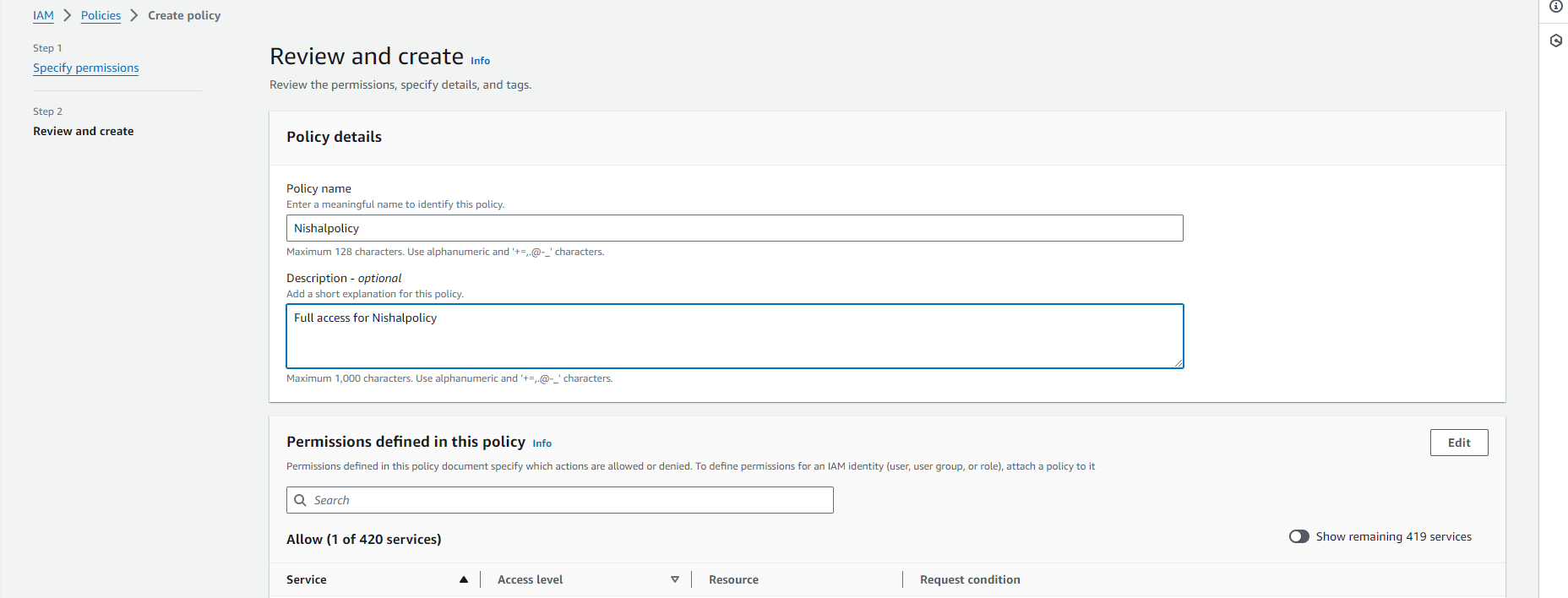
In manual actions select- all S3 actions



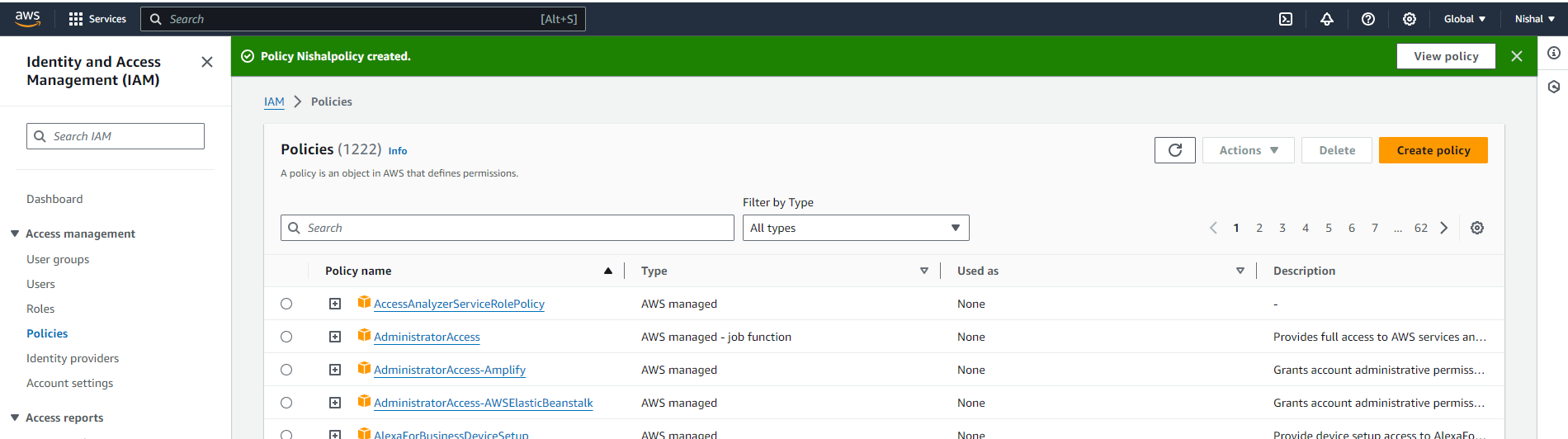
In resources select All

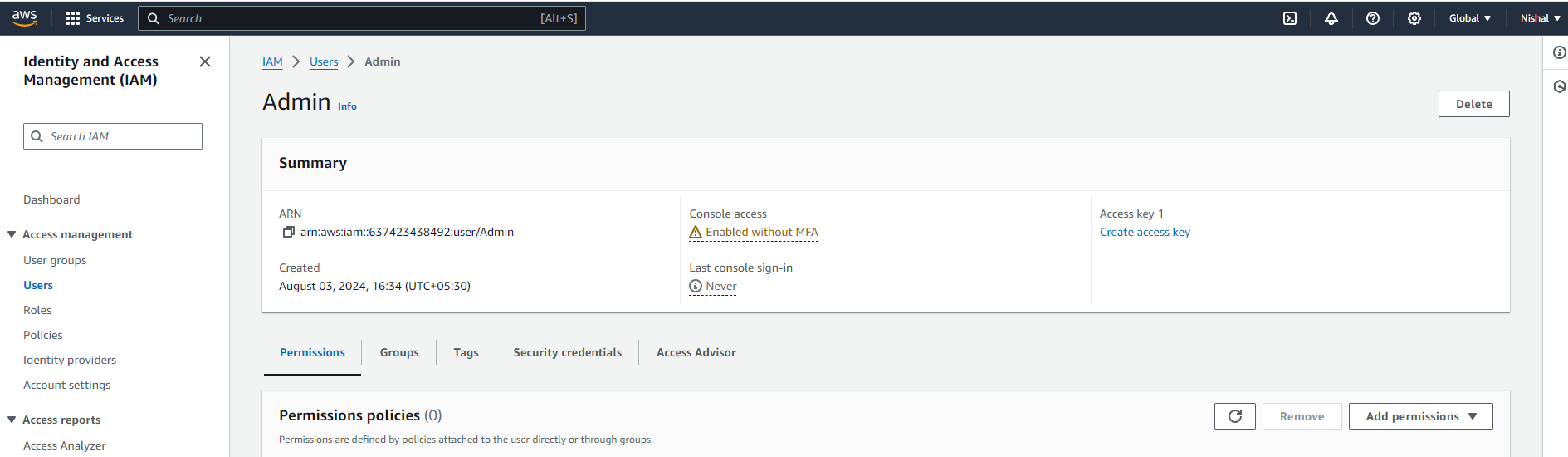


Now go in JSON and enter user name in “Sid”

