Salita Dbritto

Roll no- A010

Sap id- 86062300036

CLOUD COMPUTING PRACTICAL-3 [IAM]

WRITE UP –



1] Users and Groups

In AWS (Amazon Web Services), IAM (Identity and Access Management) allows you to control access to AWS services and resources securely.

**IAM Users**

**Definition:** An IAM user is an entity that you create in AWS to represent a person or application that interacts with AWS services. Each IAM user has a unique identity and credentials (such as a username and password or access keys).

**Key Features:**

* **Credentials:** Users can have different types of credentials, including access keys (for programmatic access) and a password (for AWS Management Console access).
* **Permissions:** Permissions determine what actions the user can perform on AWS resources. These are managed through policies attached to the user.
* **Individual Control:** You can assign specific permissions to individual users based on their needs.

**Use Cases:**

* **For Individuals:** A developer, admin, or any user who needs direct access to AWS resources.
* **For Applications:** Applications or services that need to interact with AWS APIs can use IAM users with access keys.

**IAM Groups**

**Definition:** An IAM group is a collection of IAM users. You use groups to manage permissions for multiple users at once.

**Key Features:**

* **Simplified Management:** By attaching policies to a group, all users in that group automatically inherit those permissions. This simplifies management compared to managing permissions for each user individually.
* **Organizational Structure:** Groups help you organize users by their roles or responsibilities (e.g., Developers, Admins, ReadOnlyUsers).

**Use Cases:**

* **Role-Based Access Control:** Assign users to groups based on their job function or role within the organization.
* **Policy Management:** Apply policies to groups rather than individual users to streamline permission management.

2] IAM

AWS Identity and Access Management (IAM) is a web service that enables you to manage access to AWS services and resources securely. IAM allows you to control who can access your AWS resources and what actions they can perform, providing a robust framework for managing permissions and enhancing security.

**Key Components:**

* **IAM Users:**
  + **Definition:** An IAM user is an identity created for an individual person or application that requires access to AWS resources.
  + **Credentials:** Users can have access keys for API requests or passwords for console access.
  + **Permissions:** Permissions are granted through policies that are attached directly to the user or through IAM groups.
* **IAM Groups:**
  + **Definition:** A group is a collection of IAM users managed as a unit.
  + **Policy Management:** Policies attached to a group are automatically applied to all users within that group, simplifying permission management.
  + **Use Case:** Ideal for assigning permissions based on job roles (e.g., Developers, Admins).
* **IAM Roles:**
  + **Definition:** A role is an IAM entity that defines a set of permissions for making AWS service requests. Unlike users, roles are not associated with a specific person but are intended to be assumed by anyone or anything needing access.
  + **Use Case:** Commonly used for cross-account access, temporary permissions, or AWS services assuming roles to interact with other AWS resources.
* **IAM Policies:**
  + **Definition:** Policies are documents that define permissions in JSON format. Policies specify what actions are allowed or denied on specific resources.
  + **Types:** Managed policies (AWS-provided or customer-managed) and inline policies (embedded directly into a user, group, or role).
* **IAM Permissions:**
  + **Definition:** Permissions specify what actions are allowed on which AWS resources. They are defined within policies and attached to IAM users, groups, or roles.
  + **Granularity:** Permissions can be very granular, controlling access down to specific resources and actions.
* **IAM Best Practices:**
  + **Principle of Least Privilege:** Grant users and roles only the permissions they need to perform their tasks.
  + **Use Roles for EC2 Instances:** Avoid using IAM user credentials in applications running on EC2 instances. Instead, assign roles to the instances.
  + **Enable Multi-Factor Authentication (MFA):** Add an additional layer of security for IAM users, especially those with elevated privileges.
  + **Regularly Review Permissions:** Periodically audit IAM policies and access controls to ensure they comply with your current organizational needs and security policies.

3] Role of IAM

AWS Identity and Access Management (IAM) plays several crucial roles in managing access to AWS resources.

**1. Identity Management**

* **User Creation:** IAM allows you to create and manage individual user identities within AWS.
* **Credentials Management:** Users can have different types of credentials such as passwords for console access or access keys for API calls.
* **Authentication:** Ensures that only authenticated users and services can access AWS resources.

**2. Access Control**

* **Permissions Management:** IAM enables you to define and manage permissions through policies. You can specify what actions are allowed or denied on various AWS resources.
* **Role-Based Access Control:** Users and services are assigned roles with specific permissions to streamline access management and enforce the principle of least privilege.

**3. Security**

* **Granular Access:** Provides fine-grained control over who can access what resources and what actions they can perform.
* **Multi-Factor Authentication (MFA):** Adds an extra layer of security by requiring additional authentication factors beyond just passwords.
* **Policy Enforcement:** Ensures that permissions are enforced consistently across AWS services.

**4. Resource Management**

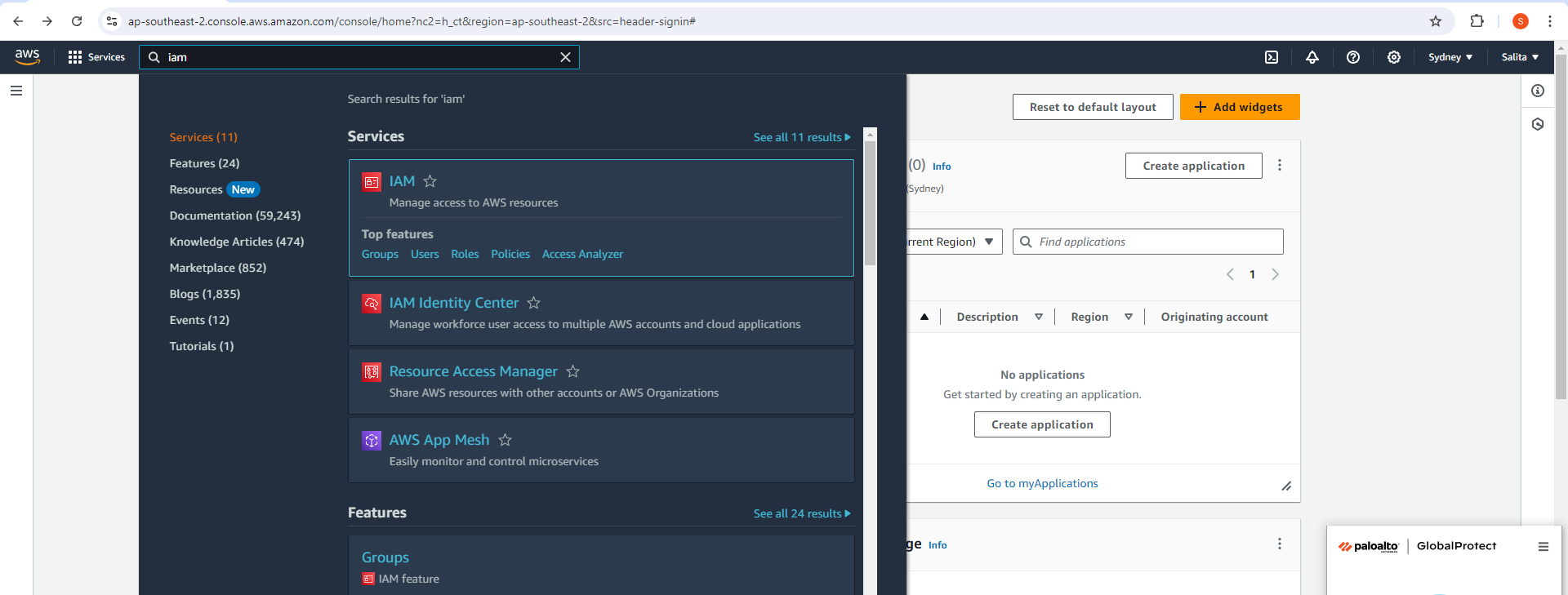
* **Group Management:** IAM groups allow you to manage permissions for multiple users collectively, simplifying administrative tasks.
* **Role Management:** Roles provide temporary or specific access to AWS resources and can be assumed by users, applications, or AWS services, facilitating secure resource management.

**5. Audit and Compliance**

* **Logging and Monitoring:** IAM integrates with AWS Cloud Trail to log API calls made to IAM, allowing you to monitor and audit access and changes.
* **Policy Review:** Regular review of IAM policies and permissions helps ensure compliance with organizational and regulatory requirements.

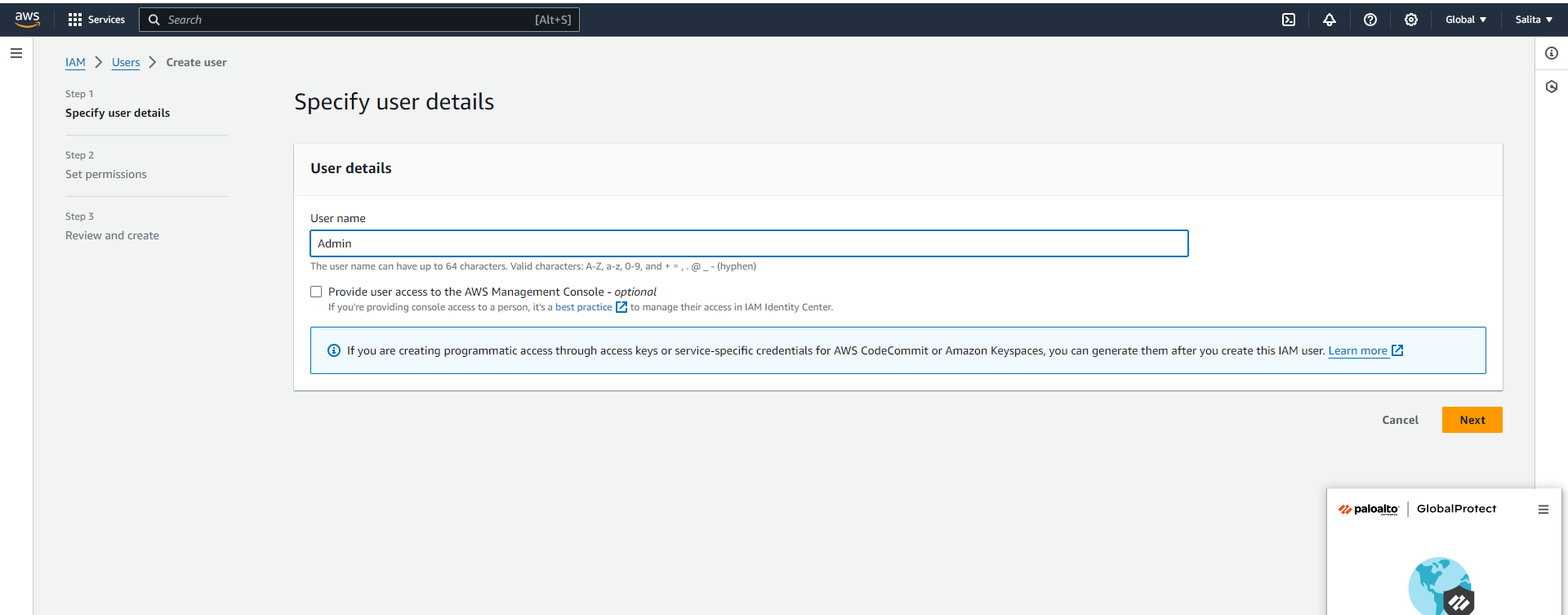
STEPS

In search bar, type “IAM” and select IAM dropdown.

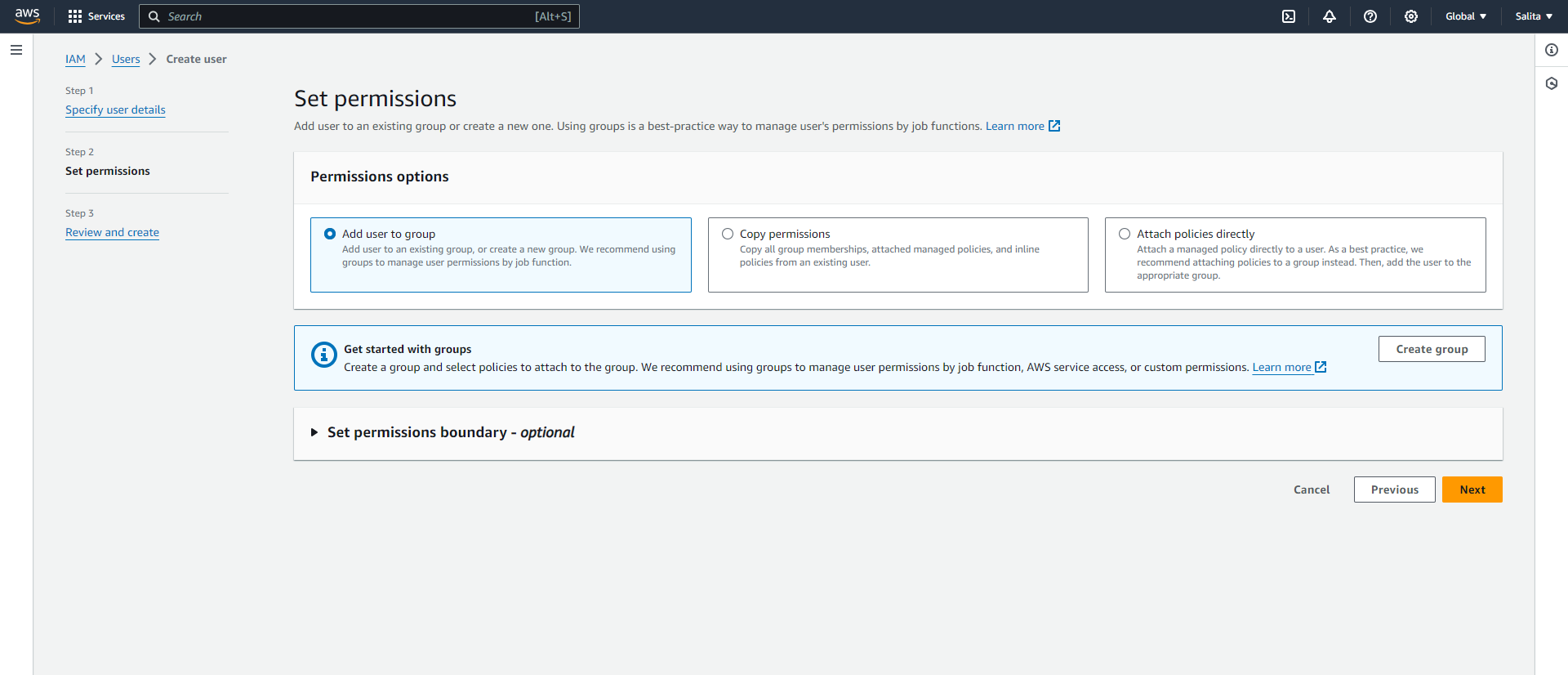


In the IAM dashboard, navigate to “Users” and click “Add User”.

