NAME- ROSHANI GUPTA

SAP ID-86062300035

ROLL NO –A018

1)

**IAM (Identity and Access Management)** in AWS is a service that helps you securely control access to AWS resources. It allows you to manage who can access your resources (authentication) and what actions they can perform (authorization).

**IAM Users**

* **Definition**: An IAM User represents an individual or a service that needs access to AWS resources. Each user has a unique name and can have their own credentials, like a password or access keys.
* **Purpose**: Users are ideal for assigning permissions to specific individuals or services. For example, you might create separate users for different employees, each with permissions tailored to their job responsibilities.
* **Permissions**: Permissions are assigned directly to users via policies, which define what actions a user can take and on which resources.

**IAM Groups**

* **Definition**: An IAM Group is a collection of IAM Users. Instead of assigning permissions to each user individually, you can assign them to a group, and all users within that group inherit those permissions.
* **Purpose**: Groups simplify management by allowing you to control access for multiple users at once. For example, you can create a "Developers" group with permissions to access certain AWS services and add all developer users to this group.
* **Permissions**: Like users, groups are assigned permissions through policies. Any user added to the group automatically gains the group's permissions.

**IAM Roles**

* **Definition**: An IAM Role is similar to a user, but instead of being tied to a single individual or service, a role defines a set of permissions that can be temporarily assumed by any user, service, or application.
* **Purpose**: Roles are useful for granting temporary access to resources. For example, you might create a role that an EC2 instance can assume to access an S3 bucket, or a role that allows cross-account access between AWS accounts.
* **Permissions**: Roles are governed by two types of policies:
  + **Trust Policy**: Defines who or what can assume the role.
  + **Permission Policy**: Defines what actions the role can perform once assumed.

2)

**IAM (Identity and Access Management)** in AWS is a security service that helps you control access to AWS resources. It allows you to manage users, groups, and roles to define who can do what in your AWS environment.

**Key Components:**

1. **IAM Users**:
   * Individual accounts representing people or services that need access to AWS.
   * Users can have login credentials (password, access keys) and permissions assigned through policies.
2. **IAM Groups**:
   * Collections of users with shared permissions.
   * Simplify management by assigning policies to groups, so all users in the group inherit the same permissions.
3. **IAM Roles**:
   * Temporary credentials that define a set of permissions which can be assumed by any entity (user, service, or application).
   * Commonly used for service-to-service access, cross-account access, and federated identities.
4. **IAM Policies**:
   * JSON documents that define permissions, specifying allowed or denied actions on AWS resources.
   * Can be attached to users, groups, or roles.
5. **Identity Providers and Federation**:
   * Allows integration with external identity providers (SAML, OIDC) for Single Sign-On (SSO) and federated access.
6. **Multi-Factor Authentication (MFA)**:
   * Adds an extra layer of security by requiring a second form of authentication, such as a one-time passcode.

**Best Practices:**

* **Principle of Least Privilege**: Grant only the permissions necessary.
* **Enable MFA**: Especially for privileged accounts.
* **Use Roles Over Long-Term Access Keys**: For temporary and secure access.
* **Monitor and Rotate Credentials**: Regularly to maintain security.
* **Organize with Groups**: To manage permissions efficiently.
* **Resource-Level Permissions**: Apply permissions to specific resources for tighter control.

IAM is essential for securing your AWS environment by providing precise control over access and permissions.

3)

IAM (Identity and Access Management) roles are an integral part of AWS (Amazon Web Services) that allow you to define a set of permissions for making AWS service requests. Instead of assigning specific permissions to individual users, IAM roles enable you to create roles that have a specific set of permissions, which can then be assumed by any entity that needs them.

**Key Concepts of IAM Roles:**

1. **Assume Role**:
   * **Users or services can assume a role** to temporarily gain the permissions attached to that role. This is particularly useful for granting cross-account access or allowing services to perform tasks on your behalf.
2. **Temporary Security Credentials**:
   * When a role is assumed, AWS provides temporary security credentials (access key ID, secret access key, and session token) that can be used to make AWS API requests.
3. **Role Types**:
   * **Service Roles**: Used by AWS services to perform actions on your behalf (e.g., a Lambda function assuming a role to read from a DynamoDB table).
   * **Cross-Account Roles**: Allow access between different AWS accounts.
   * **Identity Federation Roles**: Allow external identities (e.g., users from an external identity provider) to assume roles and access AWS resources.
4. **Trust Policy**:
   * Every IAM role has a trust policy that specifies which entities (e.g., users, services, accounts) are allowed to assume the role.
5. **Permission Policies**:
   * Attached to roles to define what actions the role can perform and on which resources.