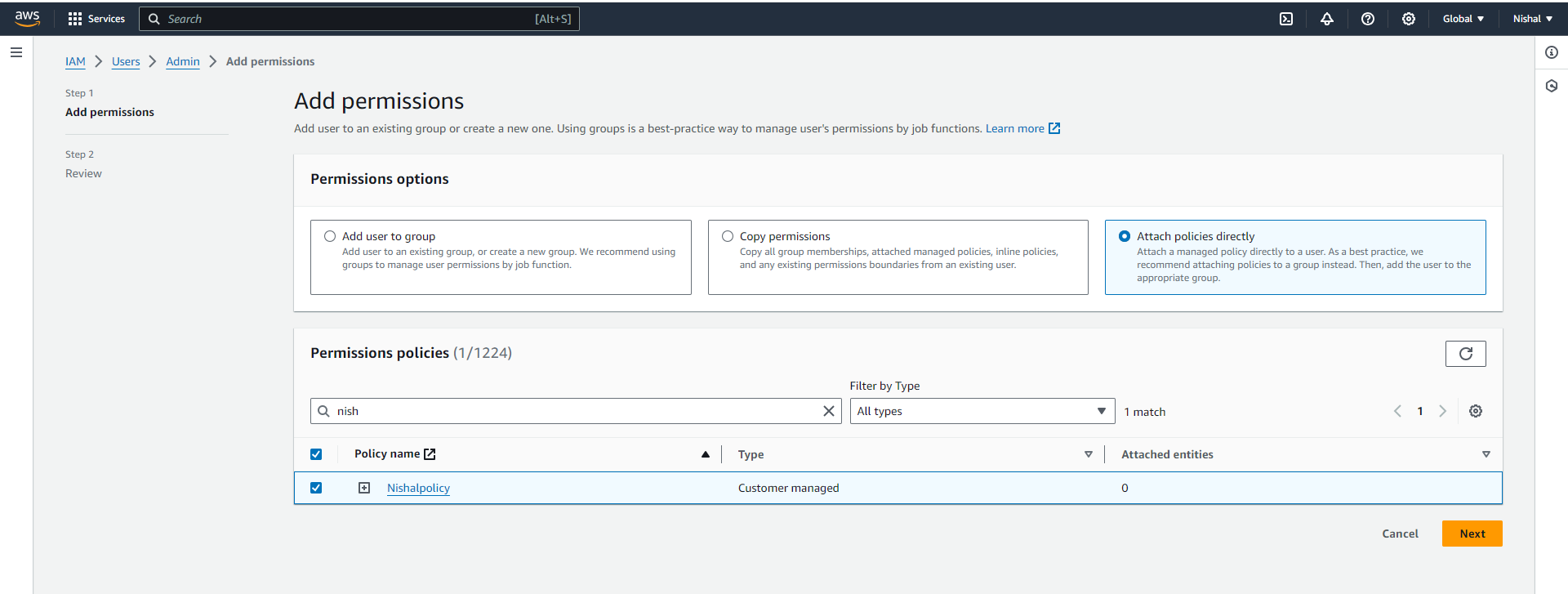


Name the policy and add description(optional) and click on create

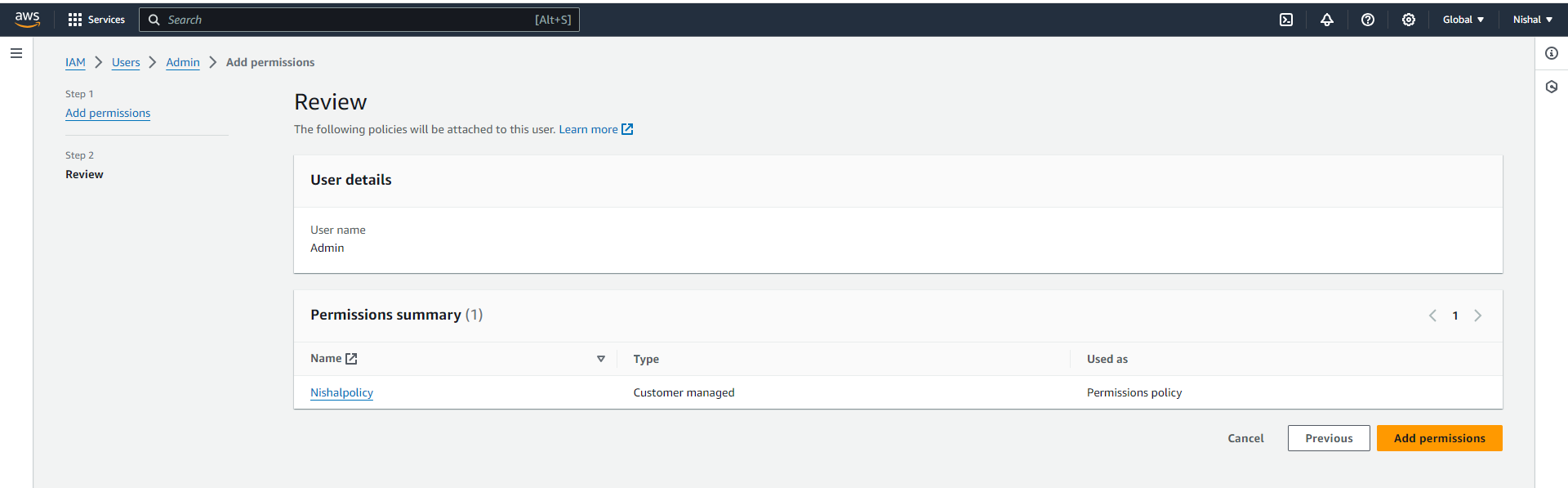




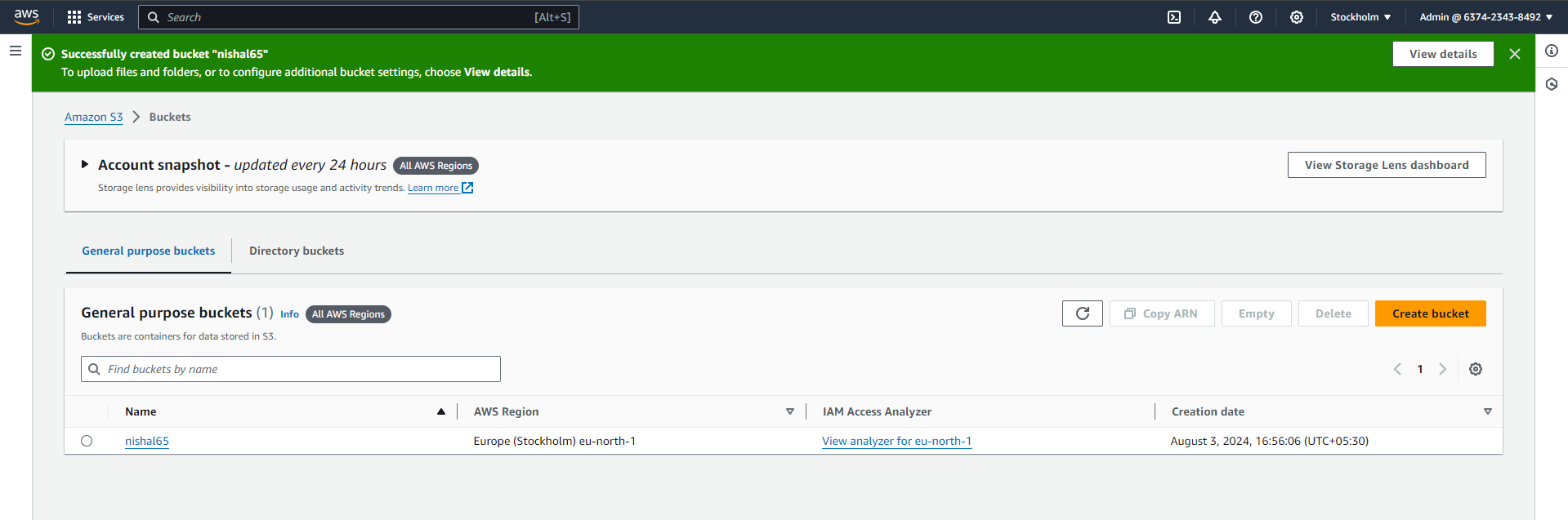
Go to users – go in permisions and select add permisions



Click on attach policies directly click on the policy you have created and click next