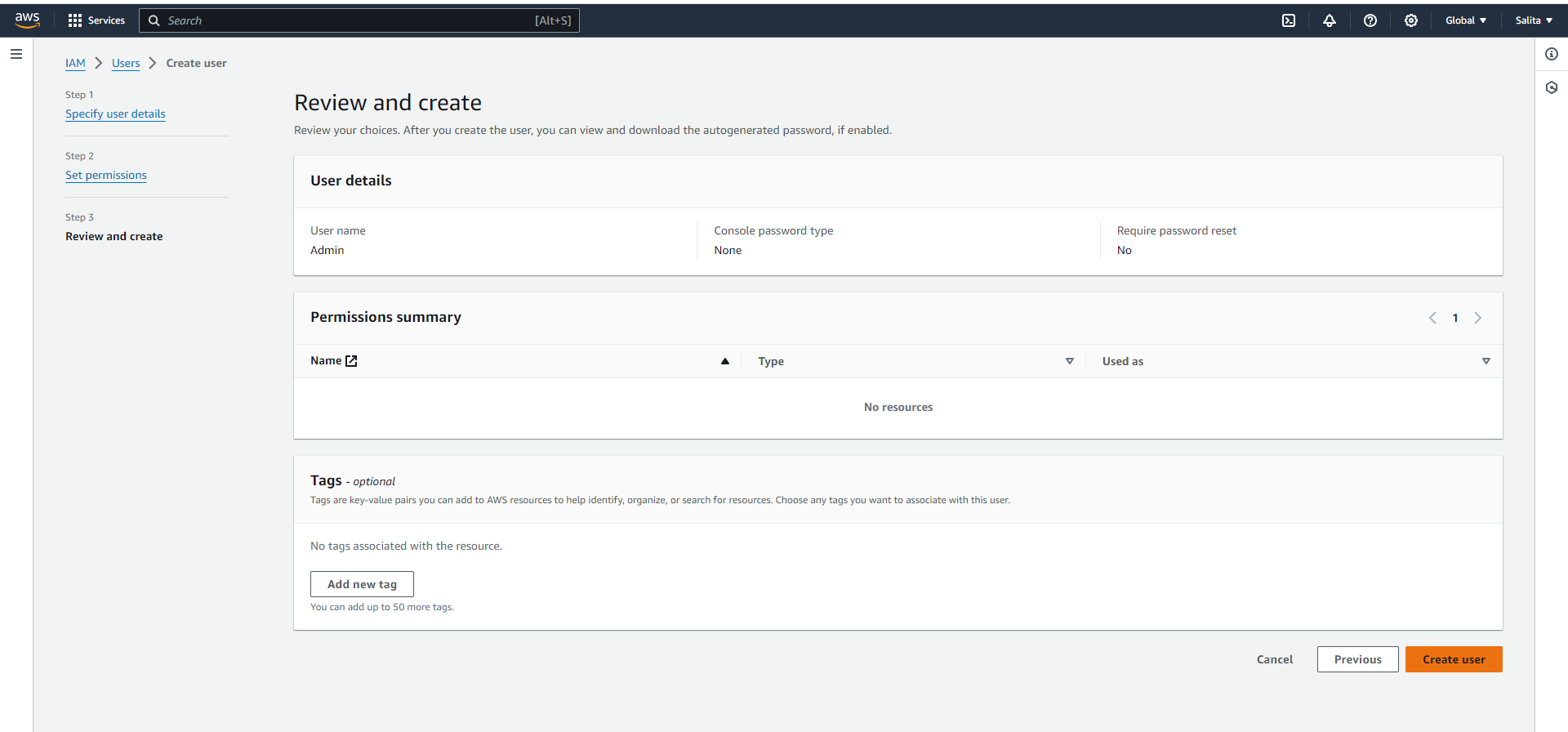
Specify the user name and Next.



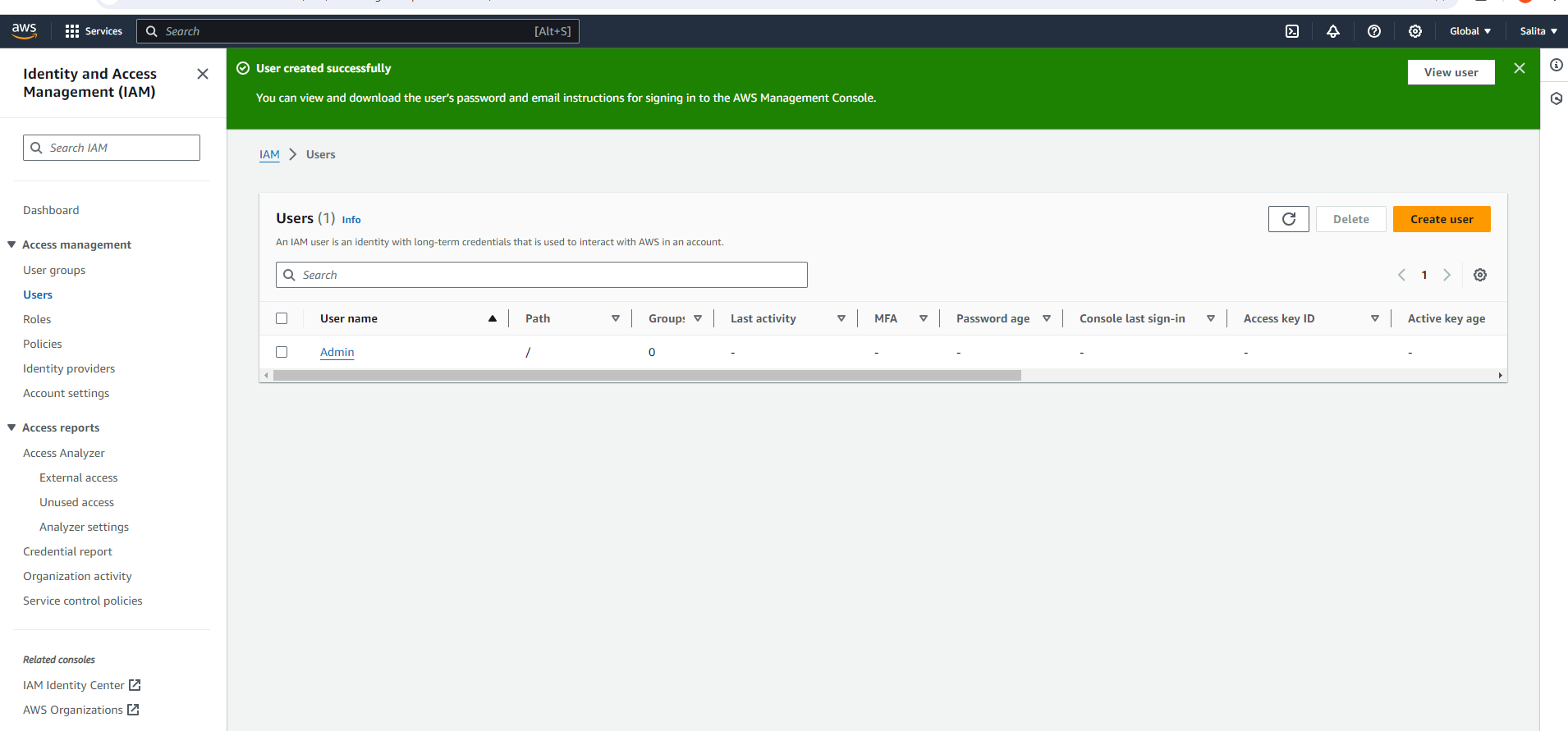
In Set permissions, Select “Add user to group” and Select Next.



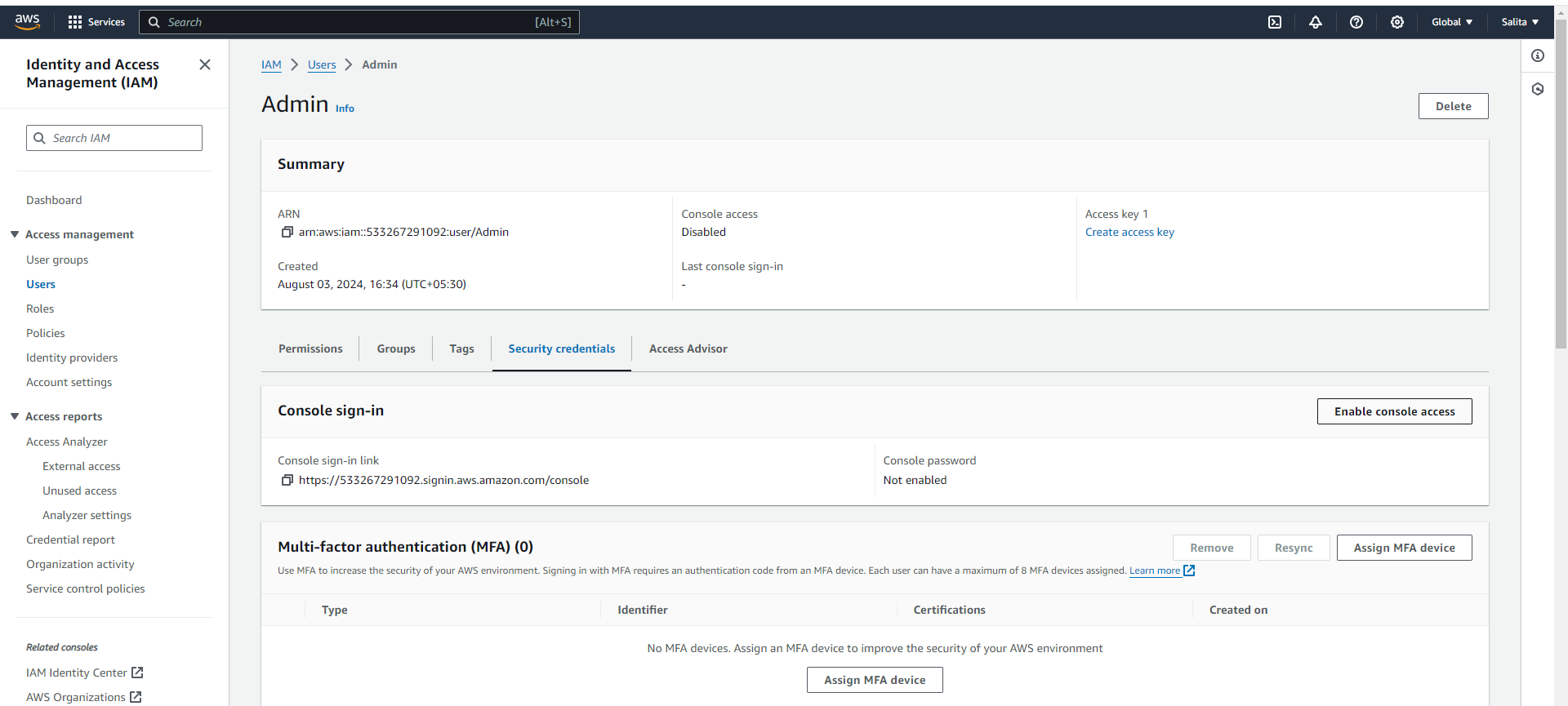
Create User.

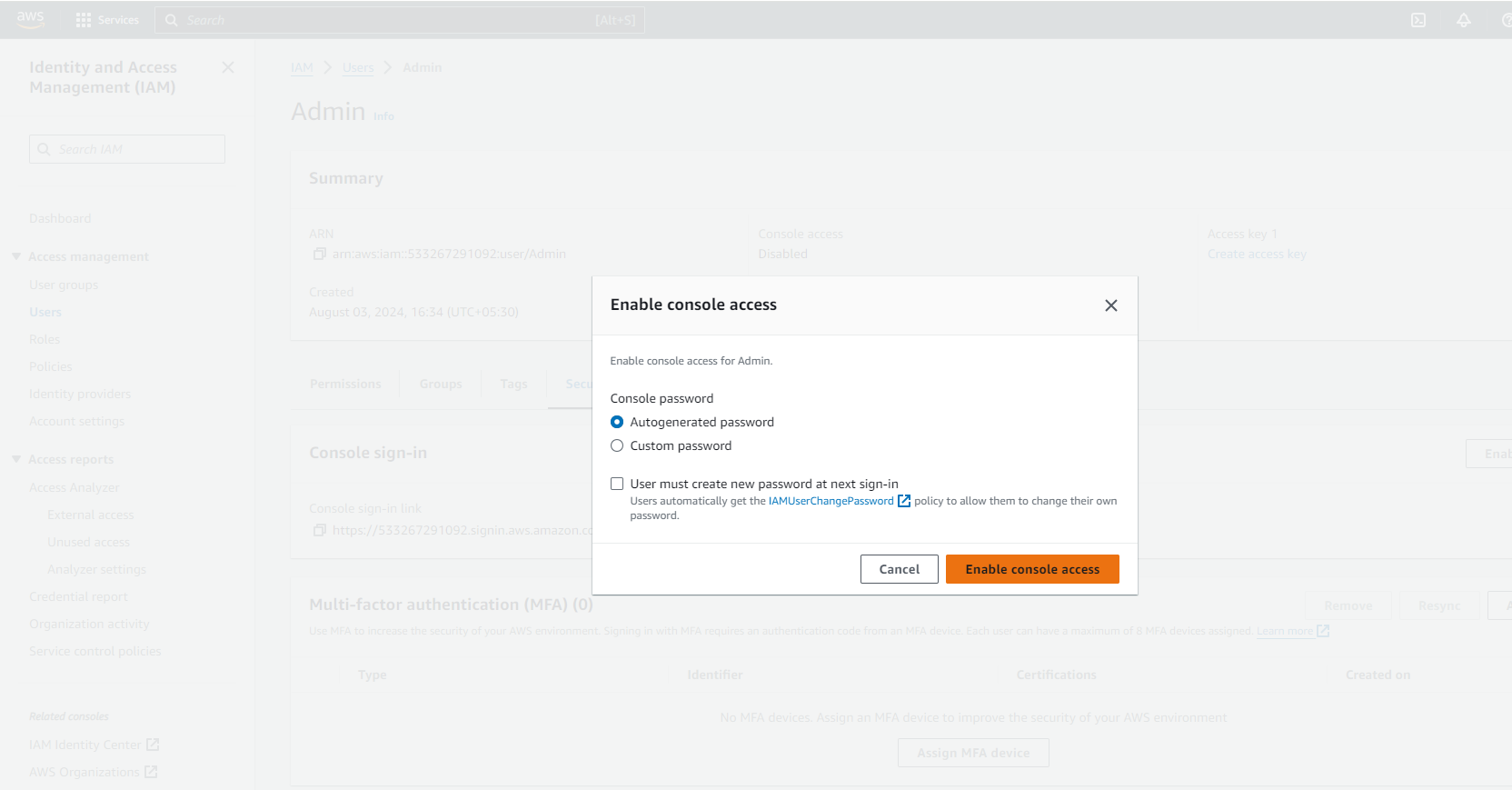


The User is successfully created.

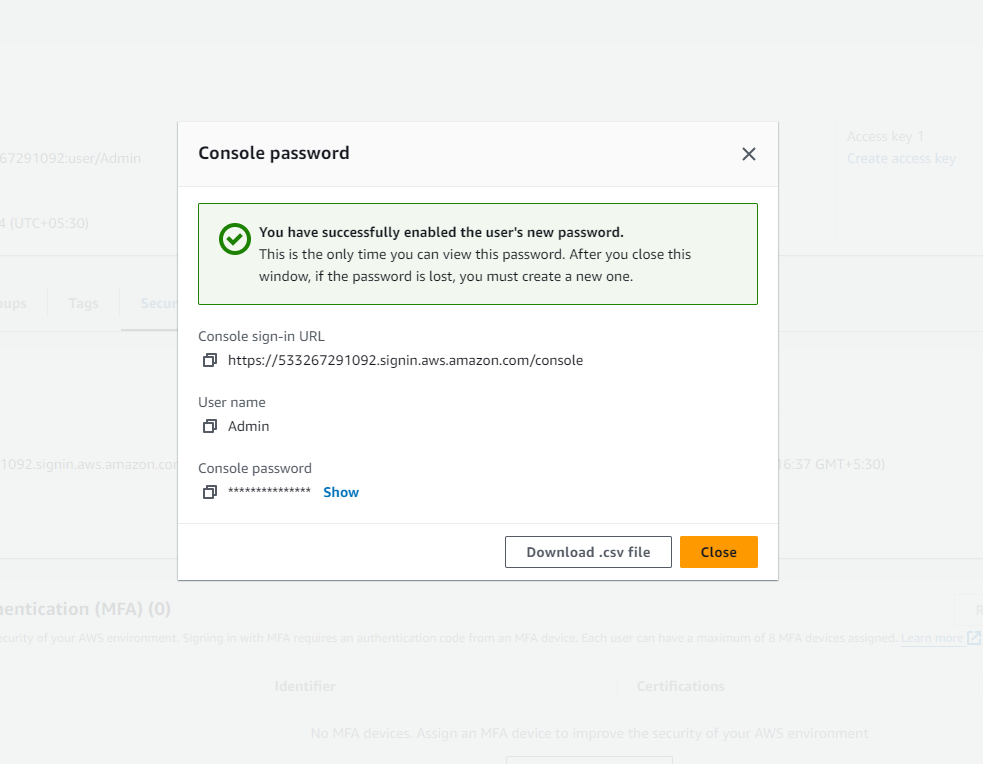


Select the user created and under Security Credentials click “Enable control access”.