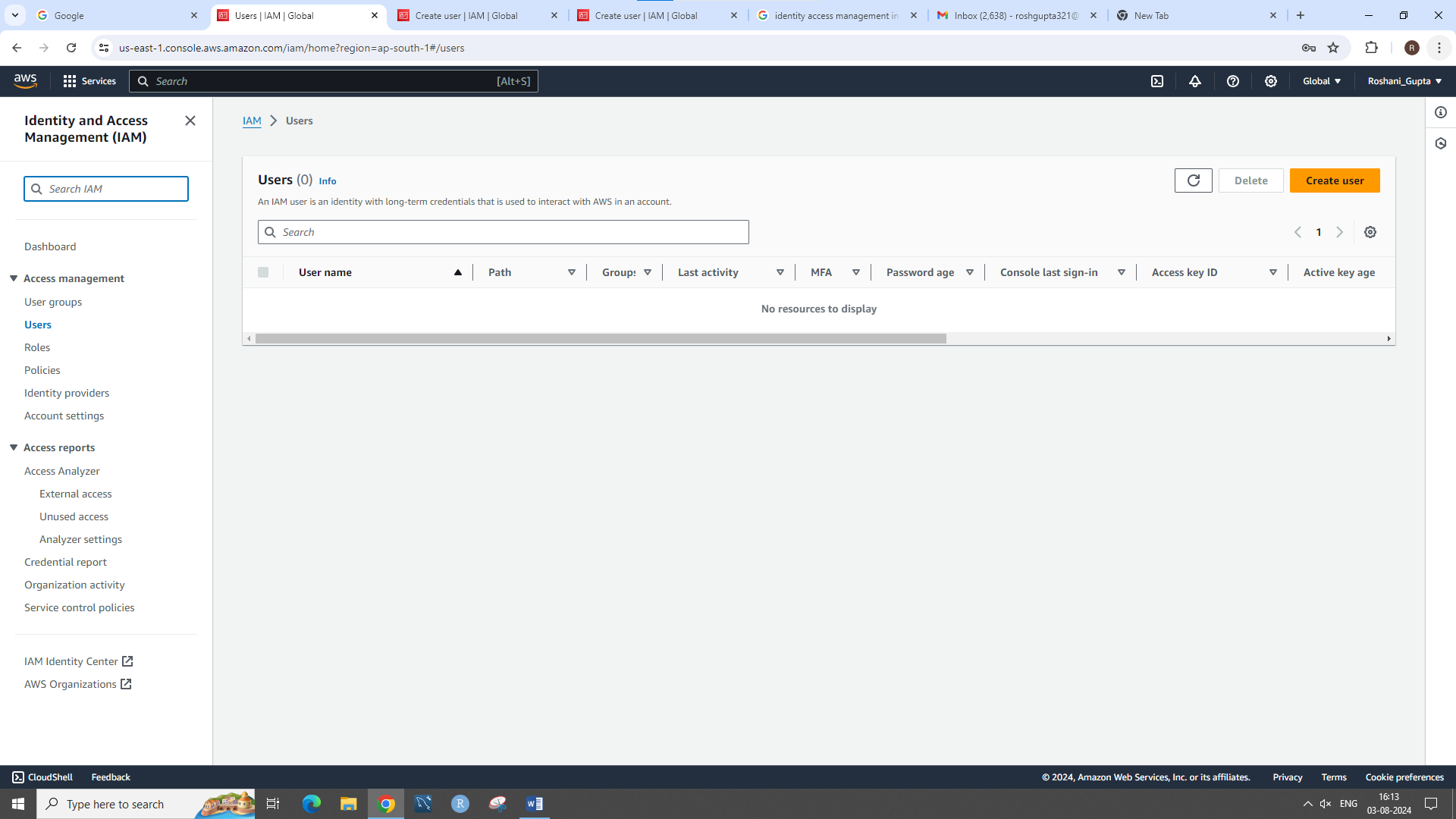
**Use Cases for IAM Roles:**

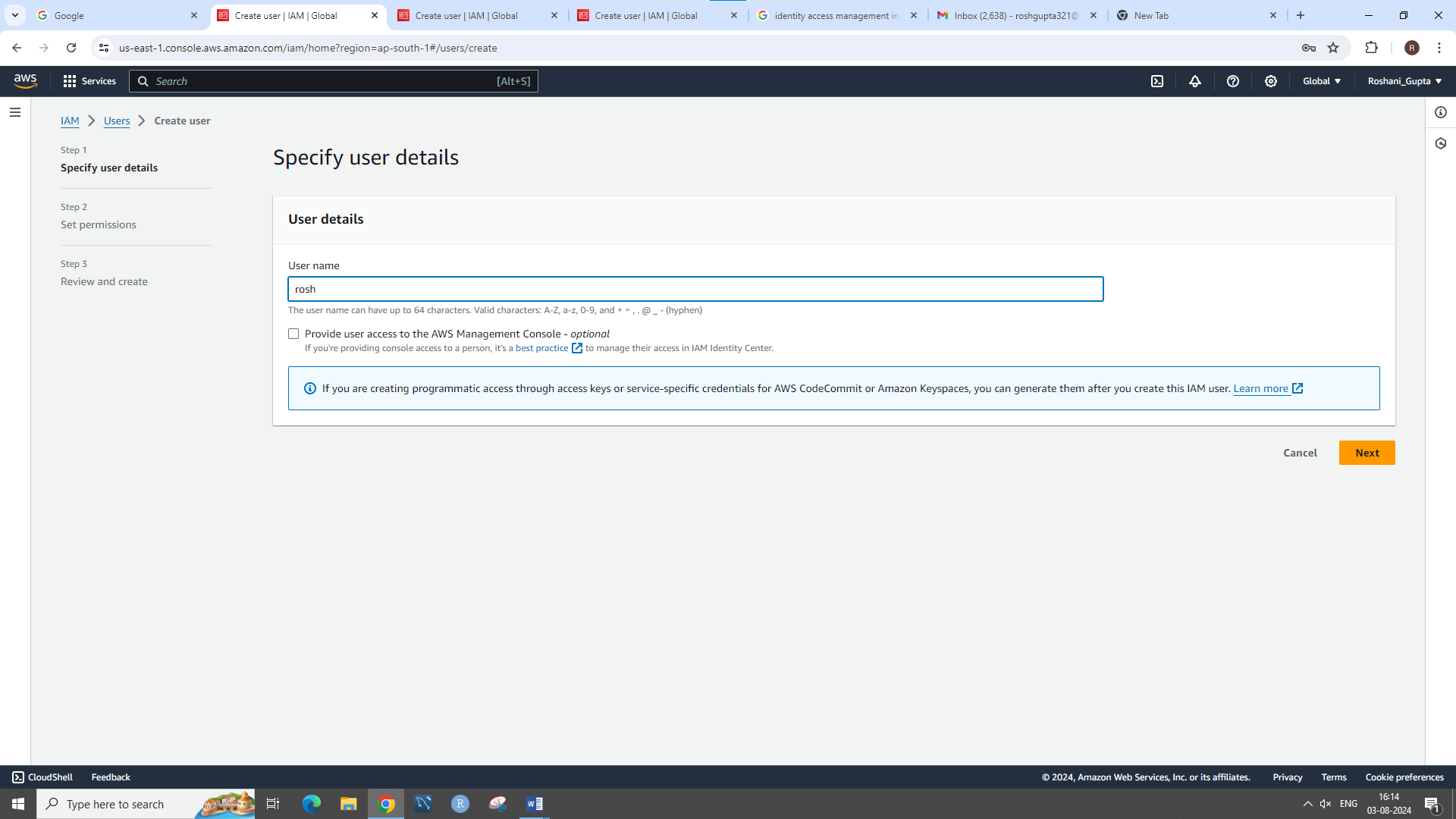
* **Service-to-Service Access**: Allow AWS services to interact with each other (e.g., EC2 accessing S3).
* **Cross-Account Access**: Share resources between different AWS accounts.
* **Temporary User Access**: Provide temporary access to AWS resources without long-term credentials.
* **Federated Access**: Integrate with external identity providers (e.g., SAML, OIDC) to allow access to AWS resources.