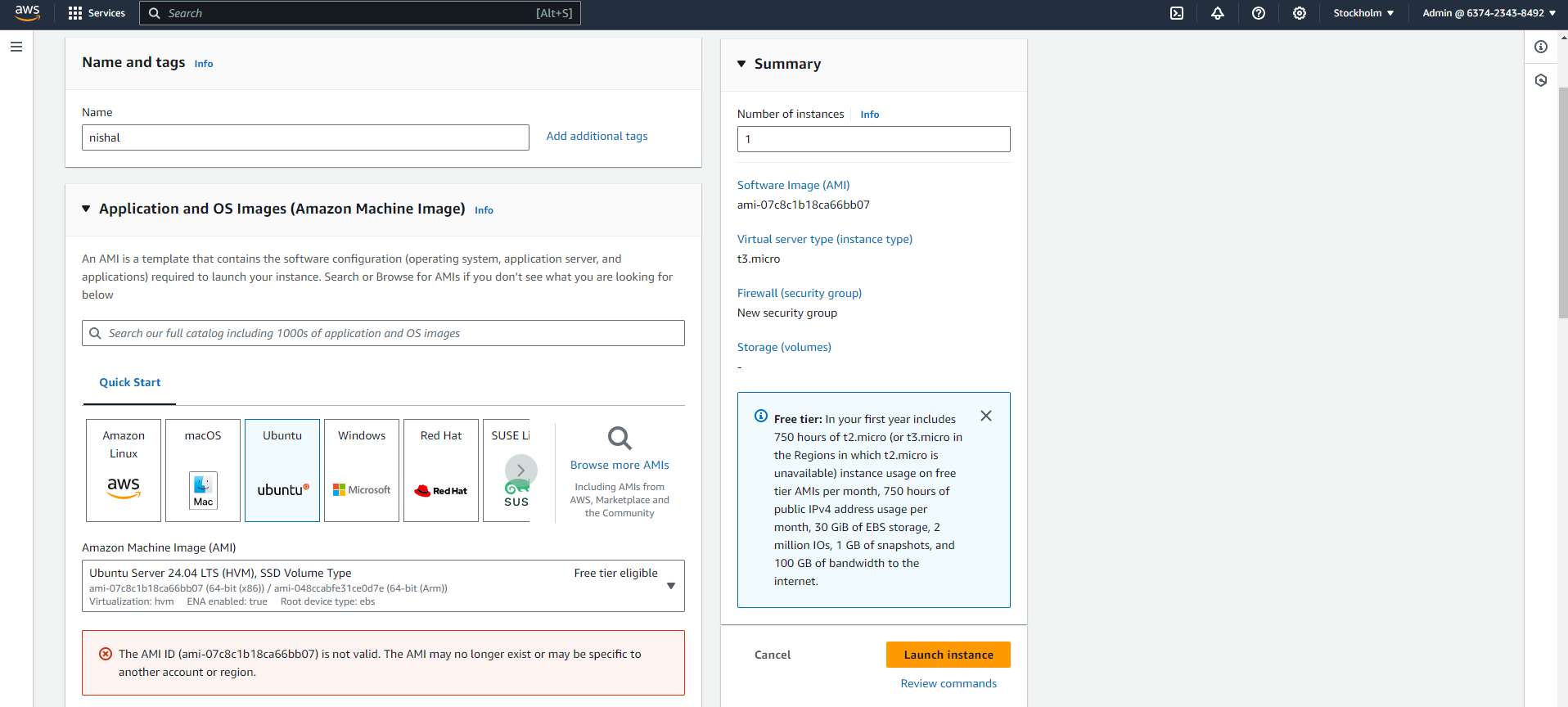
Click on add permissions

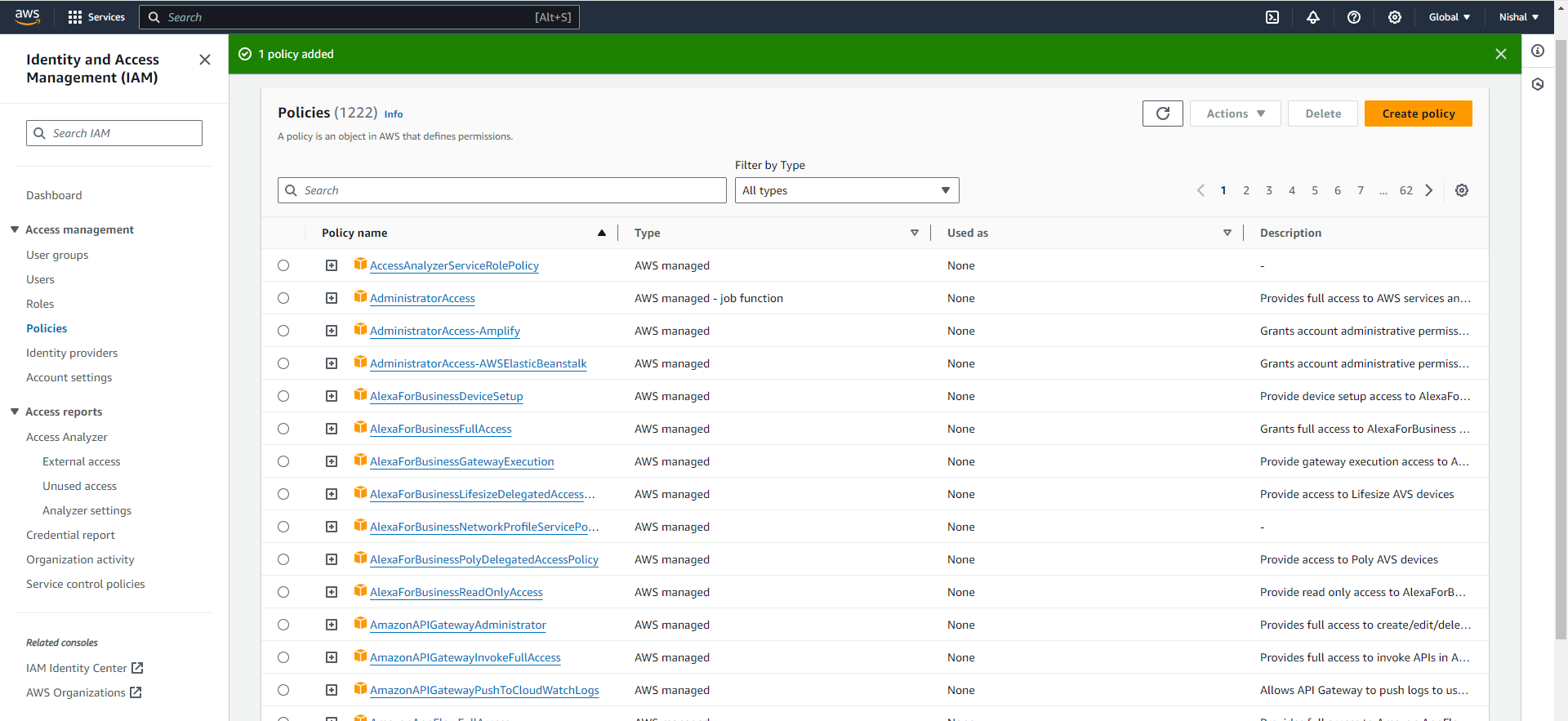