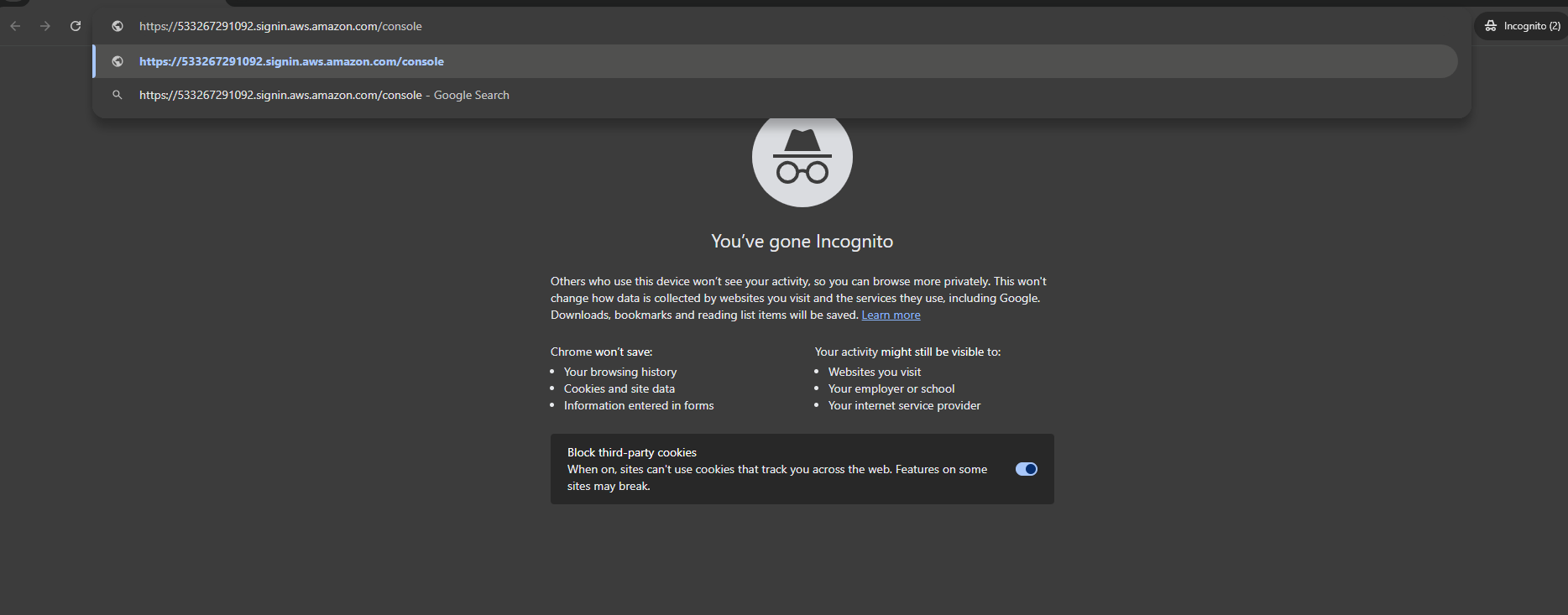


Copy the URL and download csv file for the password.

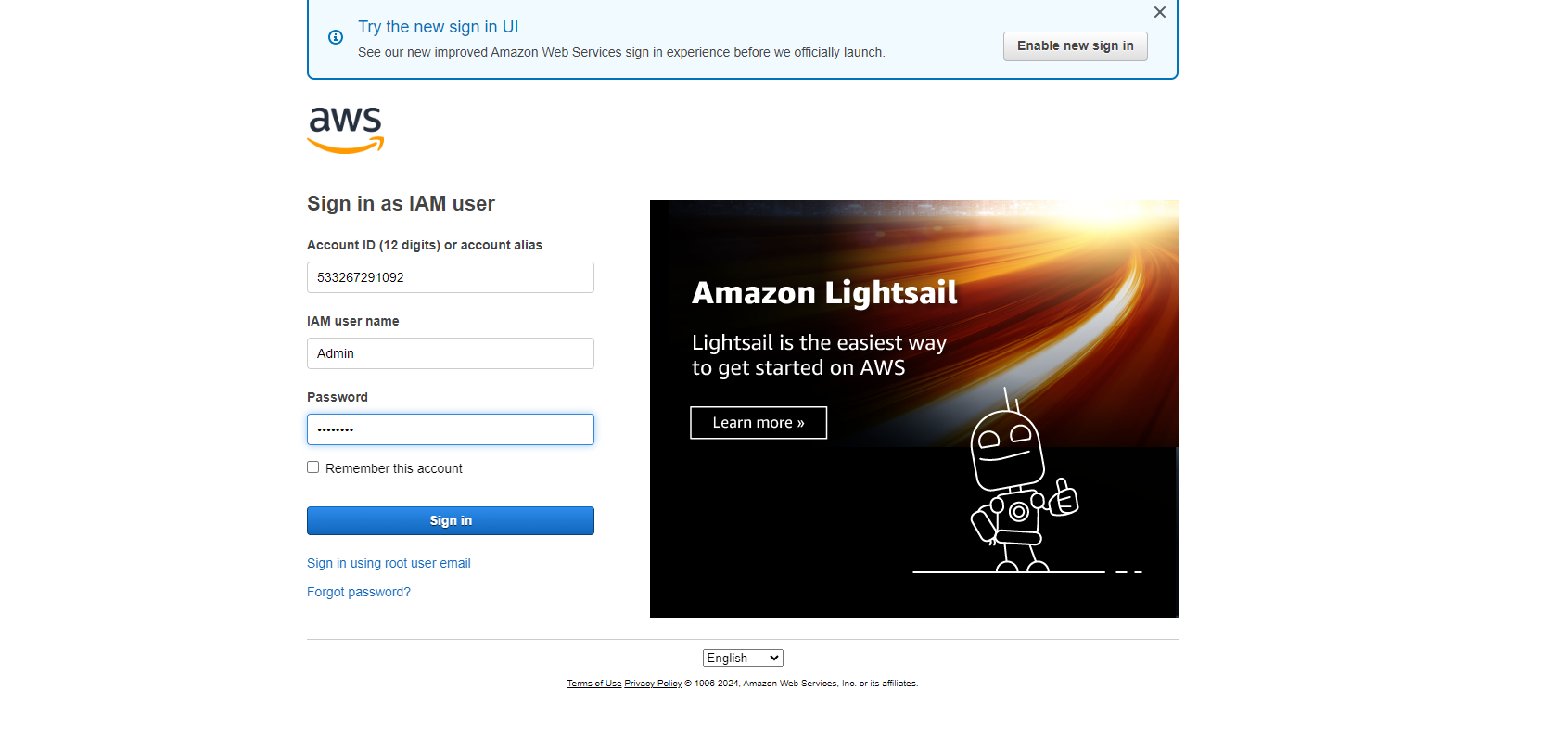




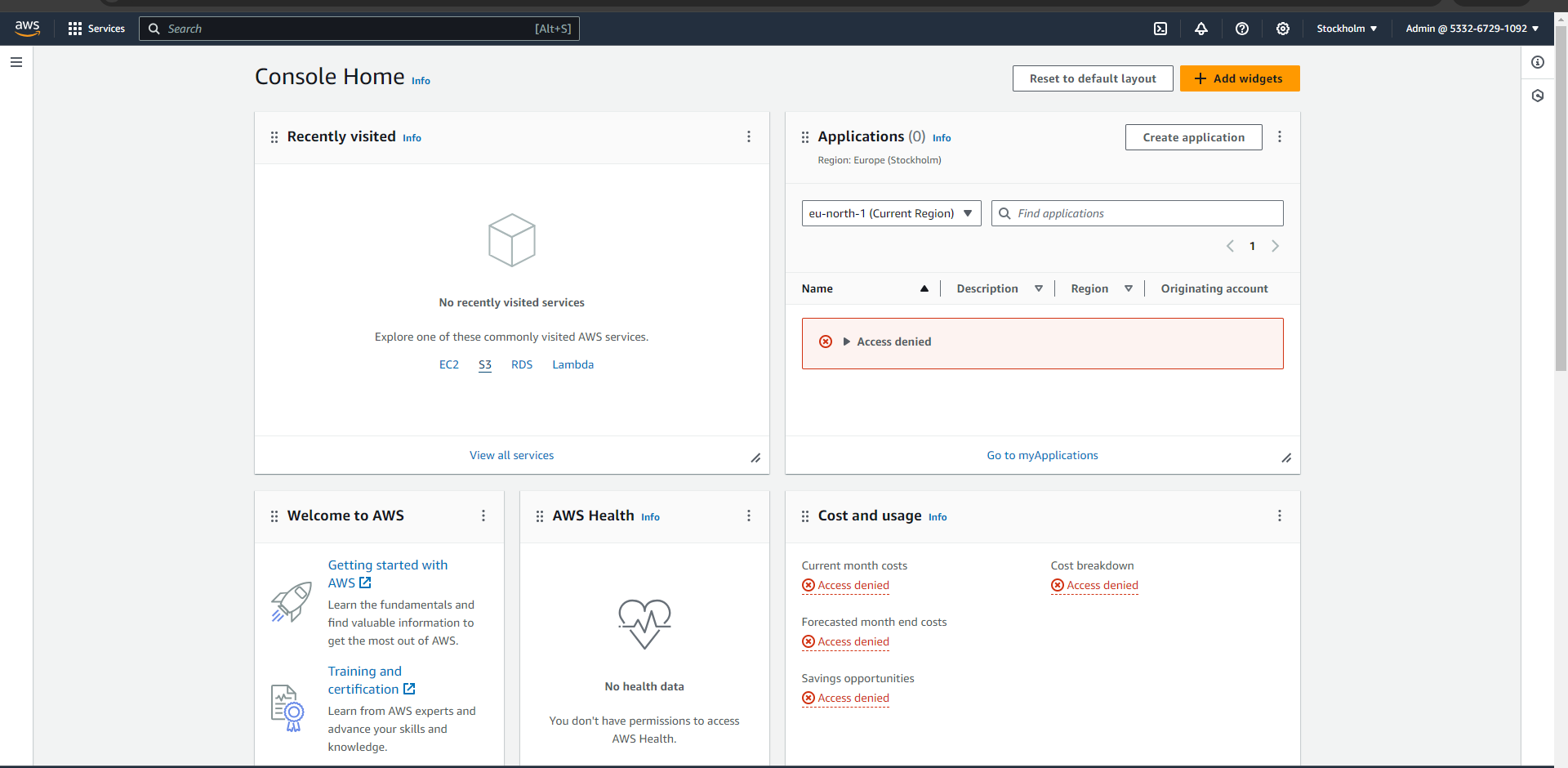
Now, Open the “Incognito Mode” and paste the URL.



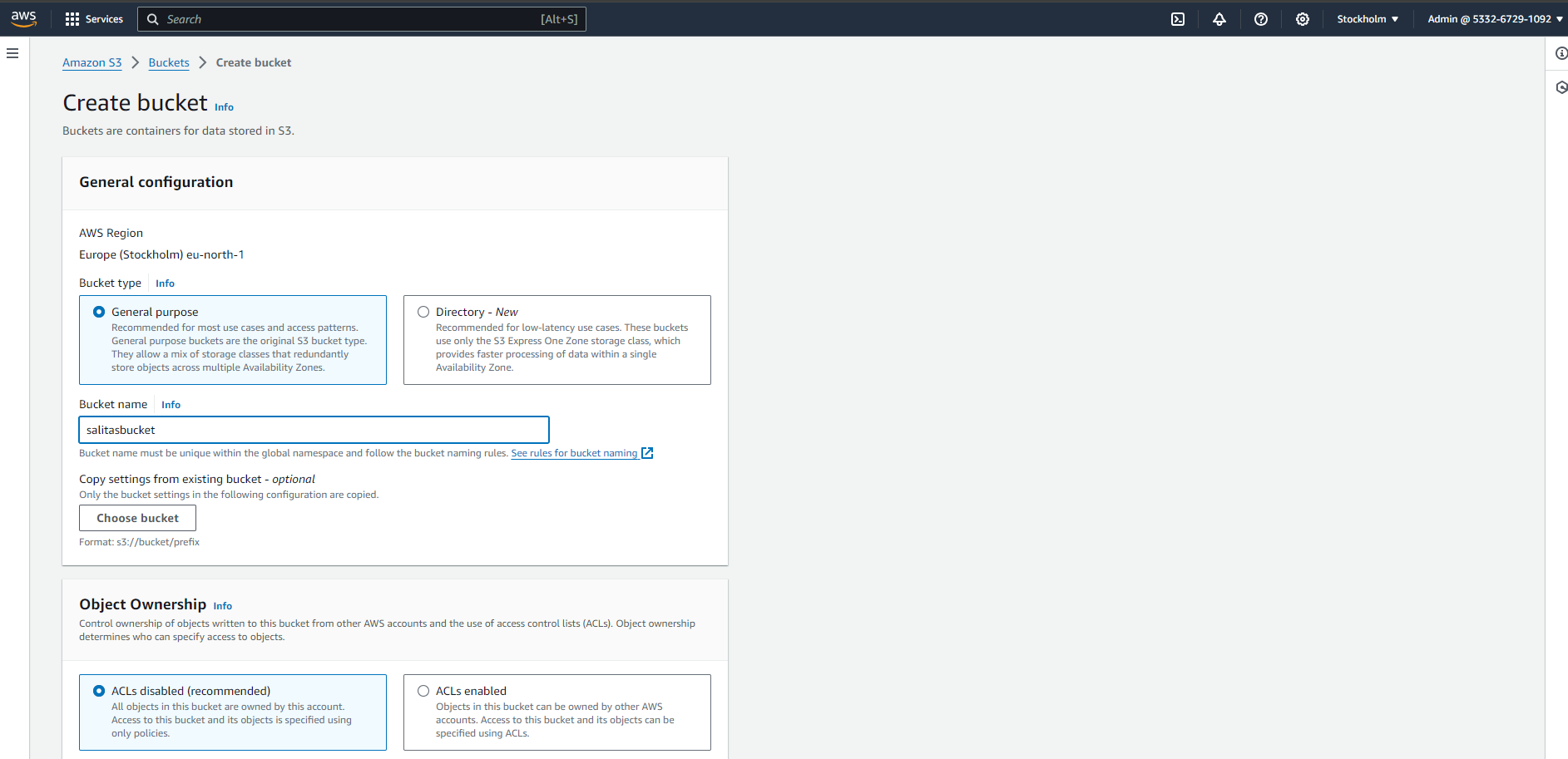
Sign in the Aws account with your root username and password downloaded earlier.