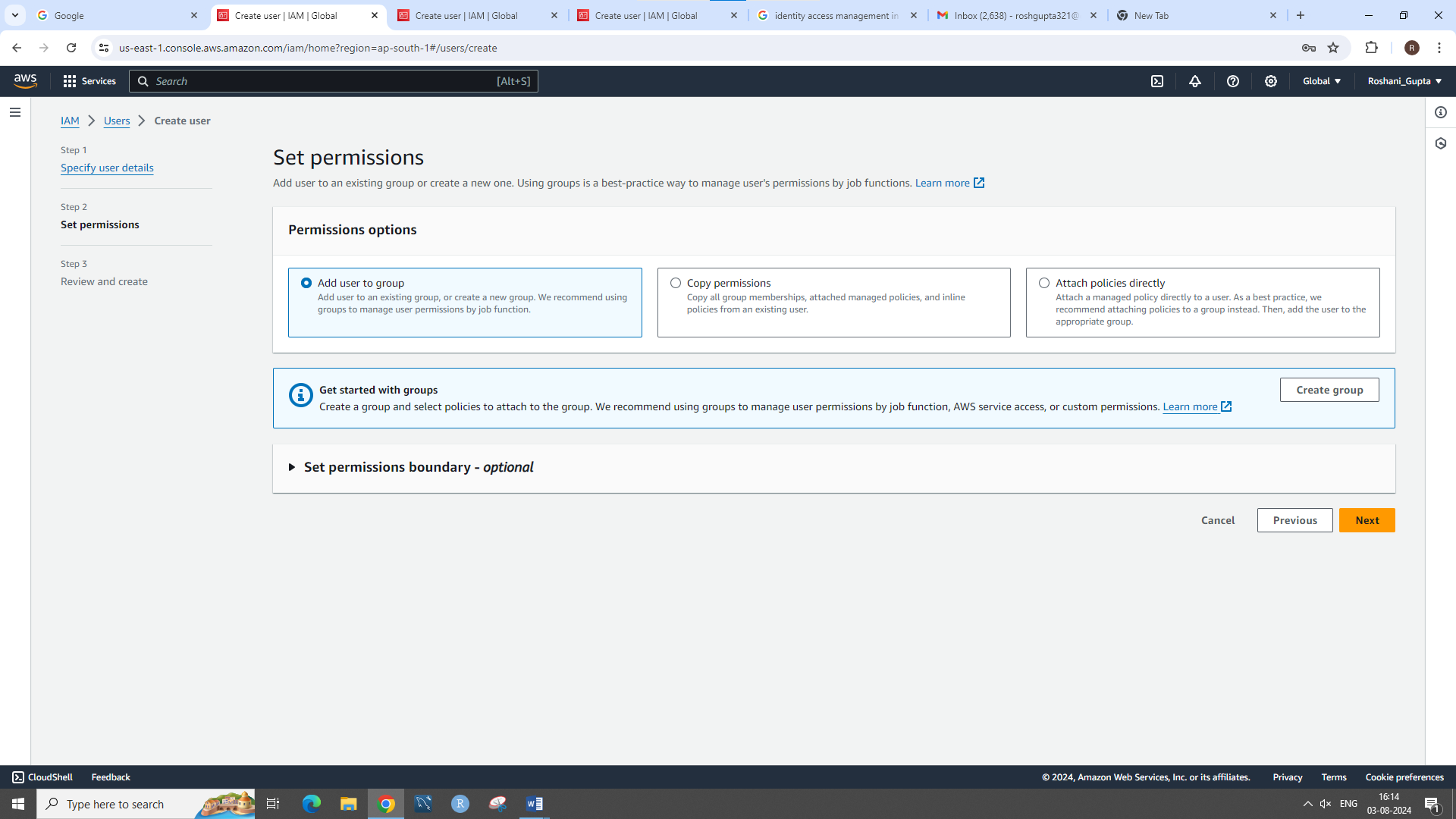
**Example Workflow:**

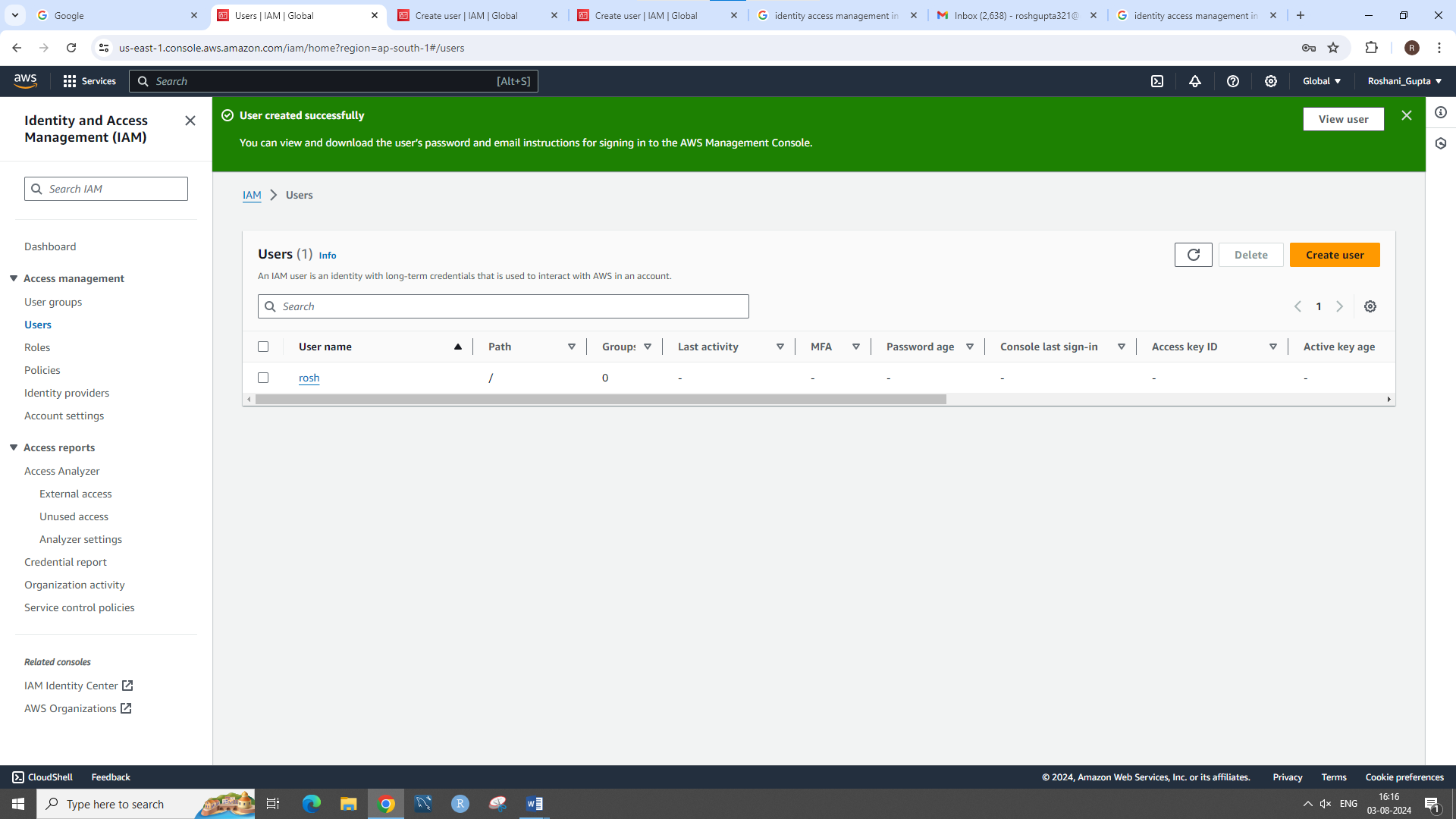
1. A user or service requests to assume a role.
2. If the request is allowed by the trust policy, AWS provides temporary security credentials.
3. The user or service uses these credentials to make requests to AWS services, within the permissions defined by the role's permission policy.

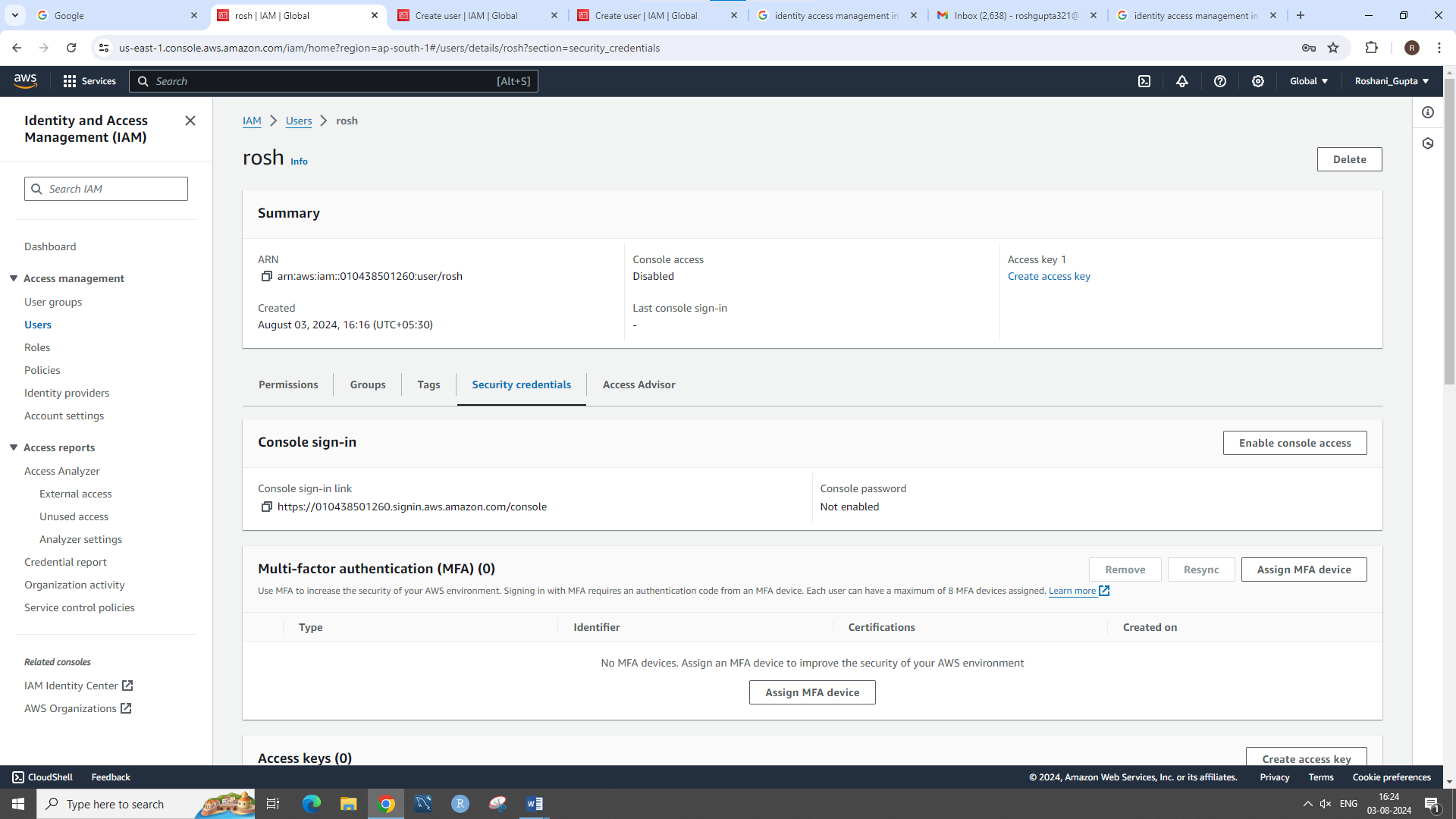
IAM roles are a powerful feature in AWS that can help manage access to resources securely and flexibly.