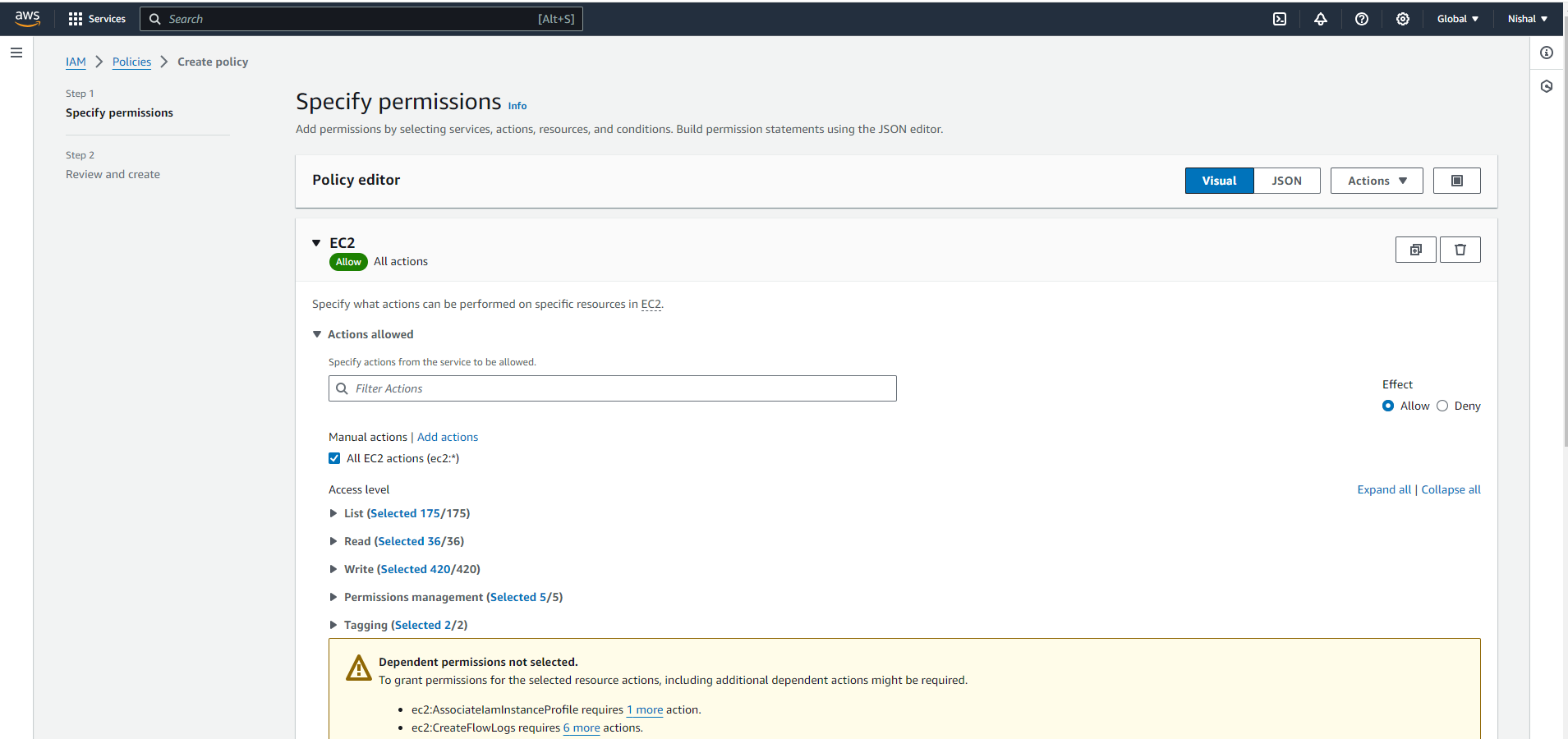
Go to incognito and select on create bucket no your bucket will be created