The Console opens and you can access any of the services. For instance we select the S3 service.

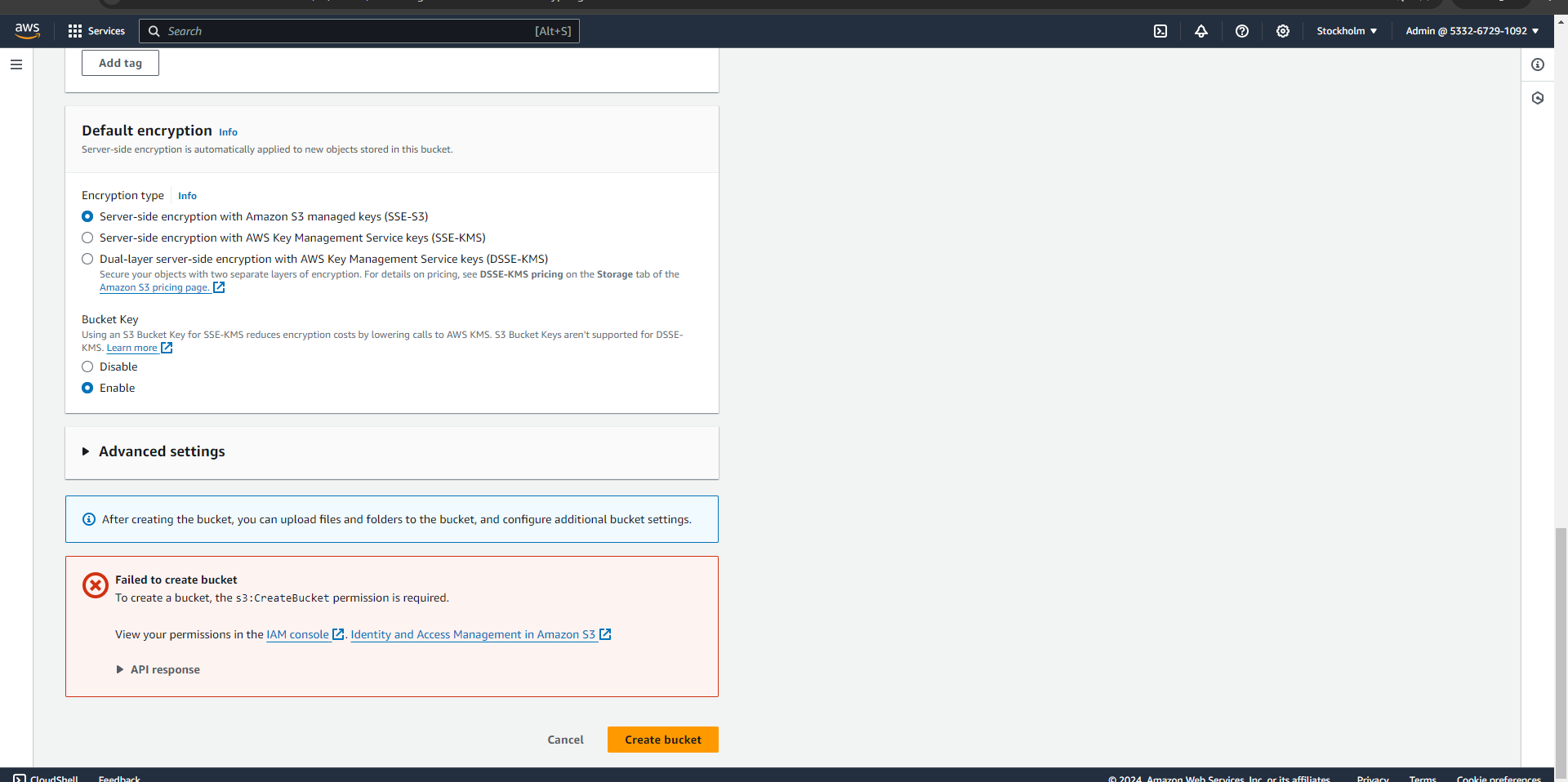


Create bucket.

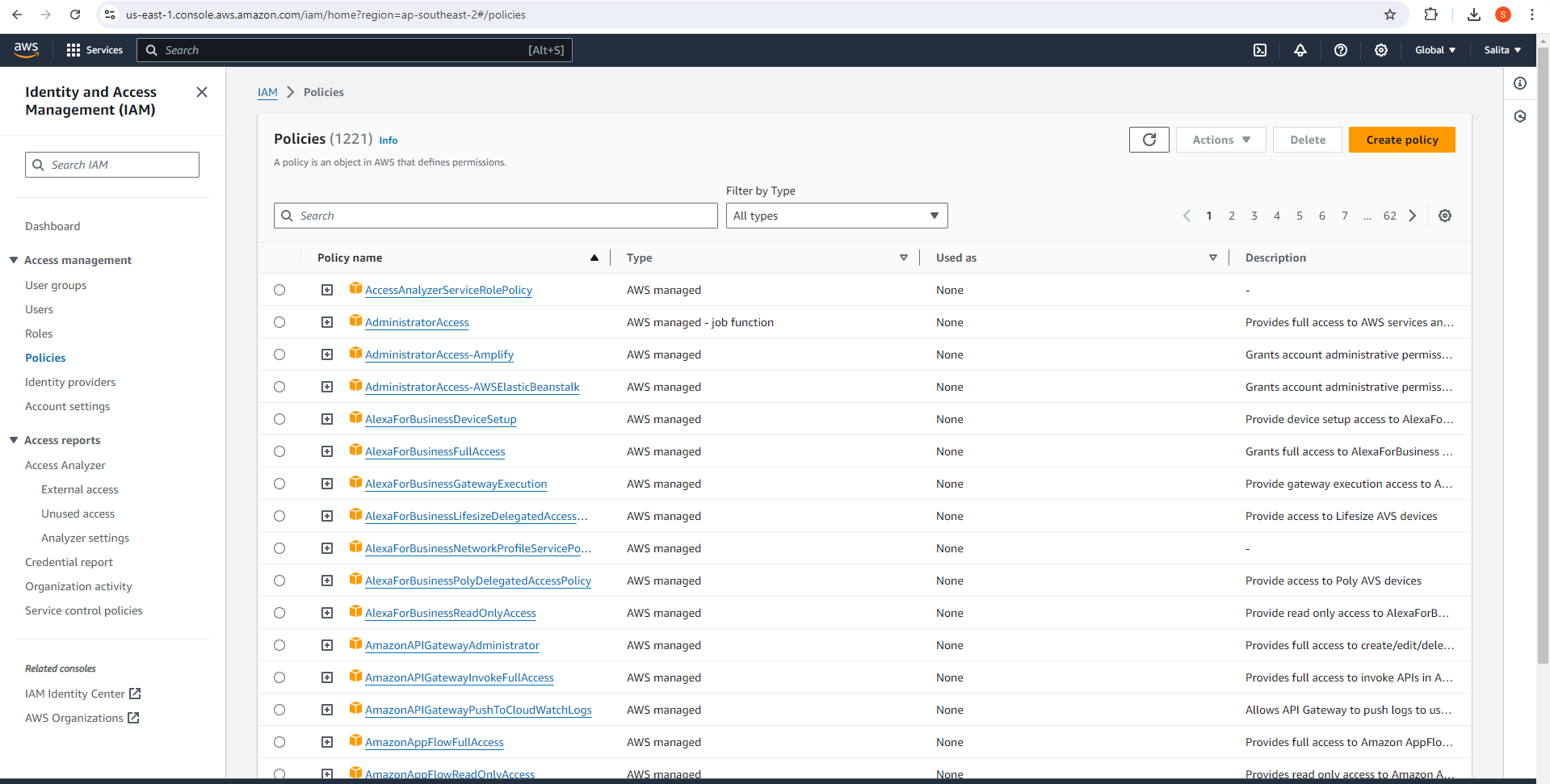




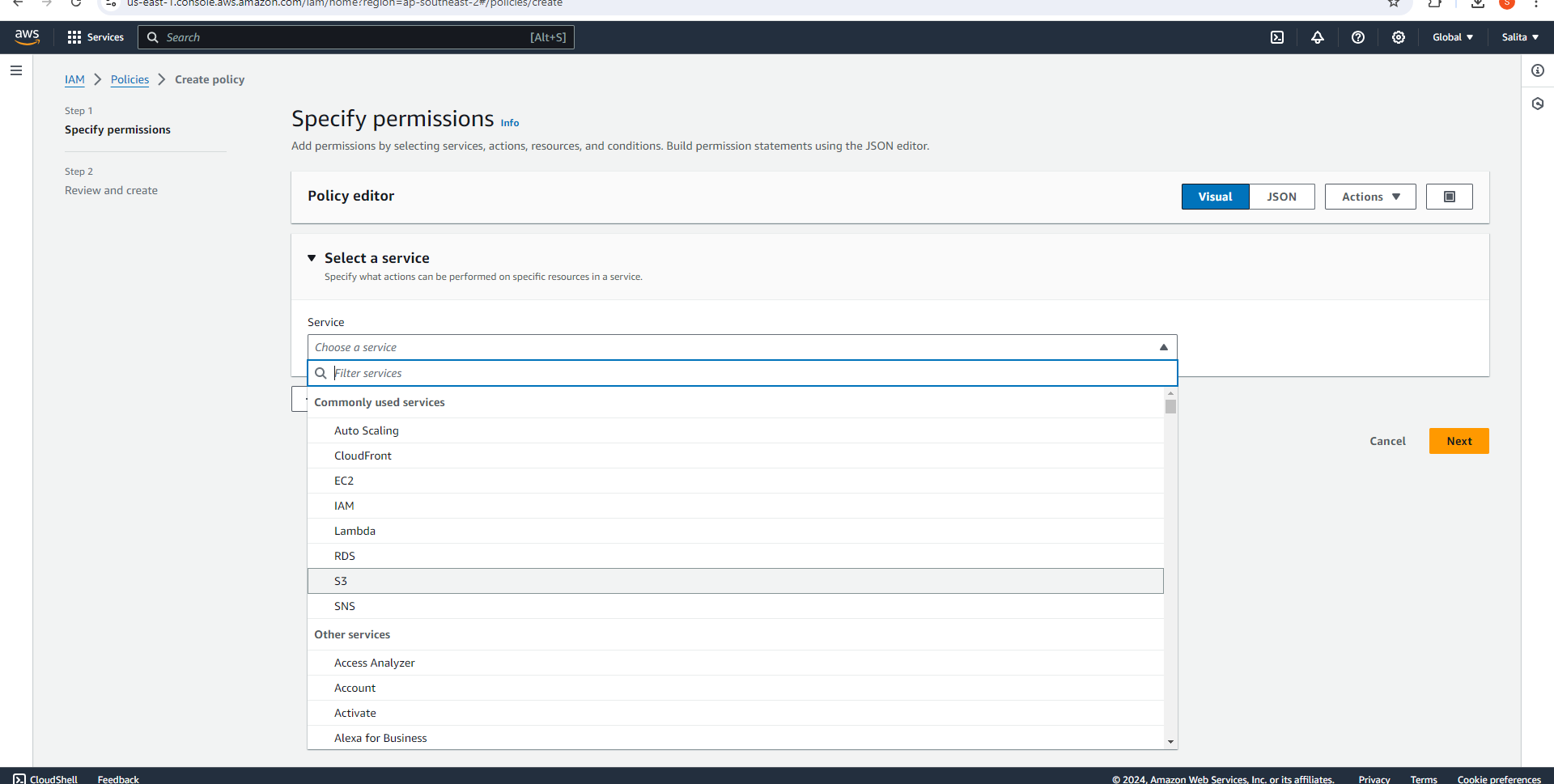
The incognito mode fails to create the bucket.



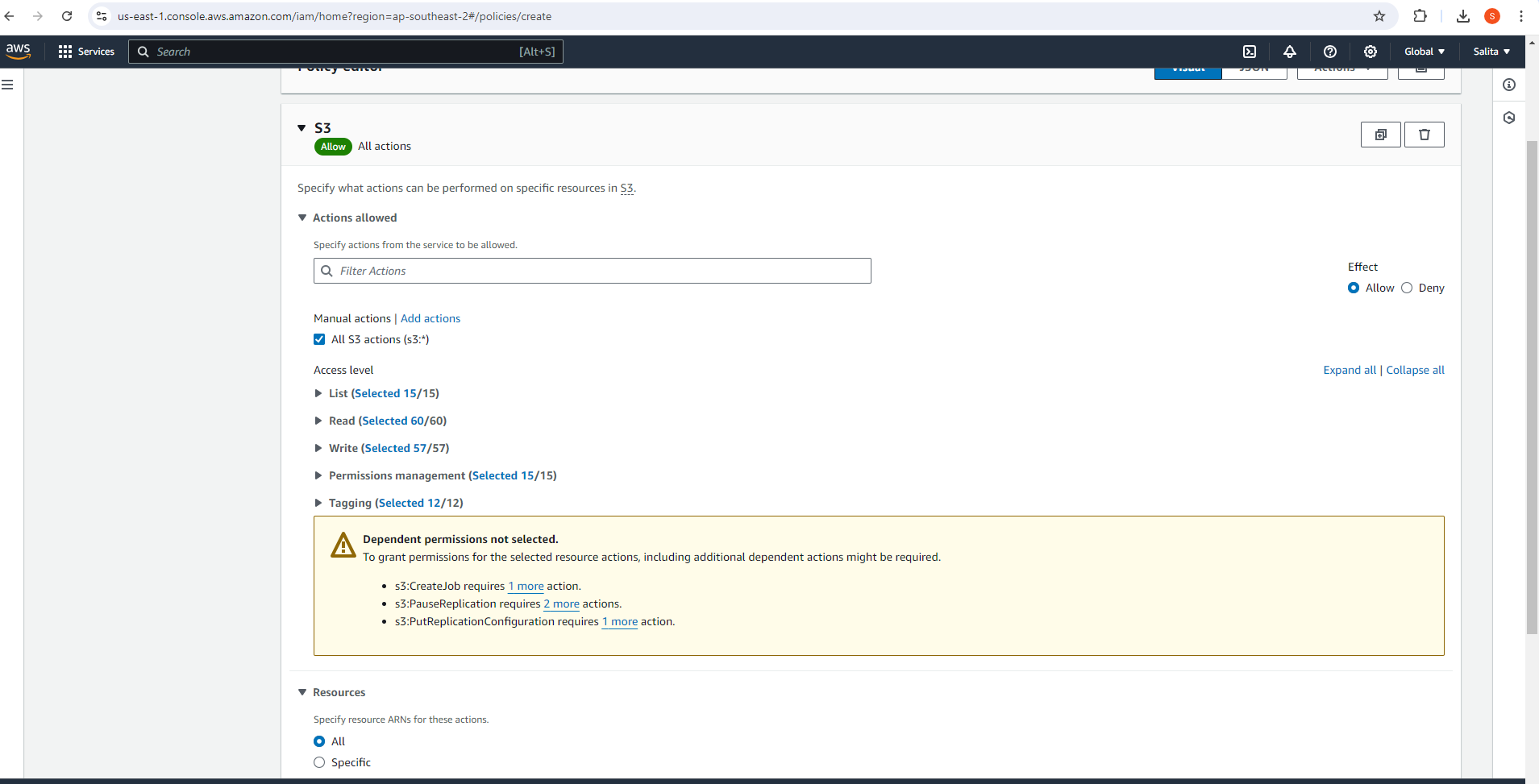
Hence, Come back to the Root user and in the IAM dashboard open the “Policies” in the left hand menu. Create Policy.