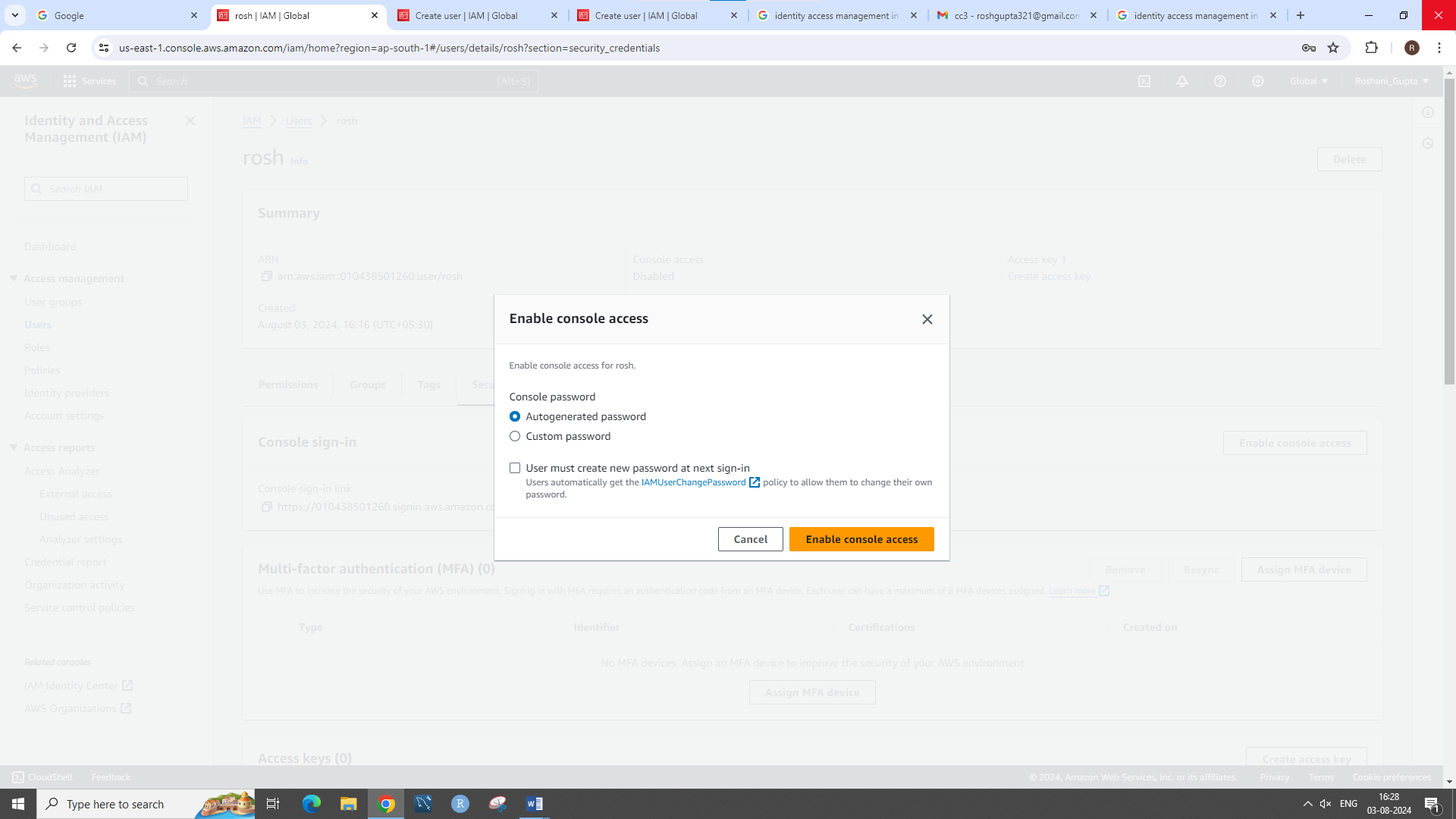


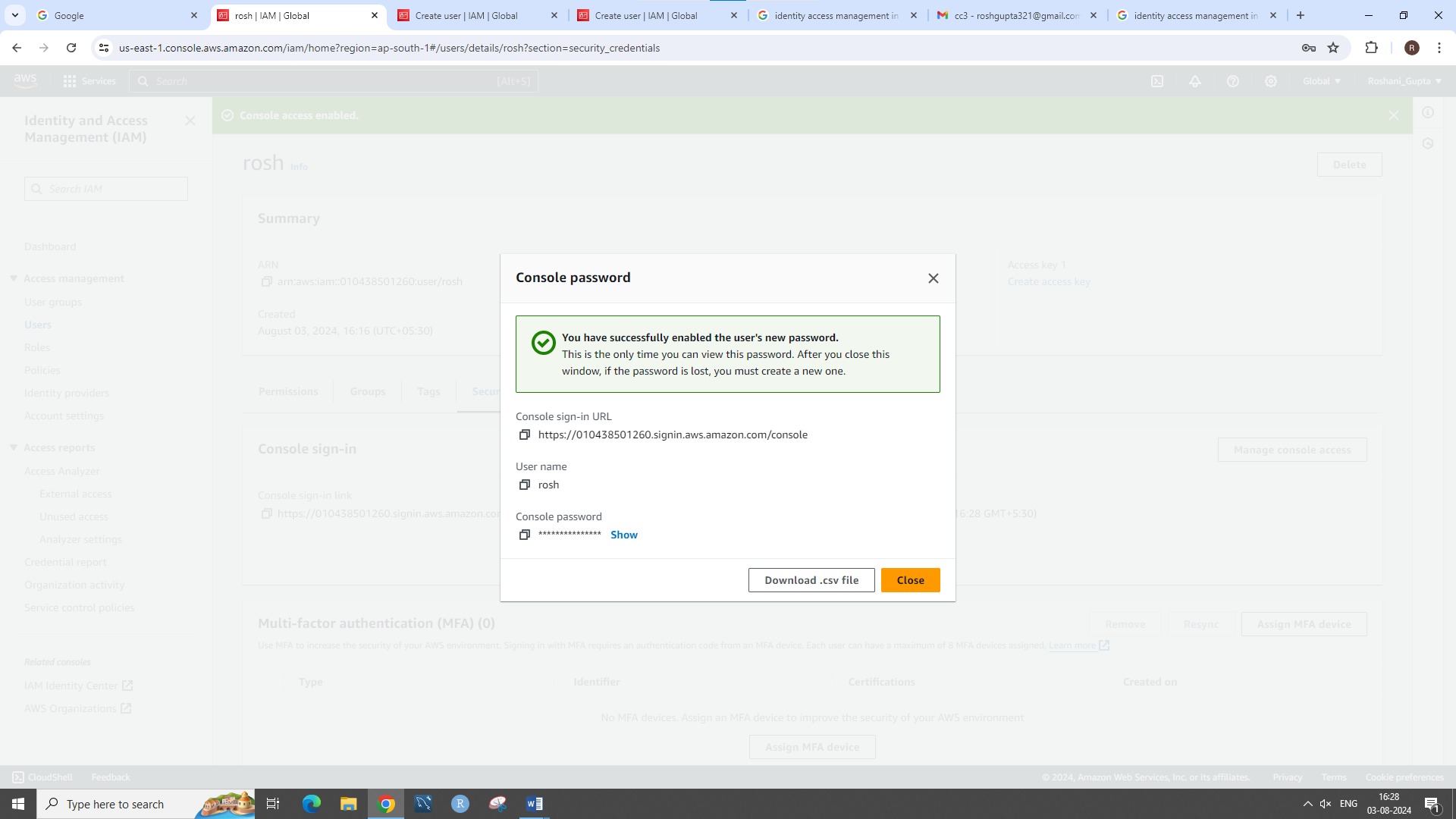