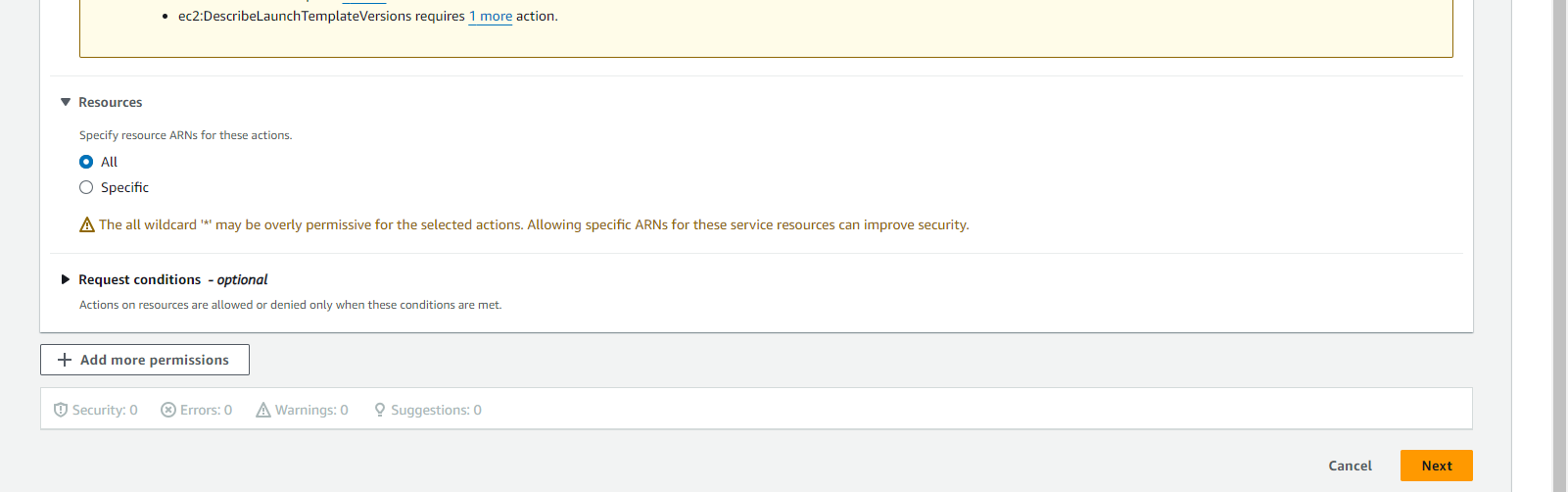
EC2

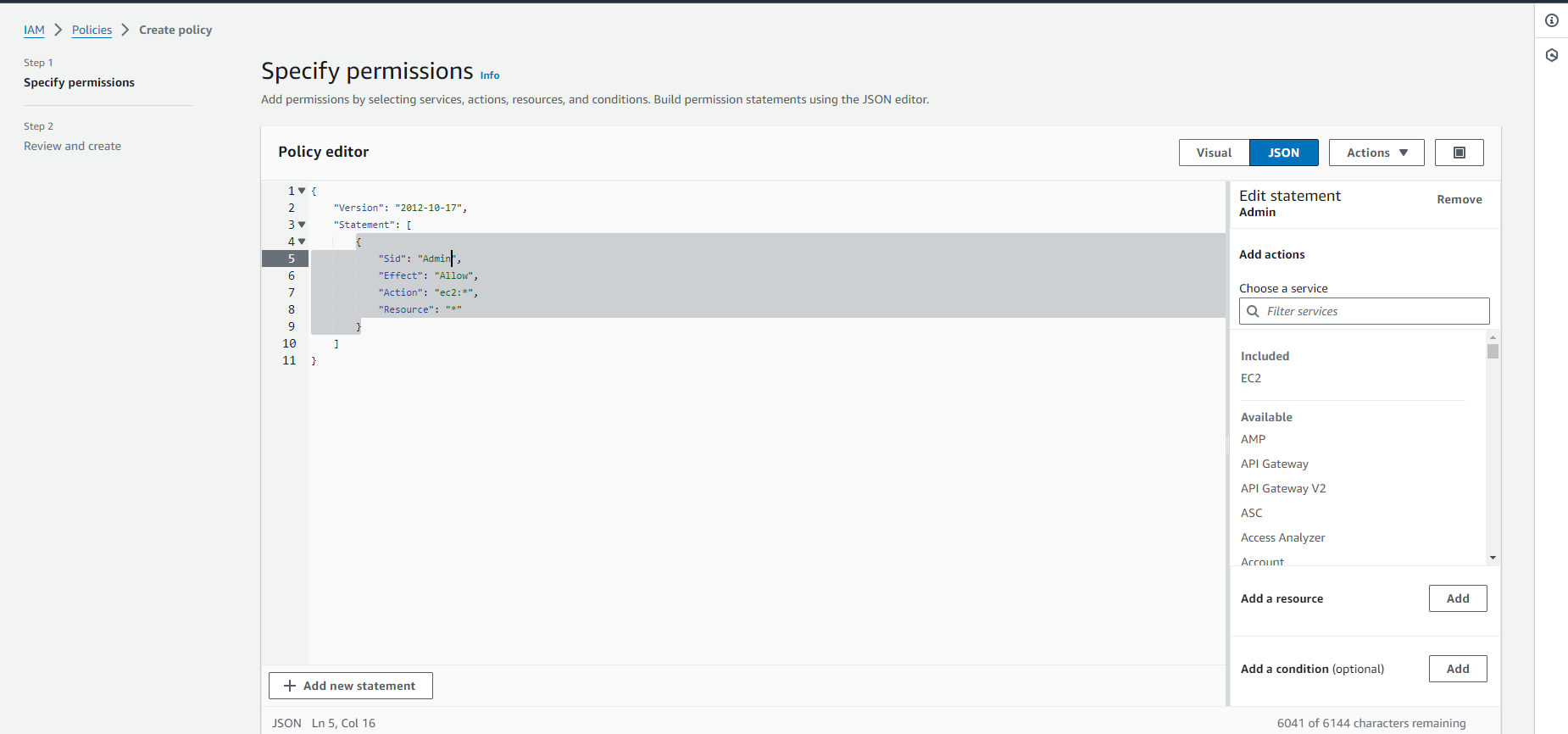
Follow the same steps as above , instead of creating the bucket you have to create the instance