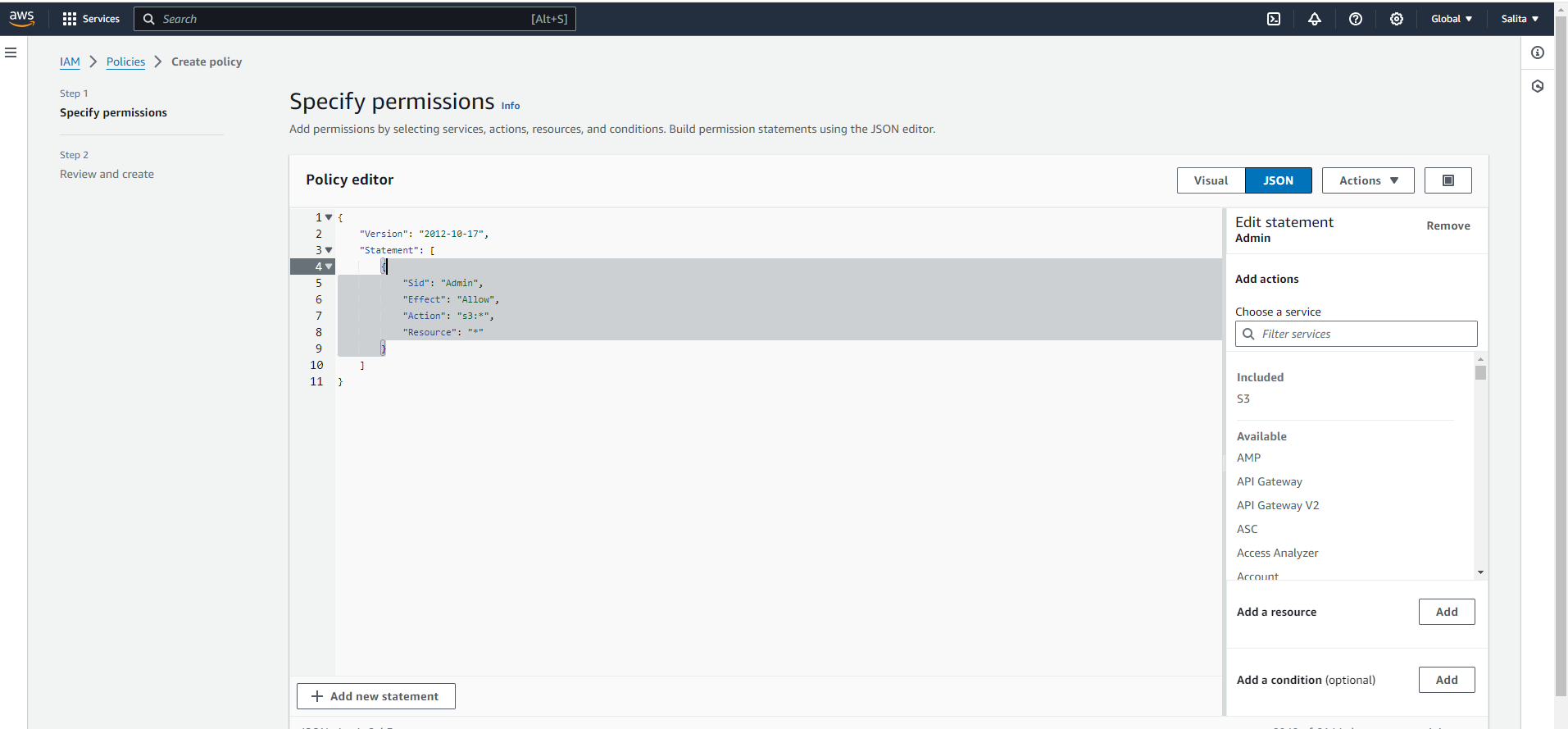
In Specify permissions, and Visual editor mode choose the Aws service “S3”.



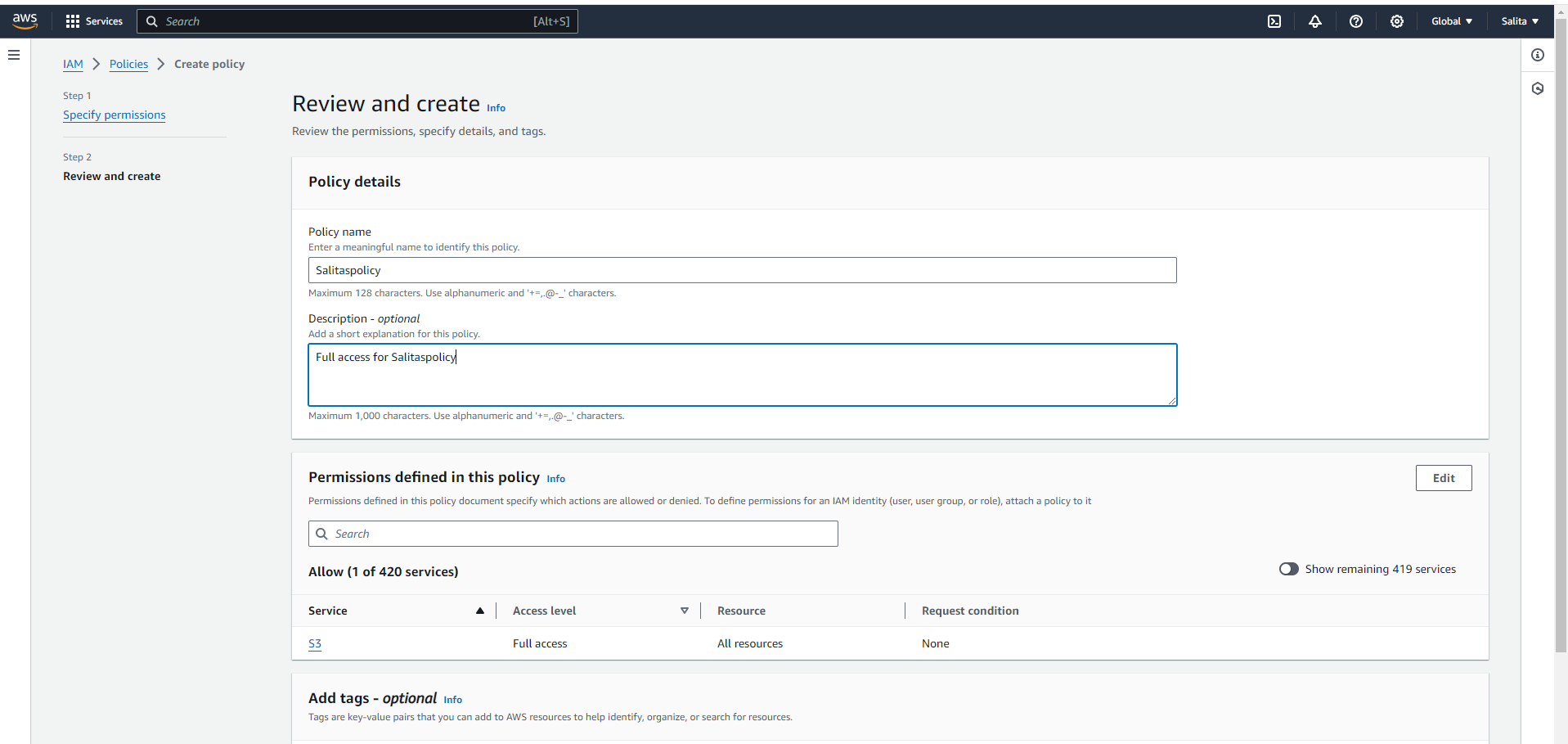
Select appropriate ‘Actions’ and ‘Resources’.



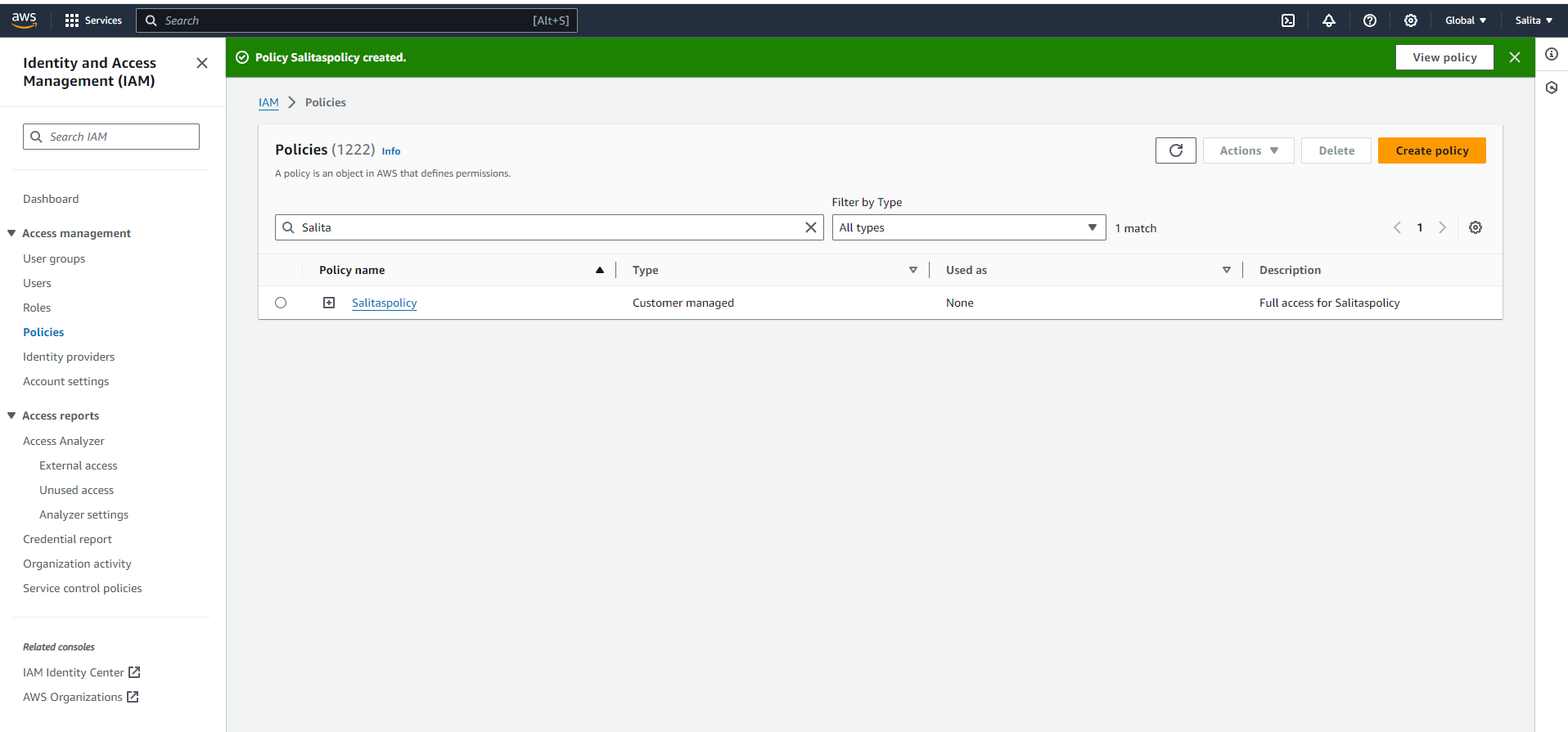
In the ‘JSON editor’, directly enter the User name.



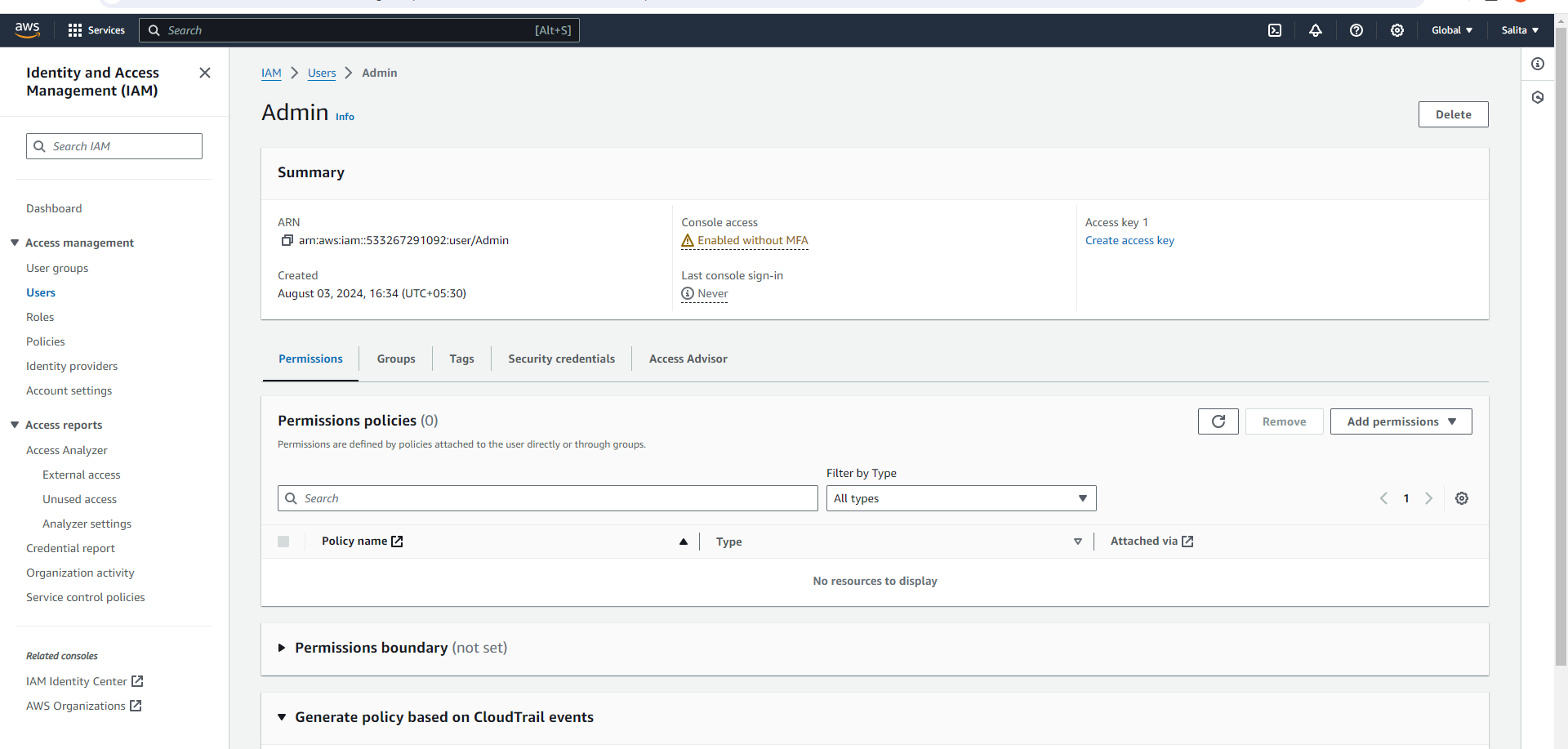
Name and Describe the Policy.