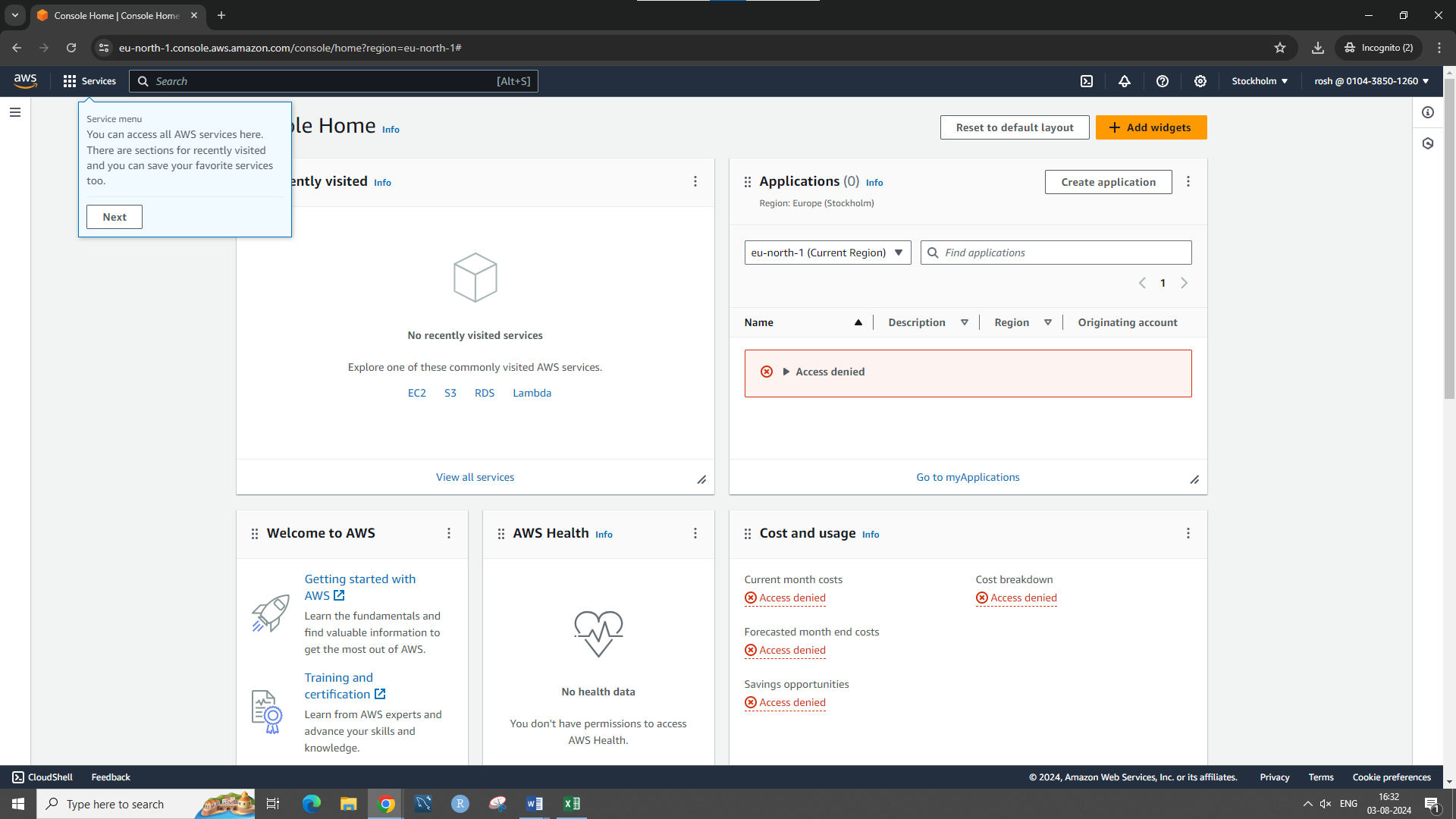


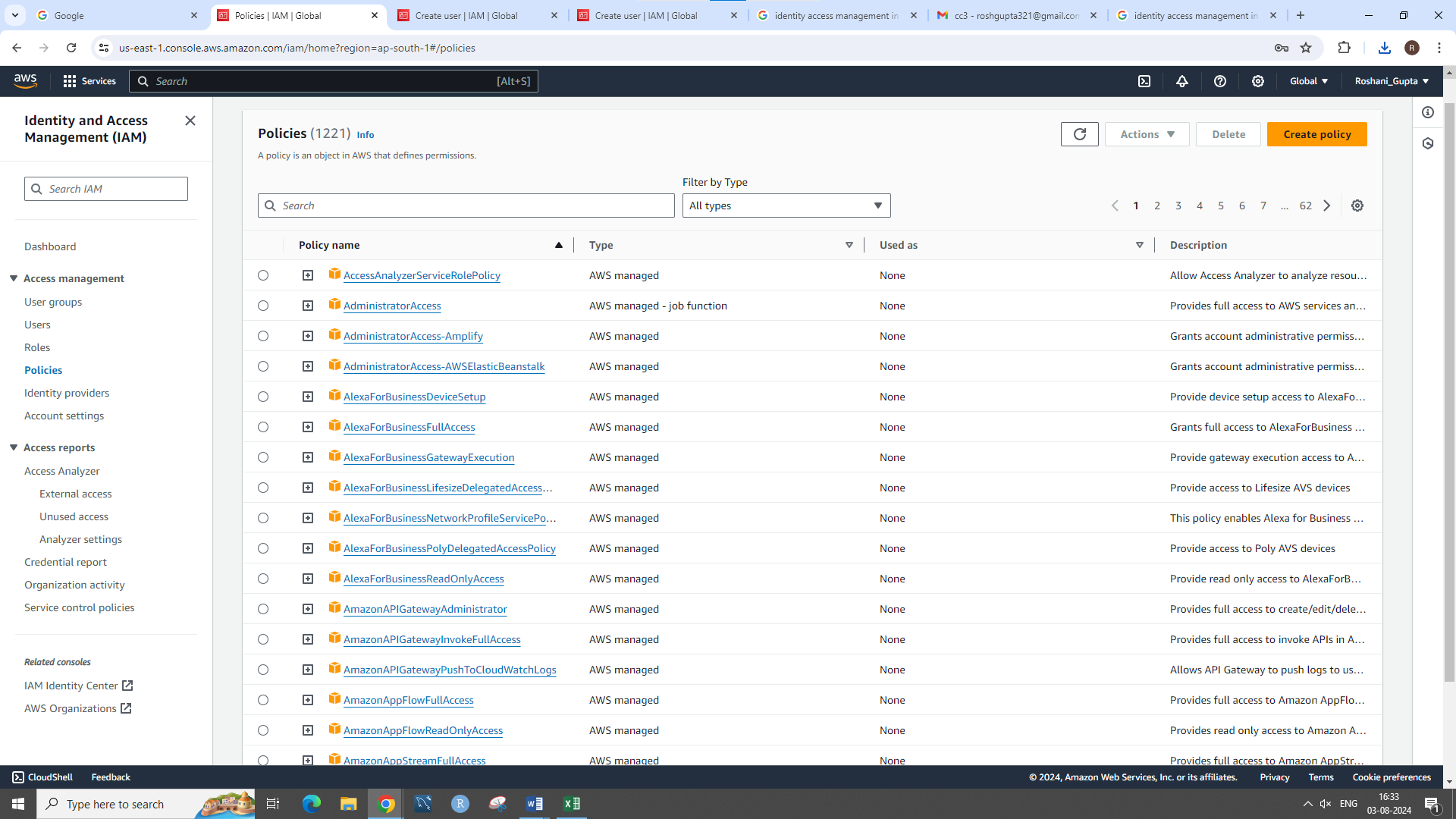