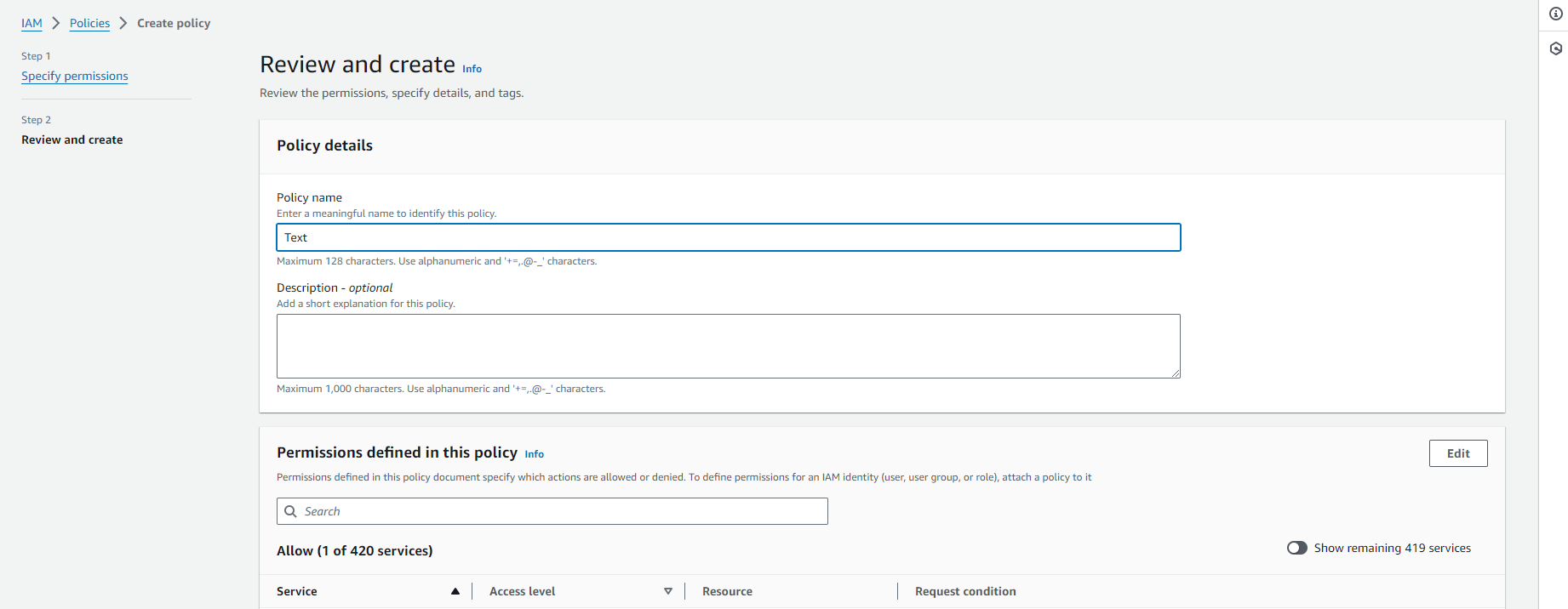


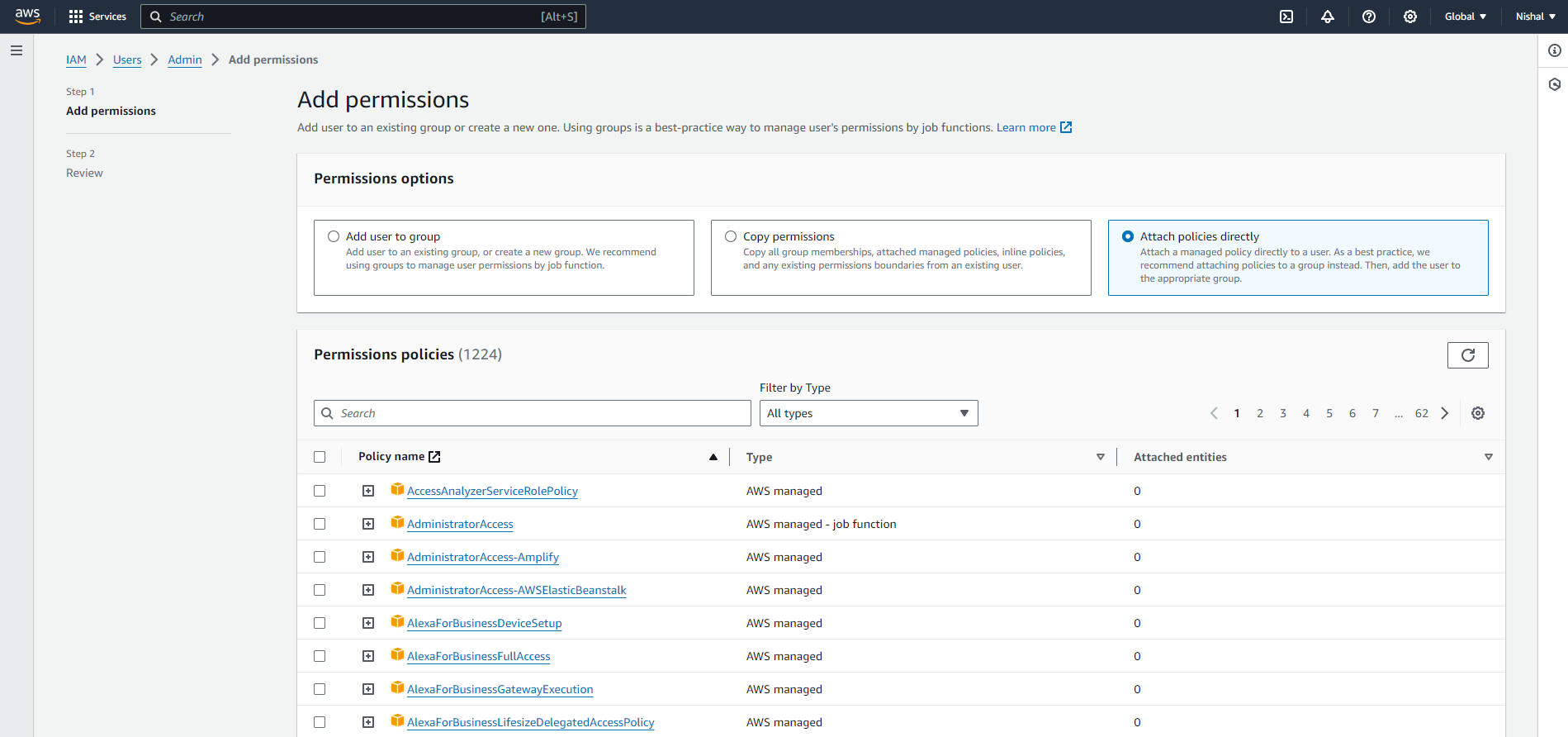


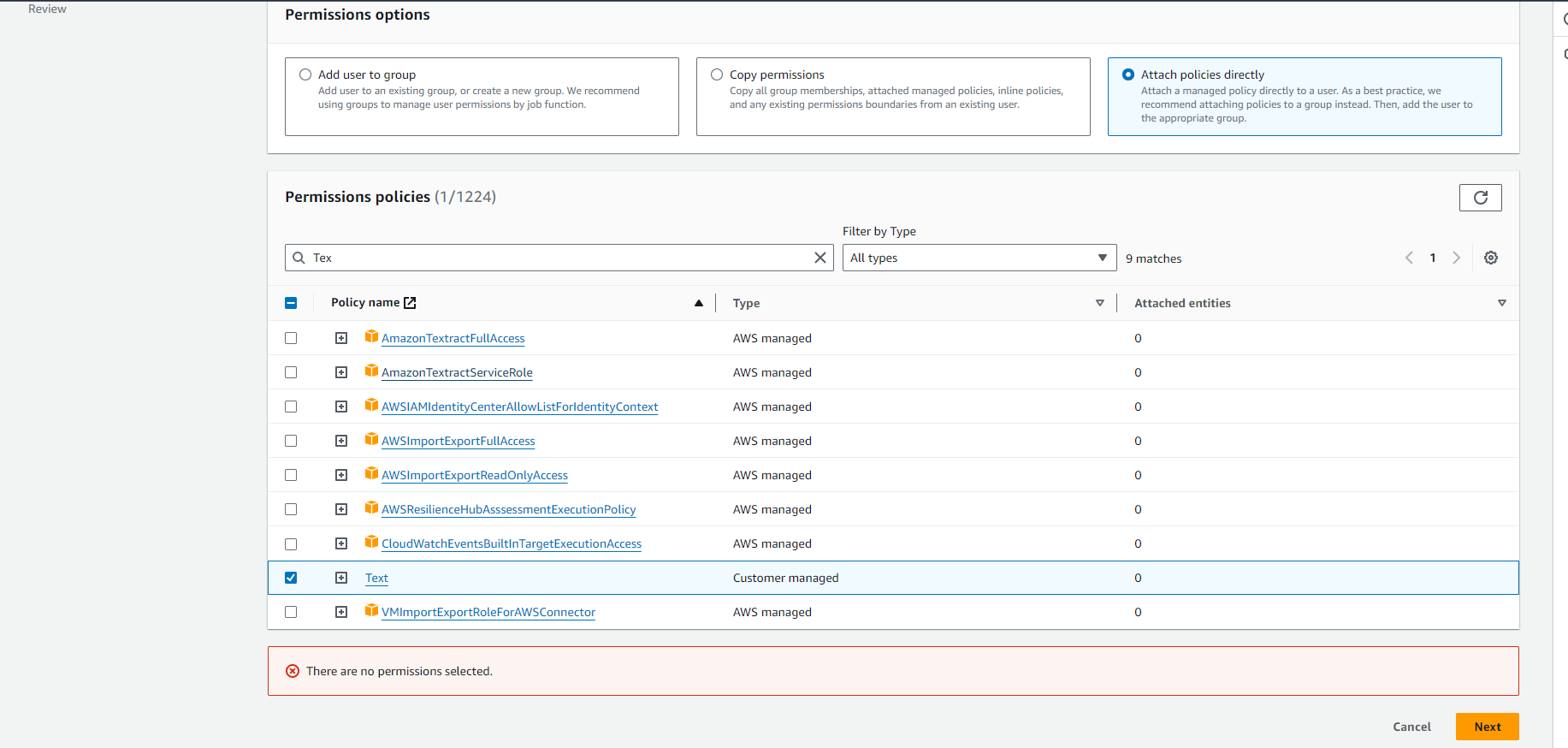


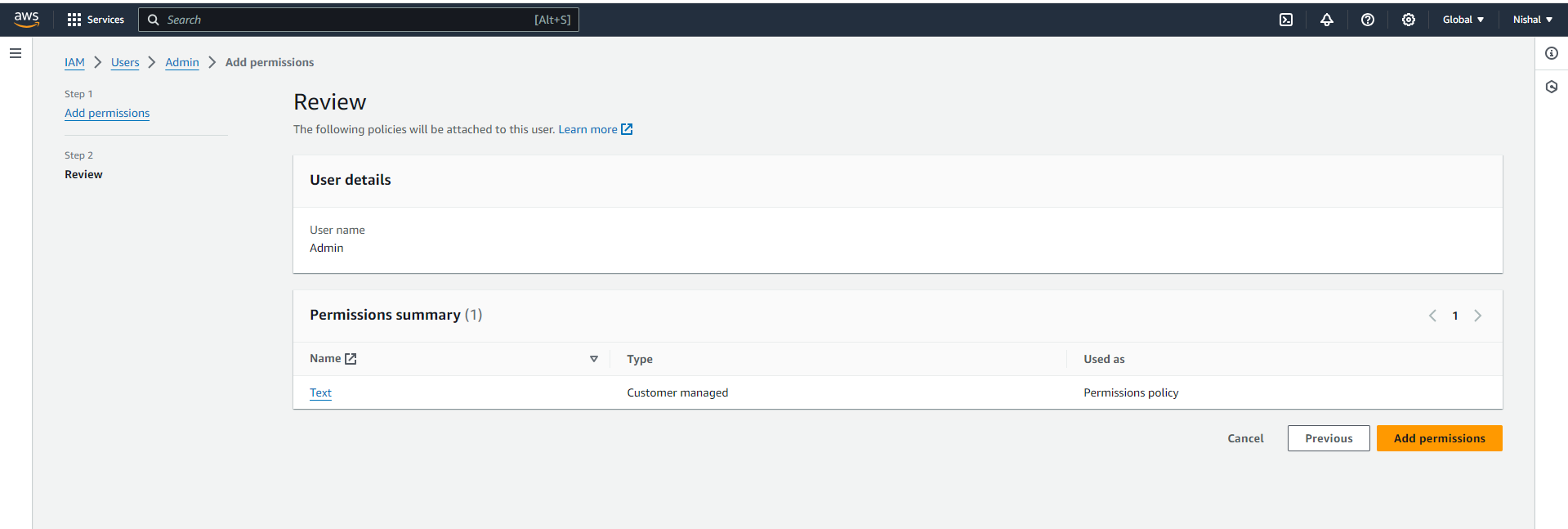