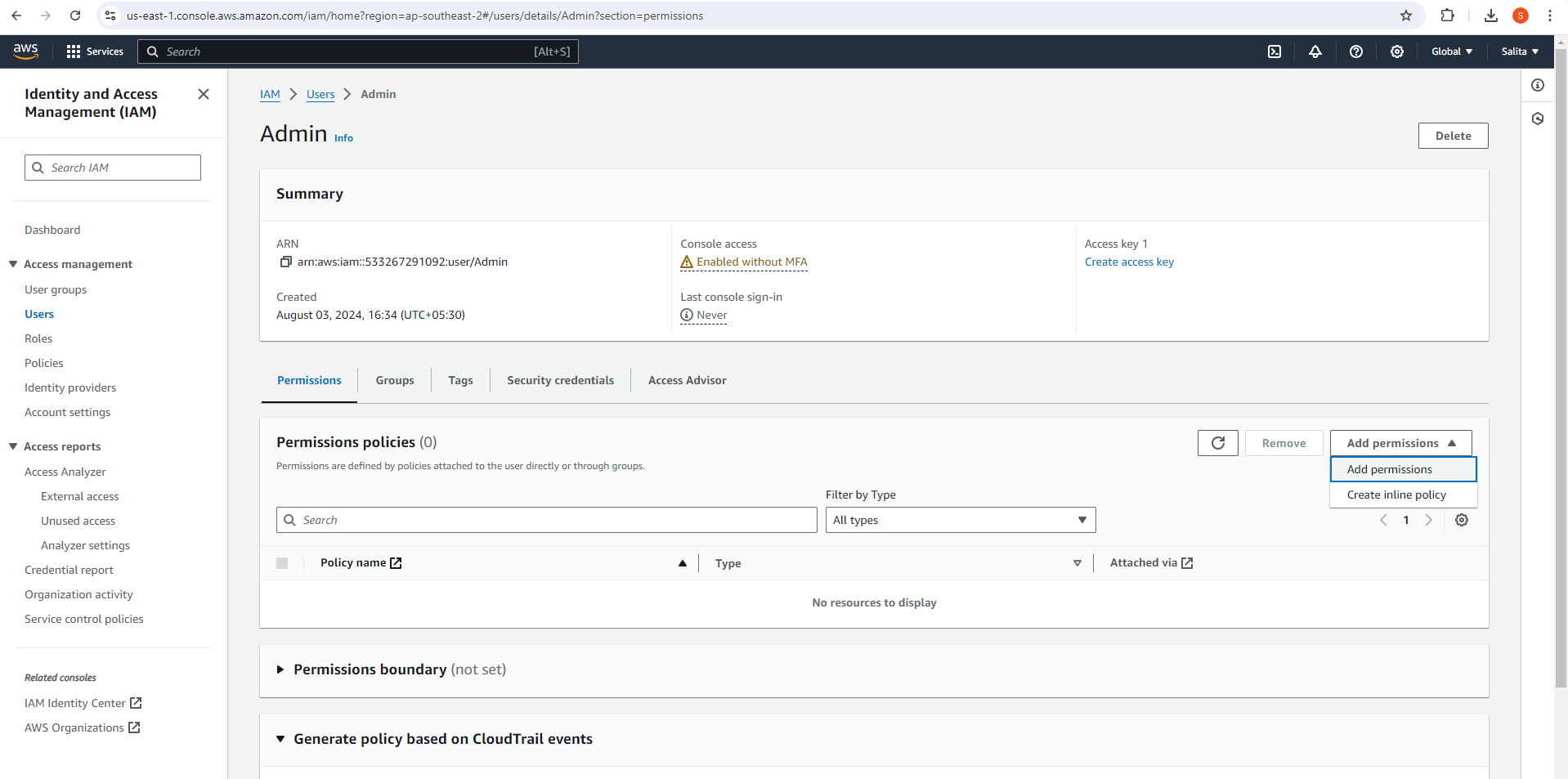
Review and Create Policy.



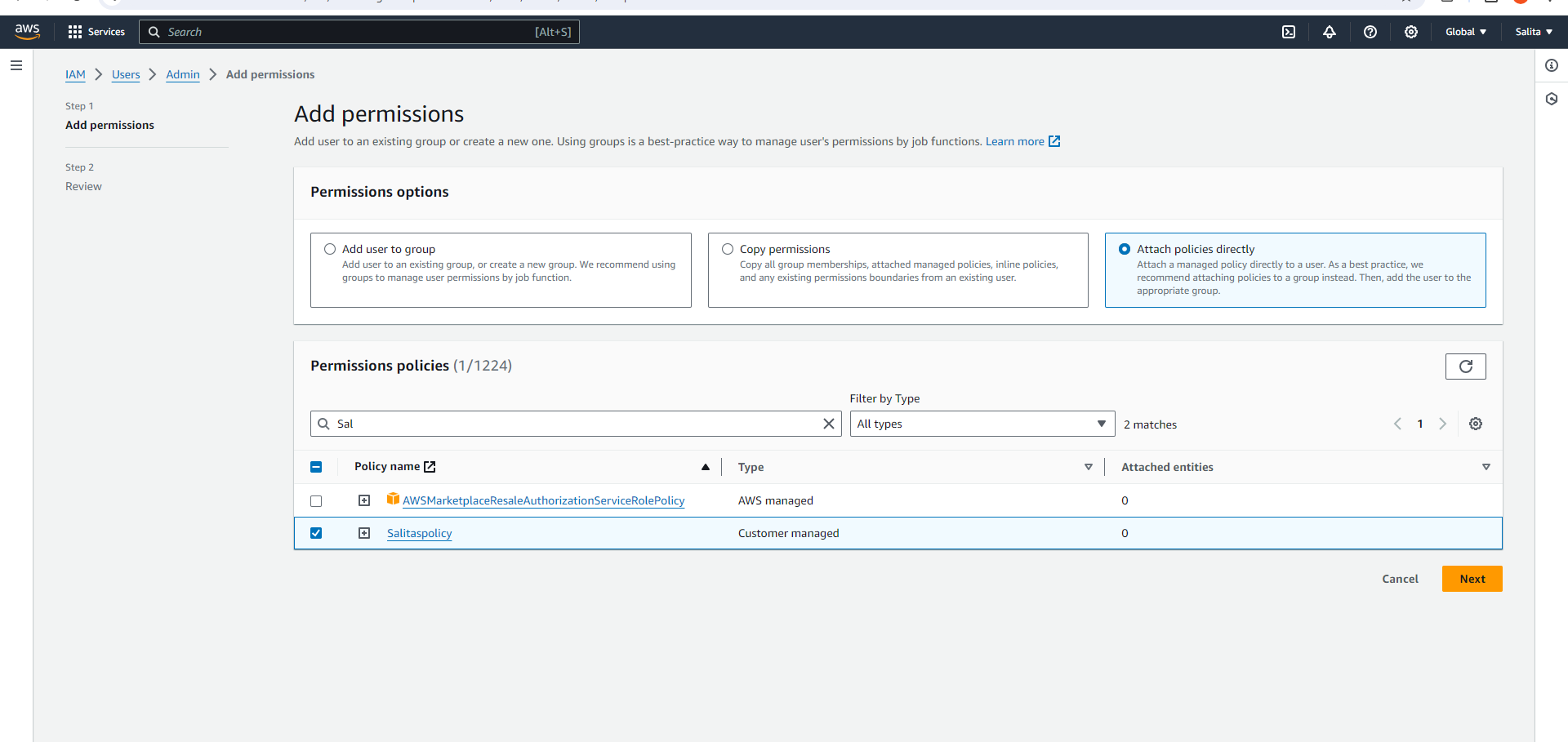
Select ‘Users’ in left hand menu and Select the username.

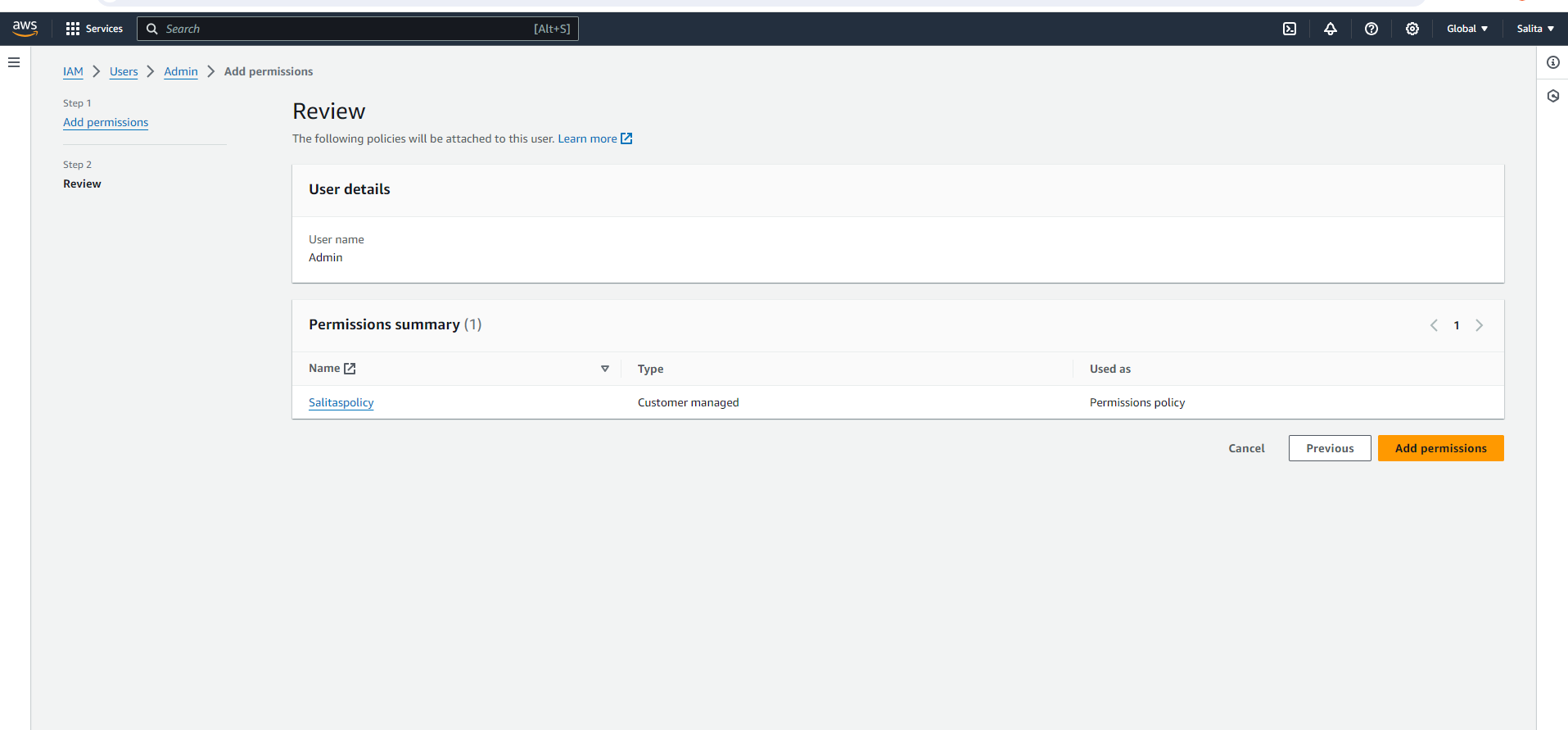


In Permissions, “Add Permissions”.

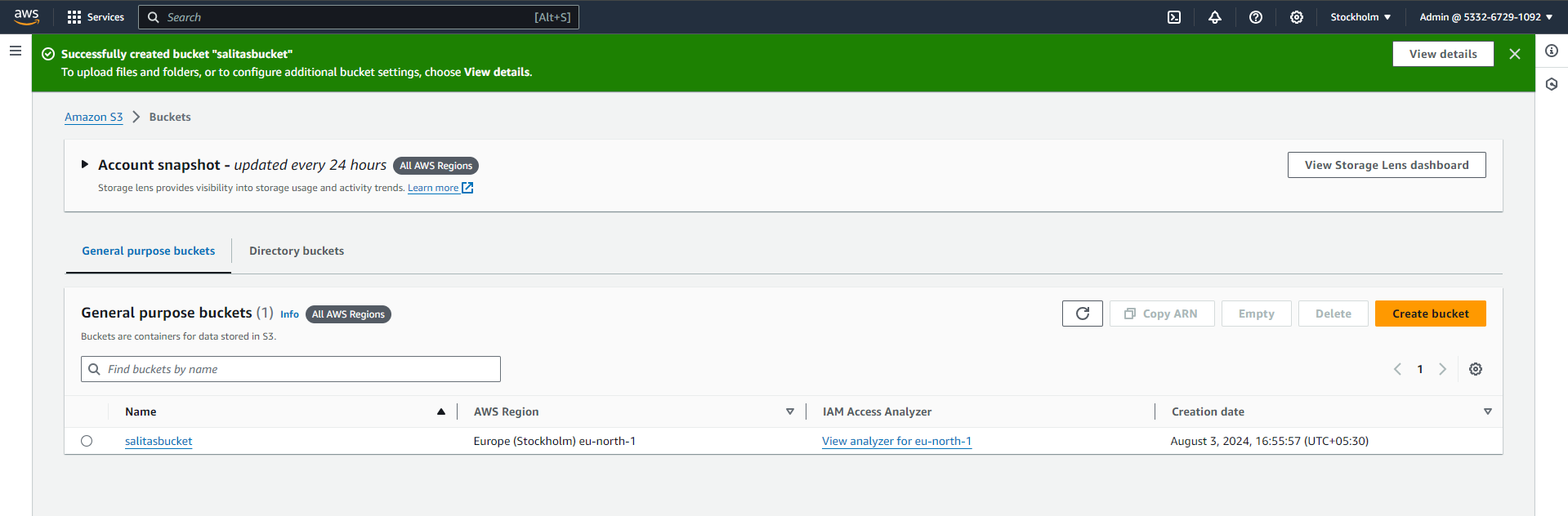


Click on “Attach policies directly” and search for your policy name. Click Next.