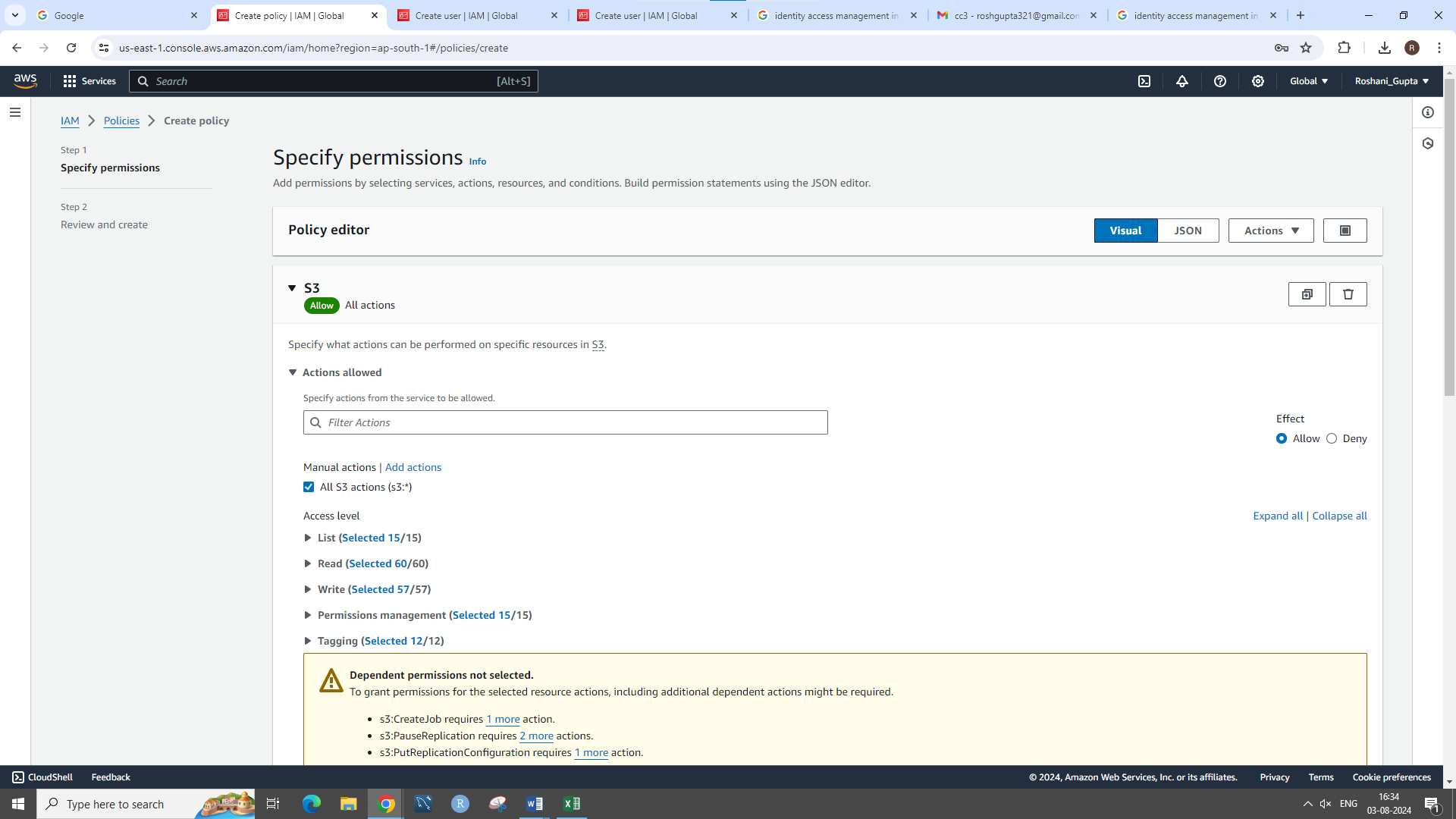


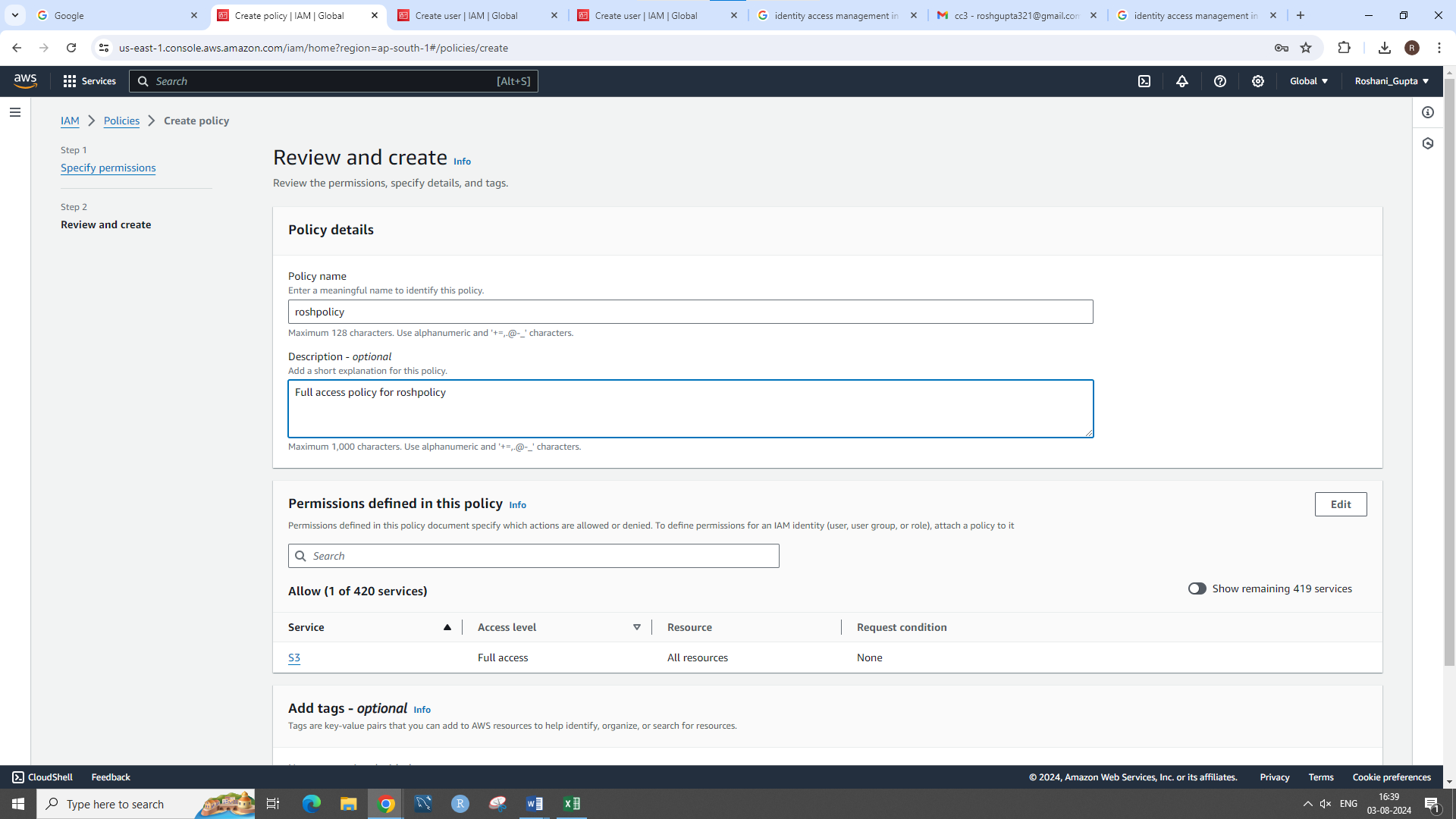