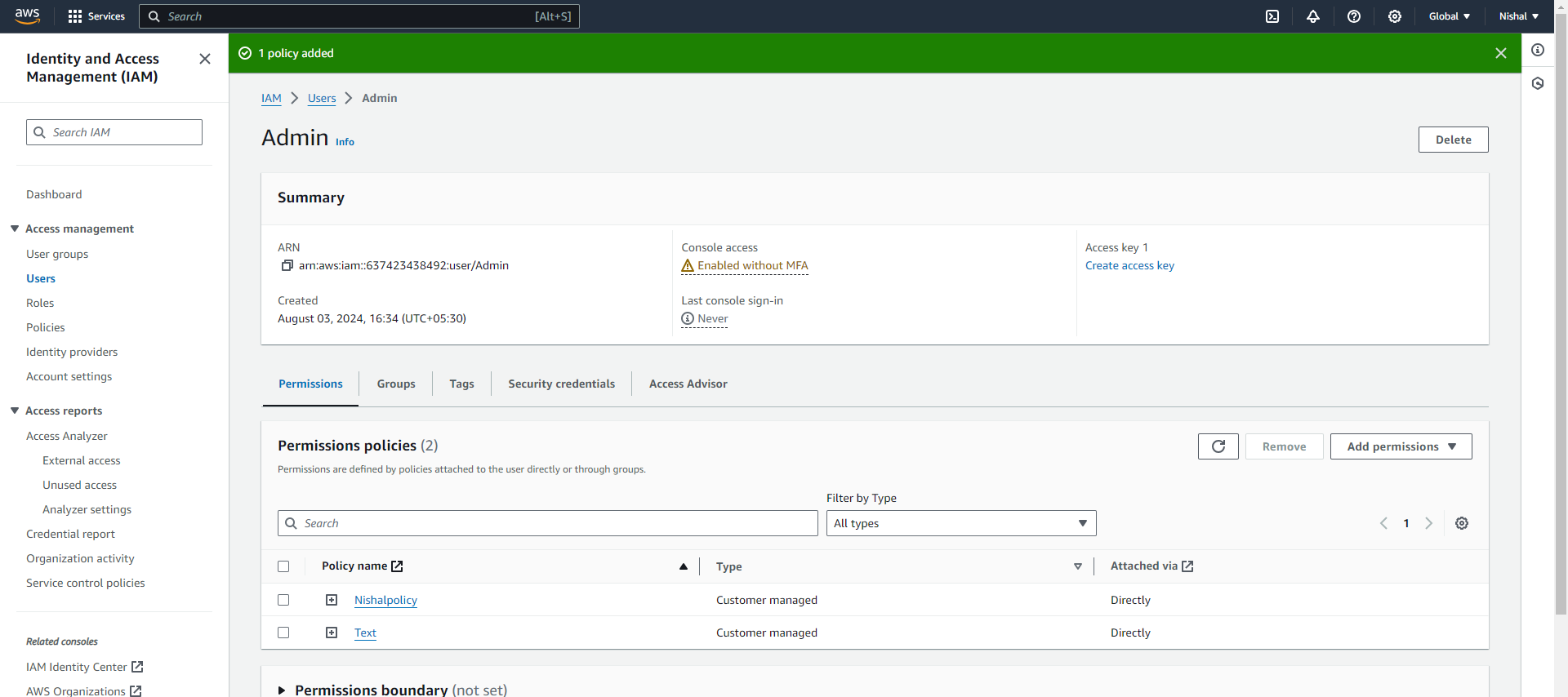












Select both the policies and click on remove your both the policies will be removed.