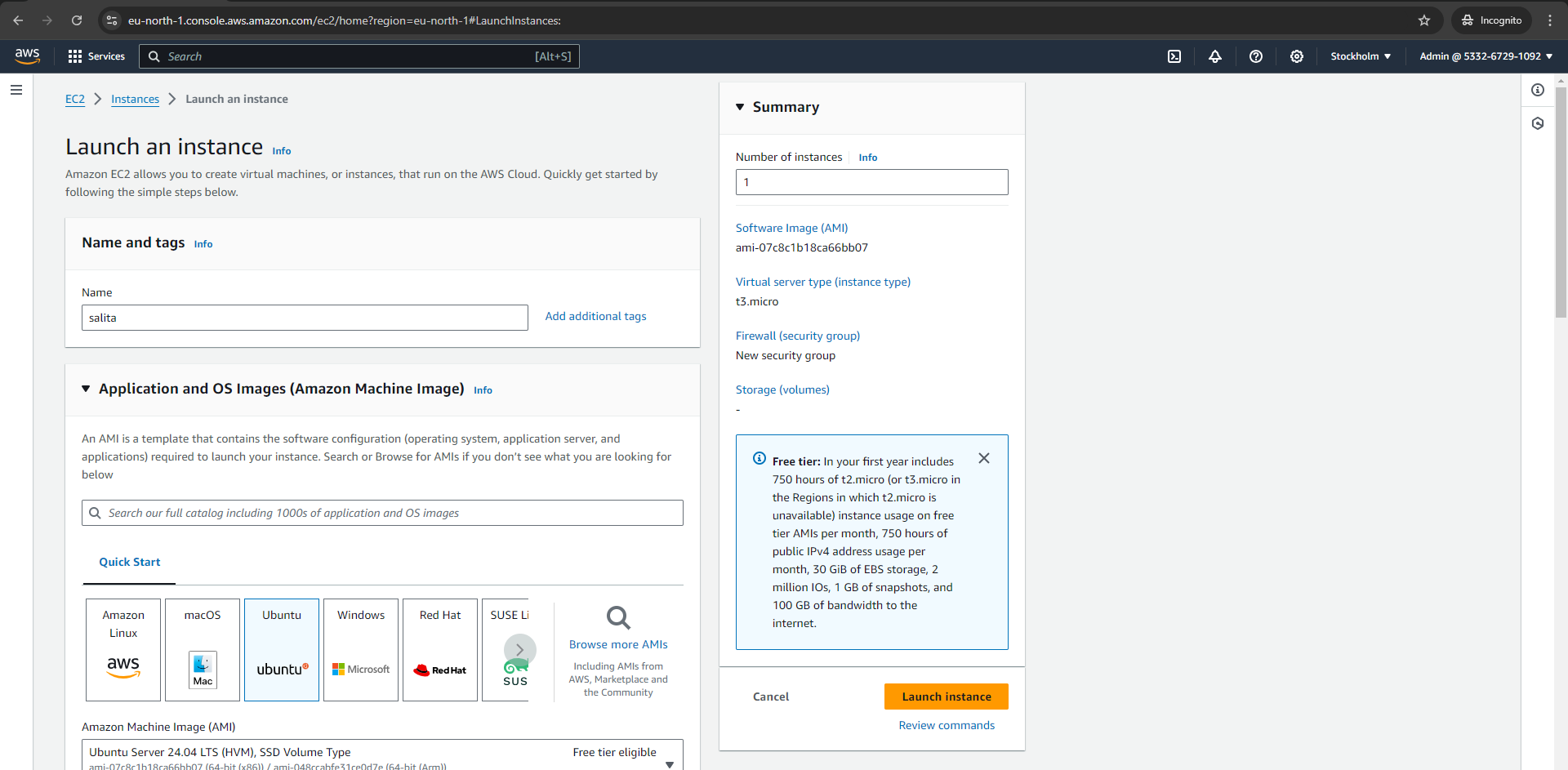


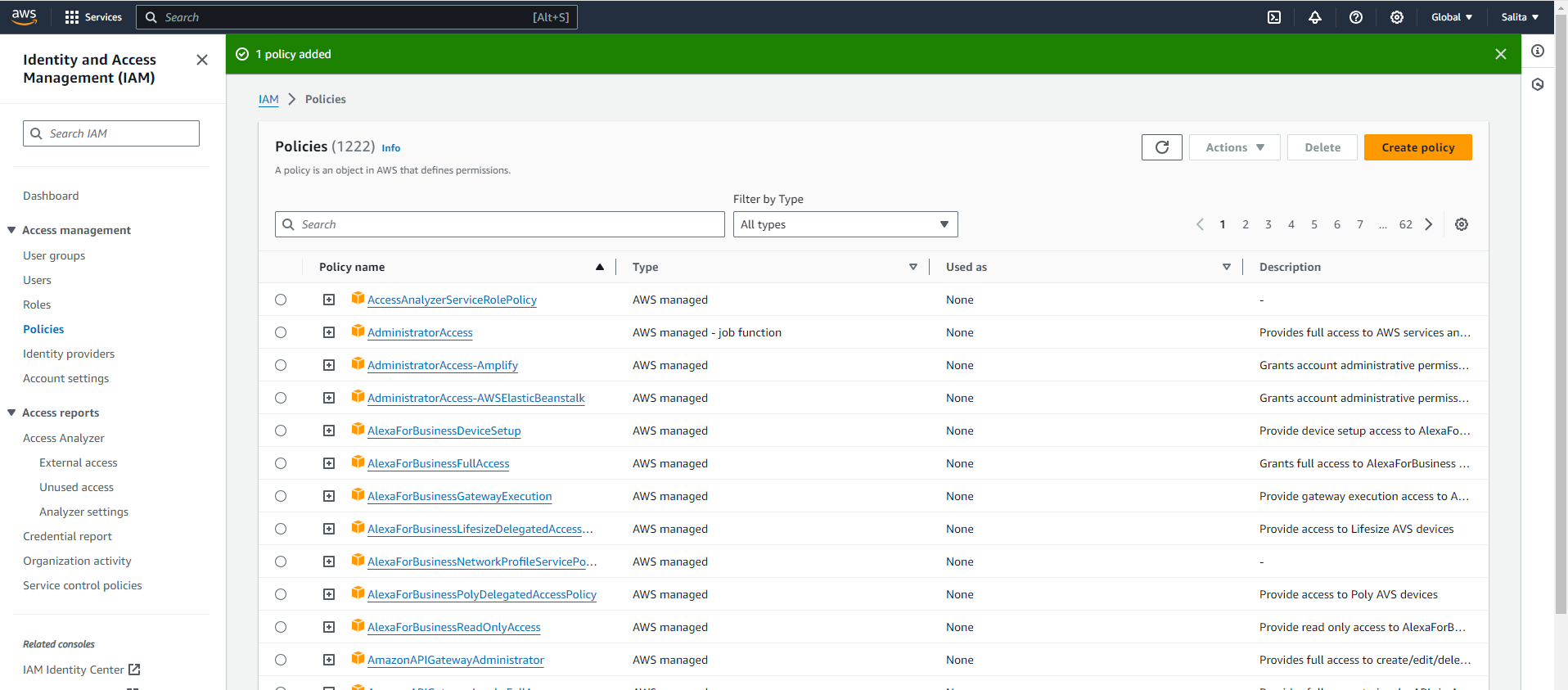


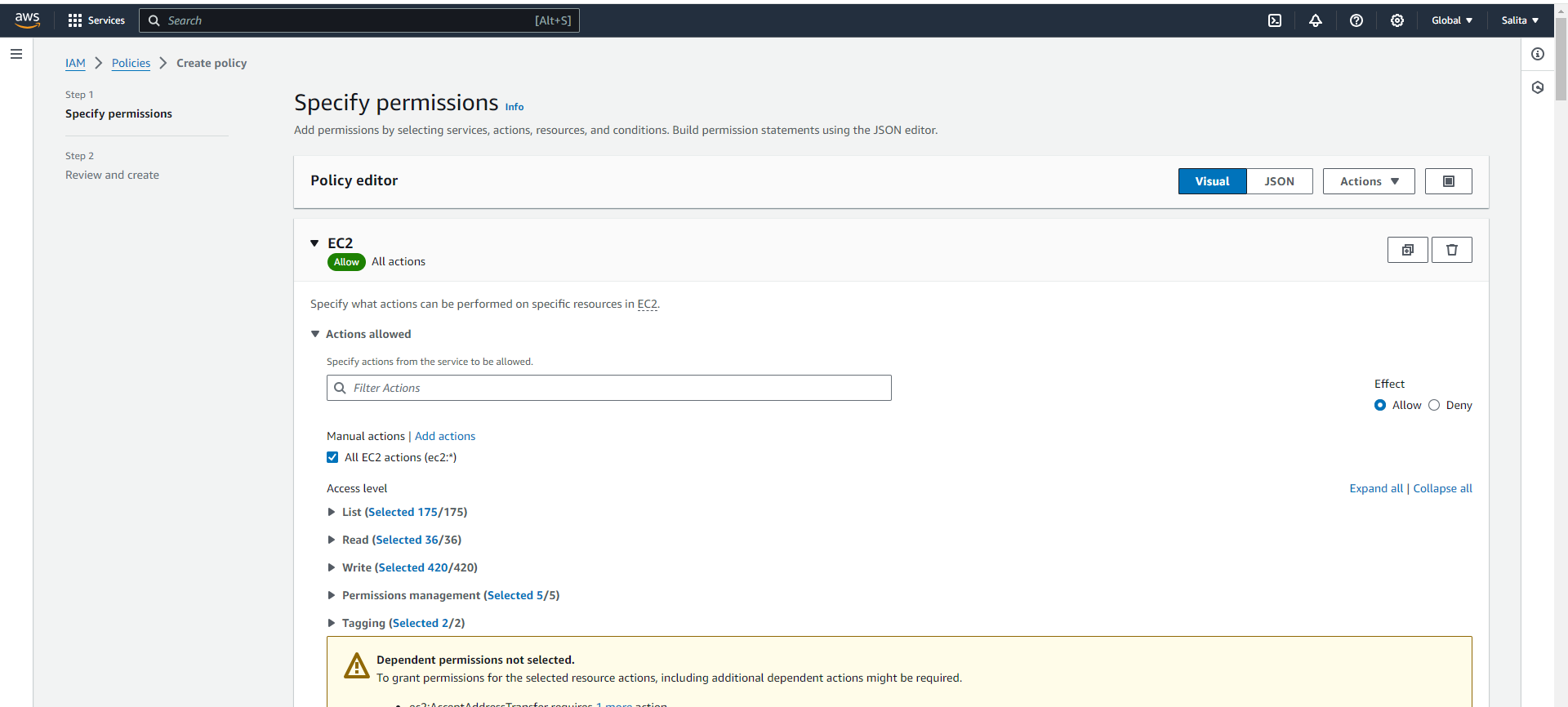
Coming to the Incognito mode click on create bucket and this time the bucket gets created successfully.

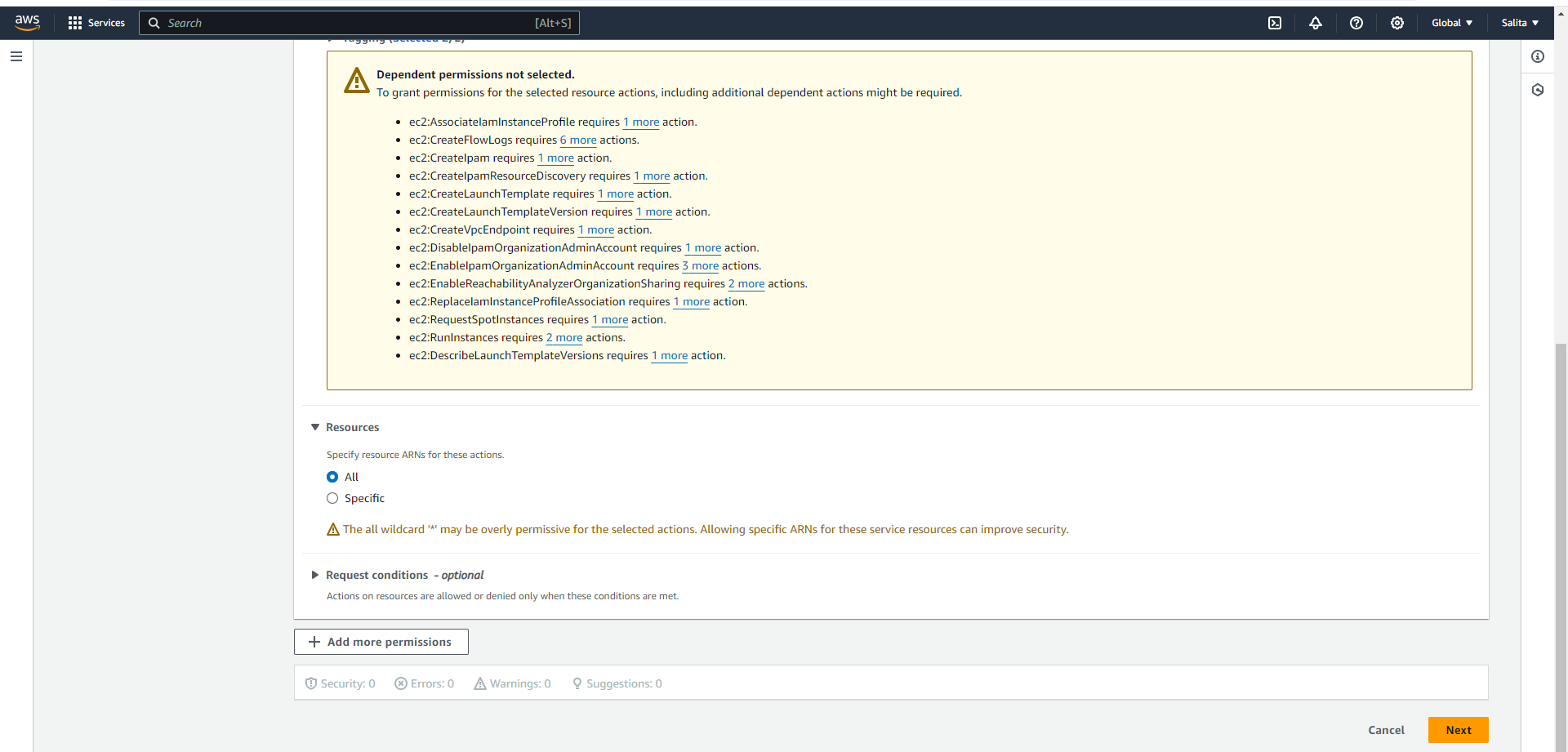


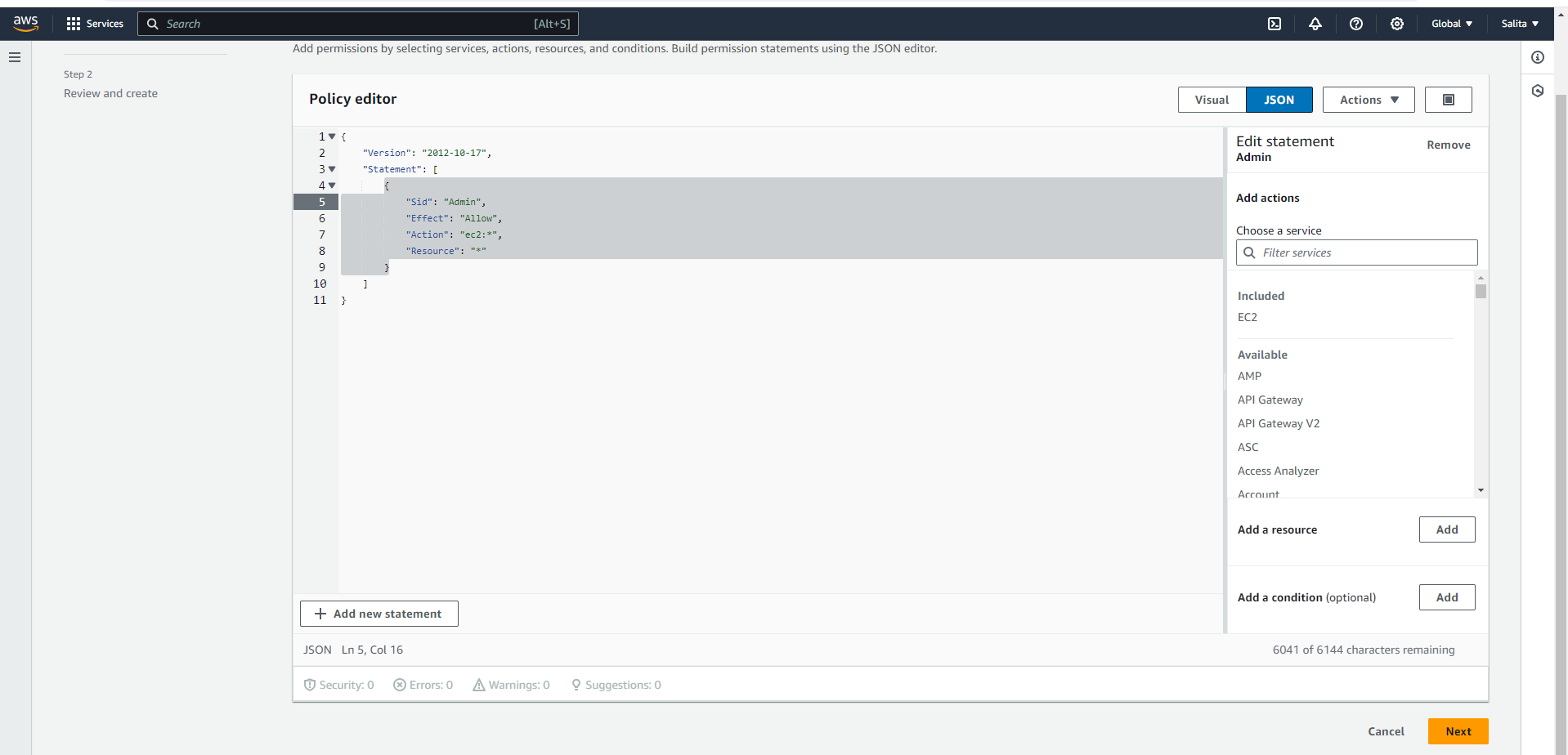
Now for “EC2” service, Repeat the same steps as above by first launching an instance.