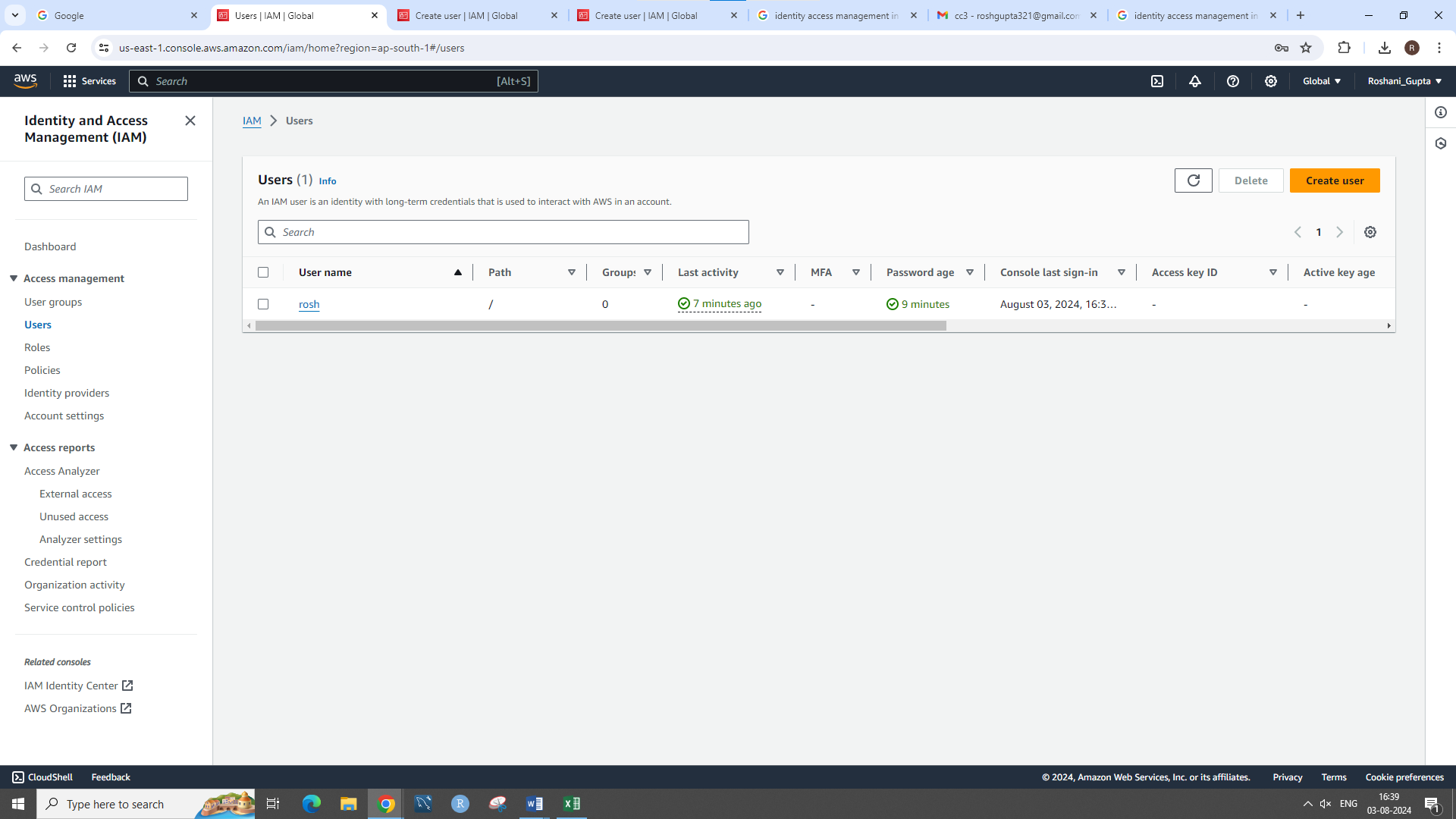


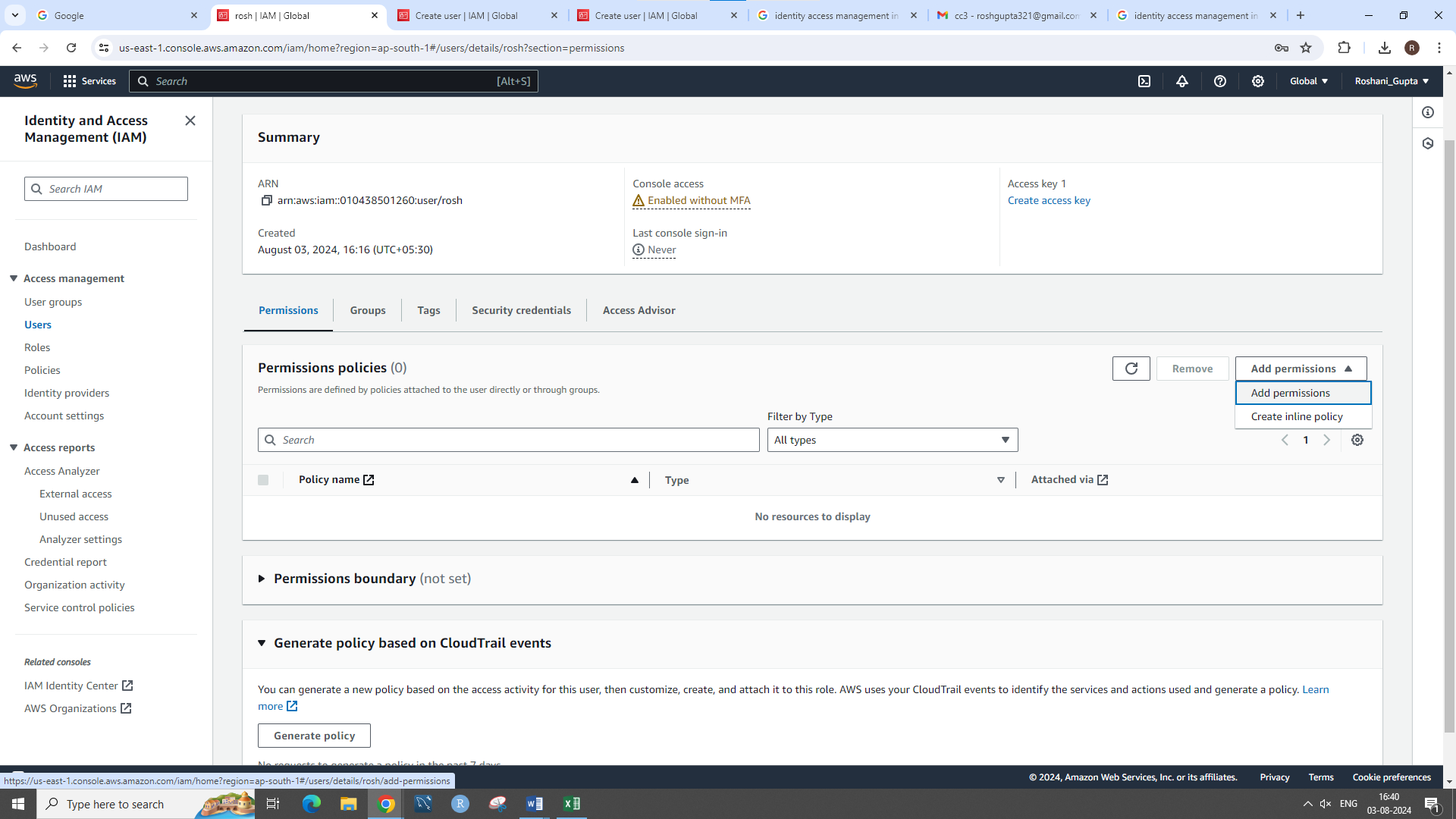