###### WRITE UP #########

1) Users and groups :

In AWS (Amazon Web Services), users and groups are part of Identity and Access Management (IAM), which is used to manage access to AWS services and resources securely.

**IAM Users**

* **Individual Access:** IAM users are created for individual people or applications that need access to AWS resources. Each user has a unique identity and can have its own security credentials (e.g., passwords, access keys).
* **Permissions:** By default, IAM users have no permissions. Permissions are granted by attaching policies to the user.
* **Security:** You can enable Multi-Factor Authentication (MFA) for users to enhance security.

**IAM Groups**

* **Group Management:** IAM groups are collections of IAM users. You can assign permissions to a group, and all users within that group inherit those permissions. This makes managing multiple users easier.
* **Permissions:** Like users, groups can have policies attached to them. When a policy is attached to a group, all users in that group get the permissions defined by the policy.

2) IAM :

AWS Identity and Access Management (IAM) is a service that helps you securely control access to AWS resources. IAM enables you to manage users, groups, and permissions to ensure that only authorized individuals or applications can access specific resources.

**Key Features of IAM**

1. **Users:**
   * **Individual Identities:** Each IAM user represents a single identity with specific credentials and permissions.
   * **Authentication:** Users can log in to AWS services using their credentials, such as passwords or access keys.
   * **Permissions:** Users can be granted permissions by attaching policies that specify what actions they can perform on which AWS resources.
2. **Groups:**
   * **Group Management:** Groups are collections of users. You can assign permissions to a group, and all users within that group will inherit those permissions.
   * **Simplified Administration:** Groups make it easier to manage permissions for multiple users.
3. **Roles:**
   * **Temporary Access:** IAM roles provide temporary access to AWS resources. They are used by applications, services, or users who need short-term access to specific resources.
   * **Cross-Account Access:** Roles can be used to grant access to users or services in other AWS accounts.
4. **Policies:**
   * **Access Control:** Policies are JSON documents that define what actions are allowed or denied for users, groups, or roles on specific resources.
   * **Granular Permissions:** Policies provide fine-grained control over who can do what in your AWS environment.
5. **Multi-Factor Authentication (MFA):**
   * **Enhanced Security:** MFA adds an extra layer of security by requiring users to provide a second form of authentication, such as a code from a mobile device.
6. **Access Control Lists (ACLs):**
   * **Resource-Level Permissions:** ACLs can be used to control access to individual resources, like S3 buckets, allowing or denying access to specific users or roles.

**Why Use IAM?**

* **Security:** IAM helps protect your AWS environment by ensuring that only authorized users have access to resources.
* **Control:** You have fine-grained control over permissions, allowing you to specify exactly what actions are allowed or denied.
* **Compliance:** IAM allows you to enforce organizational policies and meet regulatory requirements by managing access control.
* **Auditability:** IAM provides logging and monitoring features that help you track who is accessing your resources and when.

**Common Use Cases**

* **Managing Permissions for Employees:** Use IAM to create users and groups for your team, assign appropriate permissions, and enforce security best practices.
* **Granting Temporary Access:** Use IAM roles to grant temporary access to AWS resources for applications, services, or third-party vendors.
* **Securing Access Keys:** Use IAM to manage and rotate access keys securely to prevent unauthorized access.

IAM is a critical component of AWS security and governance, providing the tools to manage access and ensure that your cloud environment remains secure.

3) Role of IAM :

The role of IAM (Identity and Access Management) in AWS is crucial for ensuring the security, compliance, and efficient management of resources within an organization's cloud environment. Here are the key roles and responsibilities of IAM:

**1. Authentication:**

* **User Identity Verification:** IAM ensures that only authenticated users can access AWS resources. It manages credentials like passwords and access keys, and integrates with identity providers for federated access.
* **Multi-Factor Authentication (MFA):** IAM supports MFA, adding an extra layer of security by requiring users to provide a second form of verification.

**2. Authorization:**

* **Access Control:** IAM controls who can perform what actions on which resources. It uses policies (JSON documents) to define permissions, allowing or denying specific actions.
* **Granular Permissions:** IAM allows for fine-grained permissions, ensuring users and roles have just enough access to perform their jobs without over-provisioning.

**3. Resource Segmentation:**

* **Separation of Duties:** IAM helps in enforcing separation of duties by segmenting access based on user roles, teams, or project requirements, ensuring users only have access to the resources they need.

**4. Compliance and Governance:**

* **Policy Enforcement:** IAM enforces organizational security policies and standards, ensuring compliance with regulatory requirements and industry best practices.
* **Audit and Monitoring:** IAM integrates with AWS CloudTrail and other monitoring tools to track access and actions performed by users, providing a comprehensive audit trail for security reviews.

**5. Role-Based Access Control (RBAC):**

* **IAM Roles:** IAM roles provide temporary access to AWS resources. They are used by services, applications, or other AWS accounts, allowing them to perform tasks without needing permanent credentials.
* **Cross-Account Access:** IAM roles facilitate secure access between different AWS accounts, enabling organizations to share resources without compromising security.

**6. Managing Temporary and Federated Access:**

* **Temporary Security Credentials:** IAM can provide temporary credentials for users or applications, which are valid for a limited duration, reducing the risk of long-term exposure.
* **Federated Access:** IAM supports identity federation, allowing users to access AWS resources using their existing corporate credentials through identity providers like Active Directory.

**7. Simplifying User Management:**

* **User and Group Management:** IAM simplifies the administration of user permissions by allowing the grouping of users and assigning policies to groups instead of individual users.
* **Automated User Onboarding/Offboarding:** IAM can be integrated into workflows for automated provisioning and de-provisioning of user access as employees join or leave the organization.

**8. Protecting Resources:**

* **Securing Sensitive Resources:** IAM helps protect sensitive AWS resources by ensuring only authorized users and services can access them.
* **Least Privilege Principle:** By adhering to the principle of least privilege, IAM ensures that users have only the necessary permissions to perform their tasks, minimizing potential security risks.

**9. Scalability:**

* **Managing Access at Scale:** IAM can manage access for thousands of users, groups, and roles across large, complex AWS environments, making it suitable for both small and large organizations.

**10. Integration with Other AWS Services:**

* **Seamless Integration:** IAM integrates with nearly every AWS service, allowing consistent and centralized management of access controls across the entire AWS ecosystem.

Overall, IAM is fundamental in ensuring that AWS resources are accessed securely and appropriately, playing a pivotal role in an organization’s cloud security strategy.