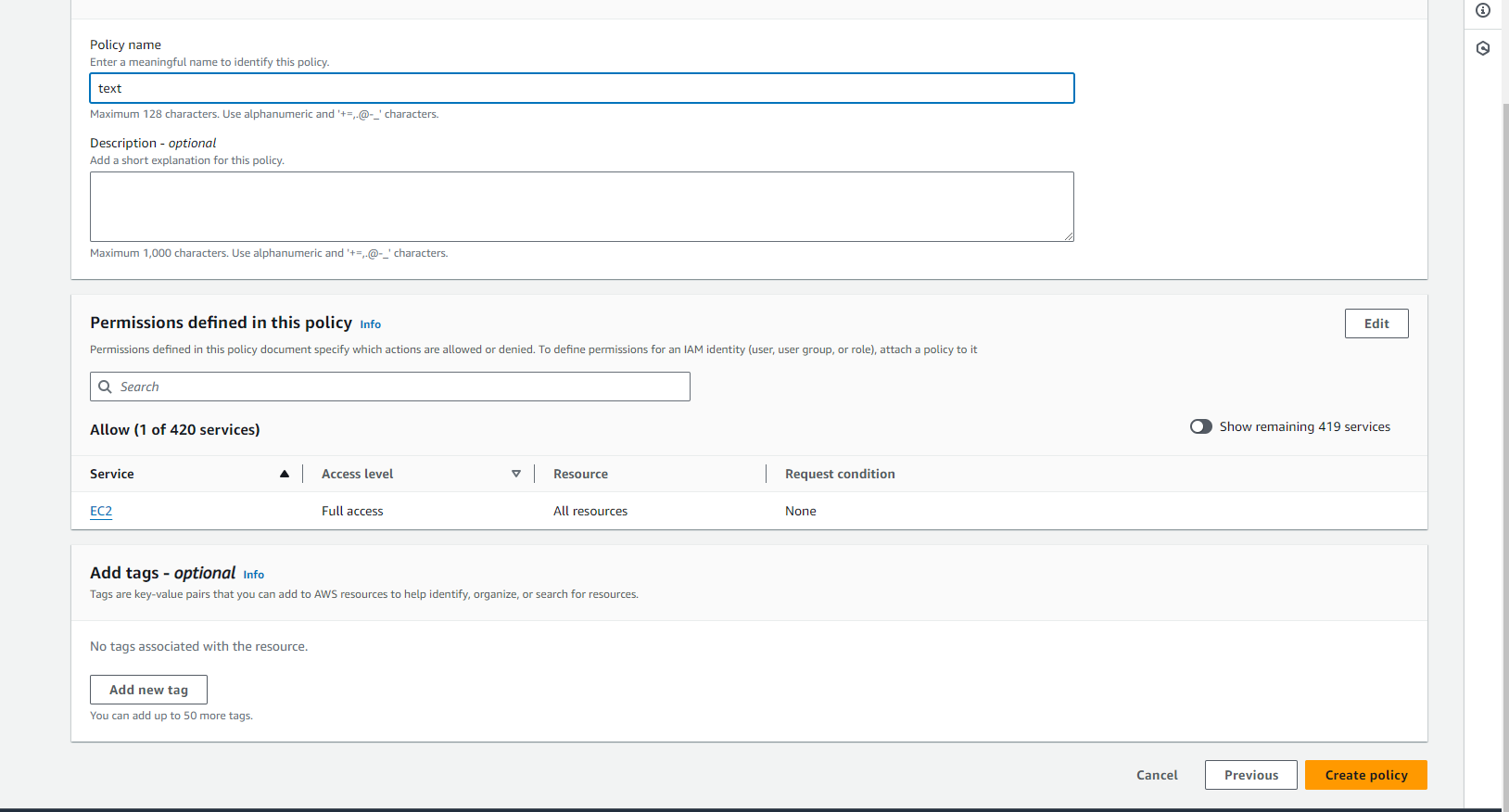


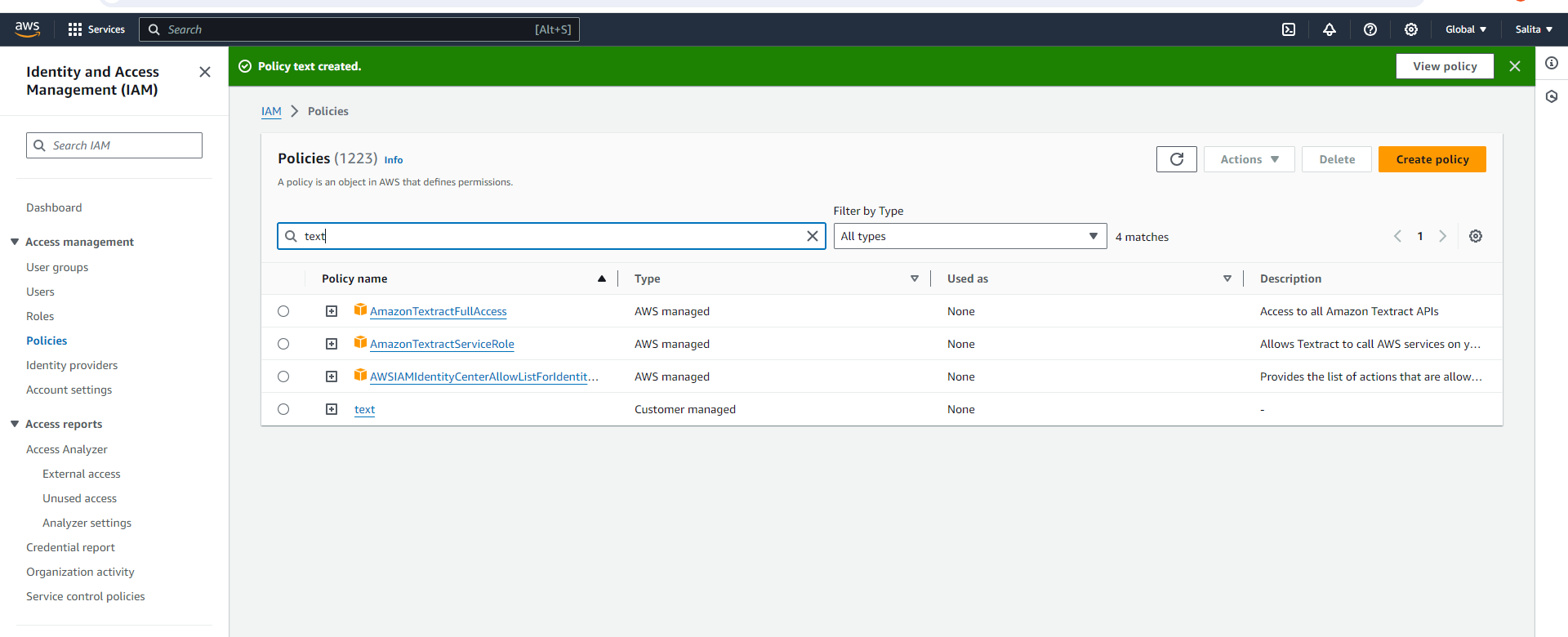


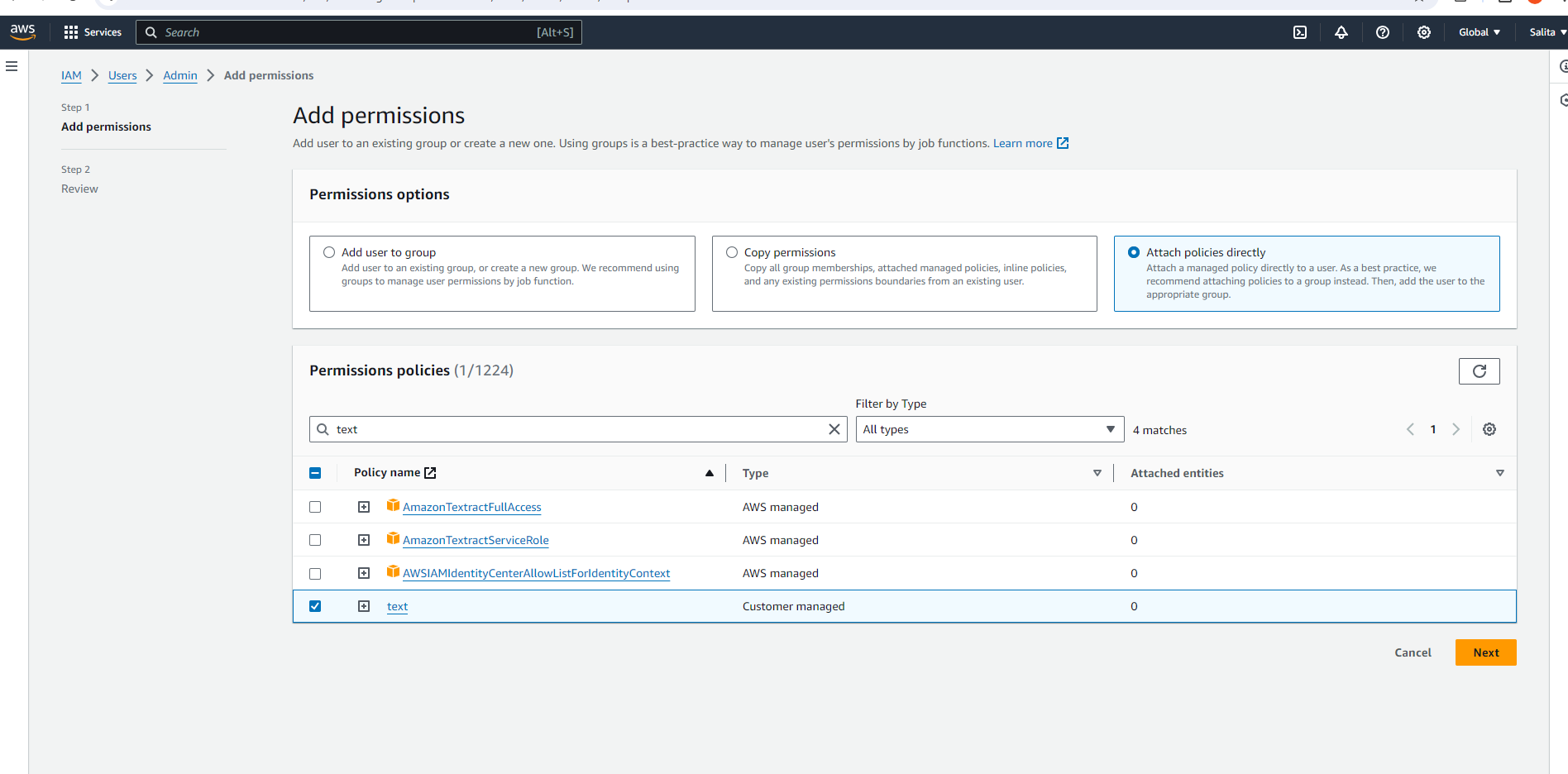


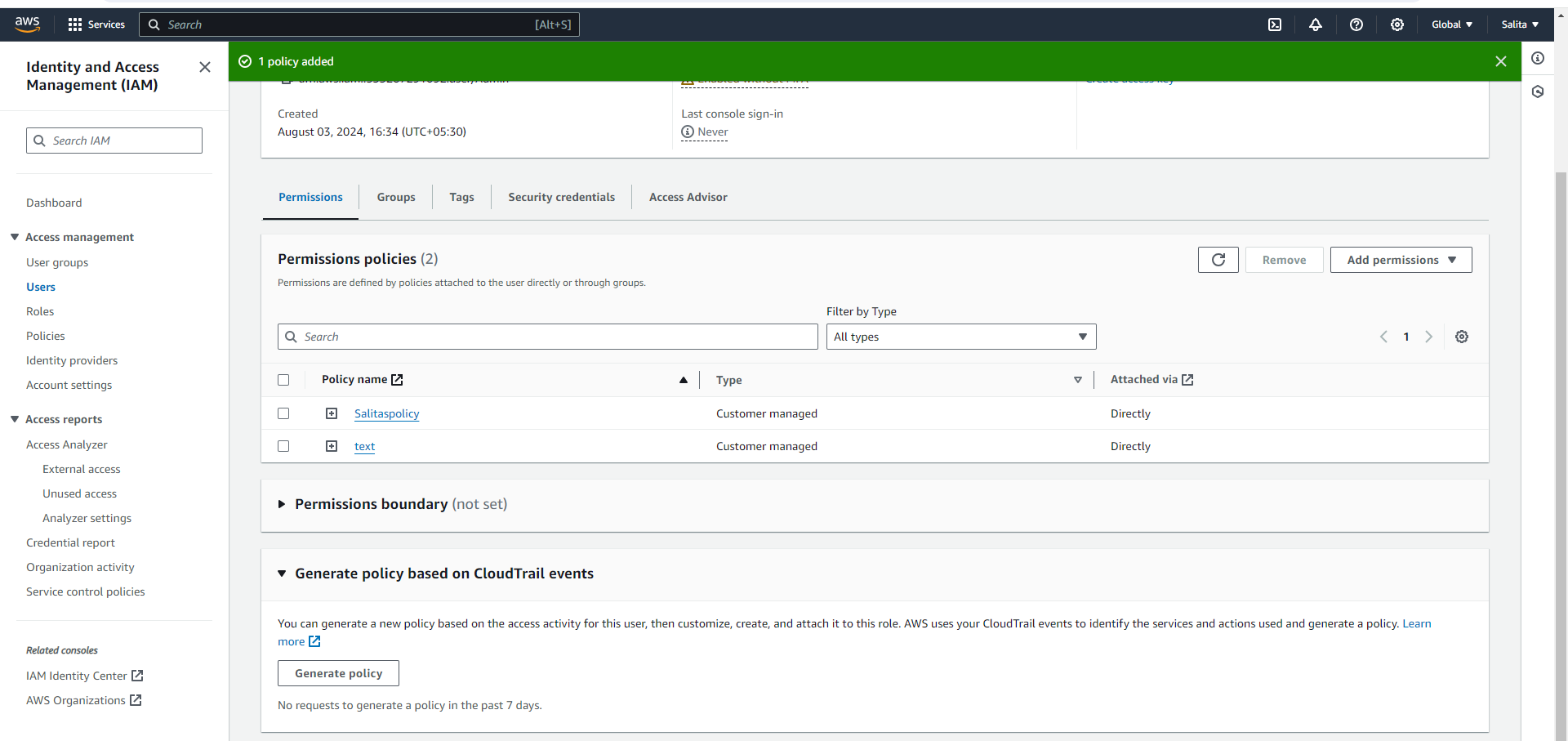












Lastly select the created policies and click Remove to delete them permanently.