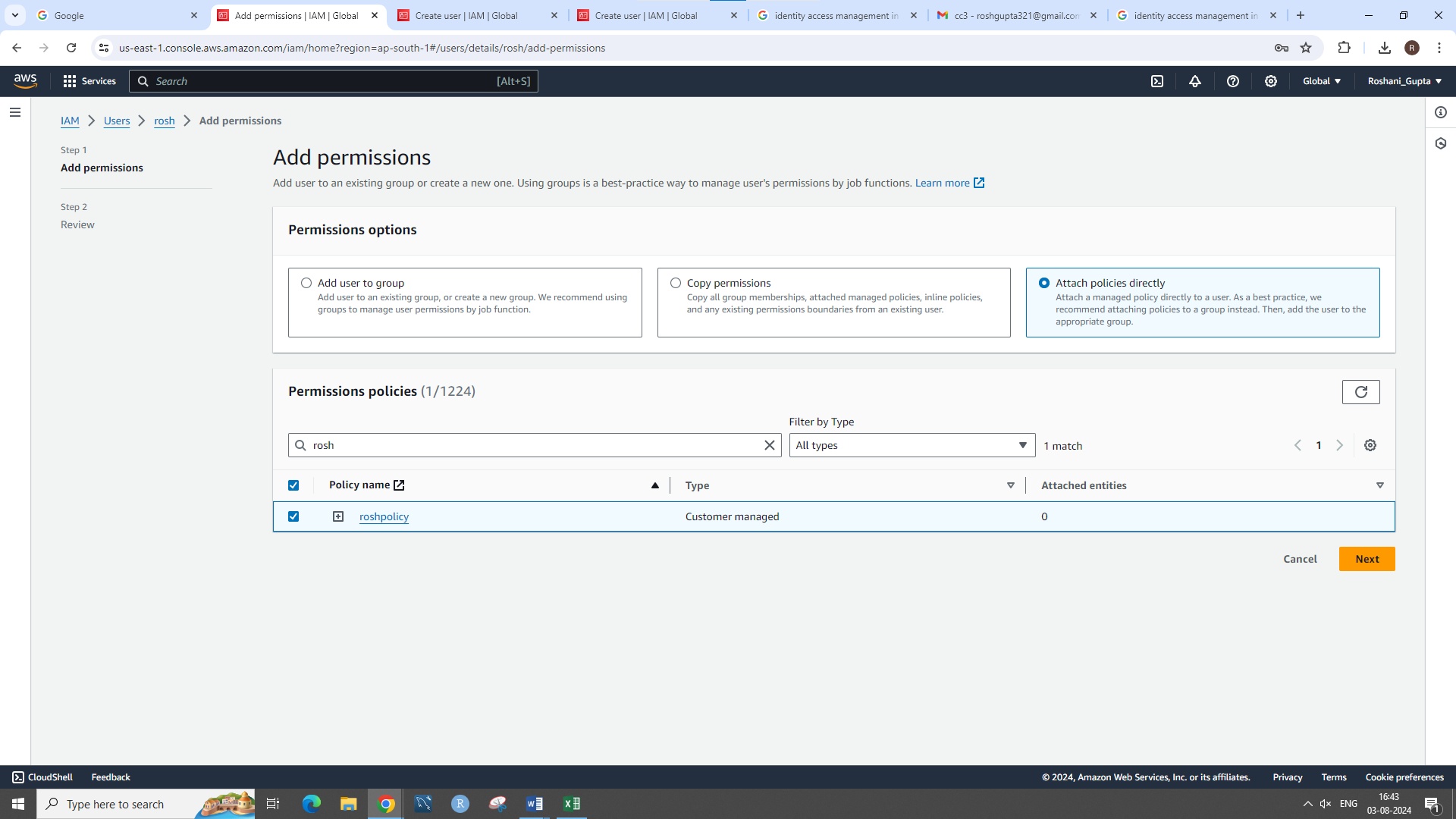




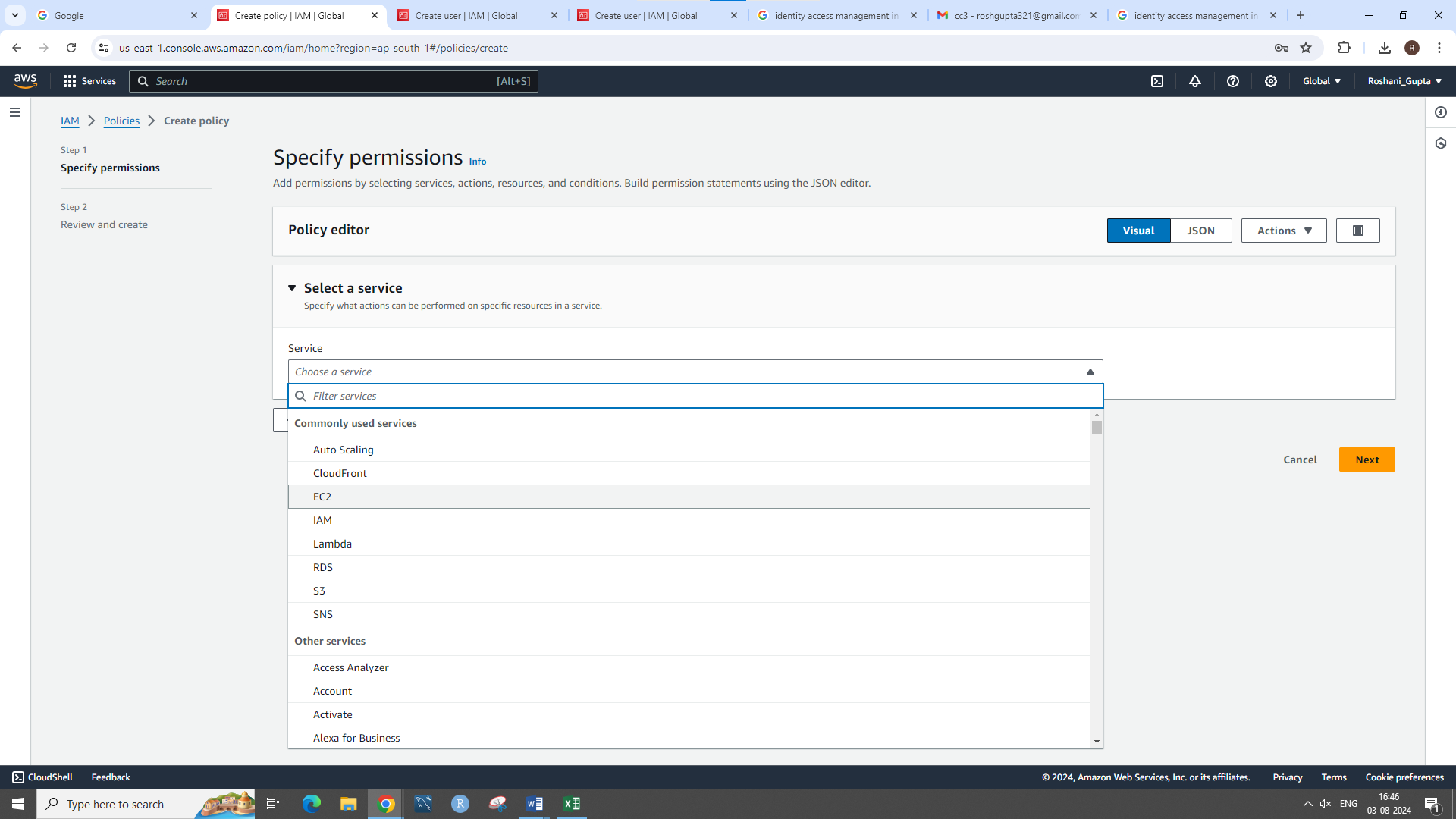


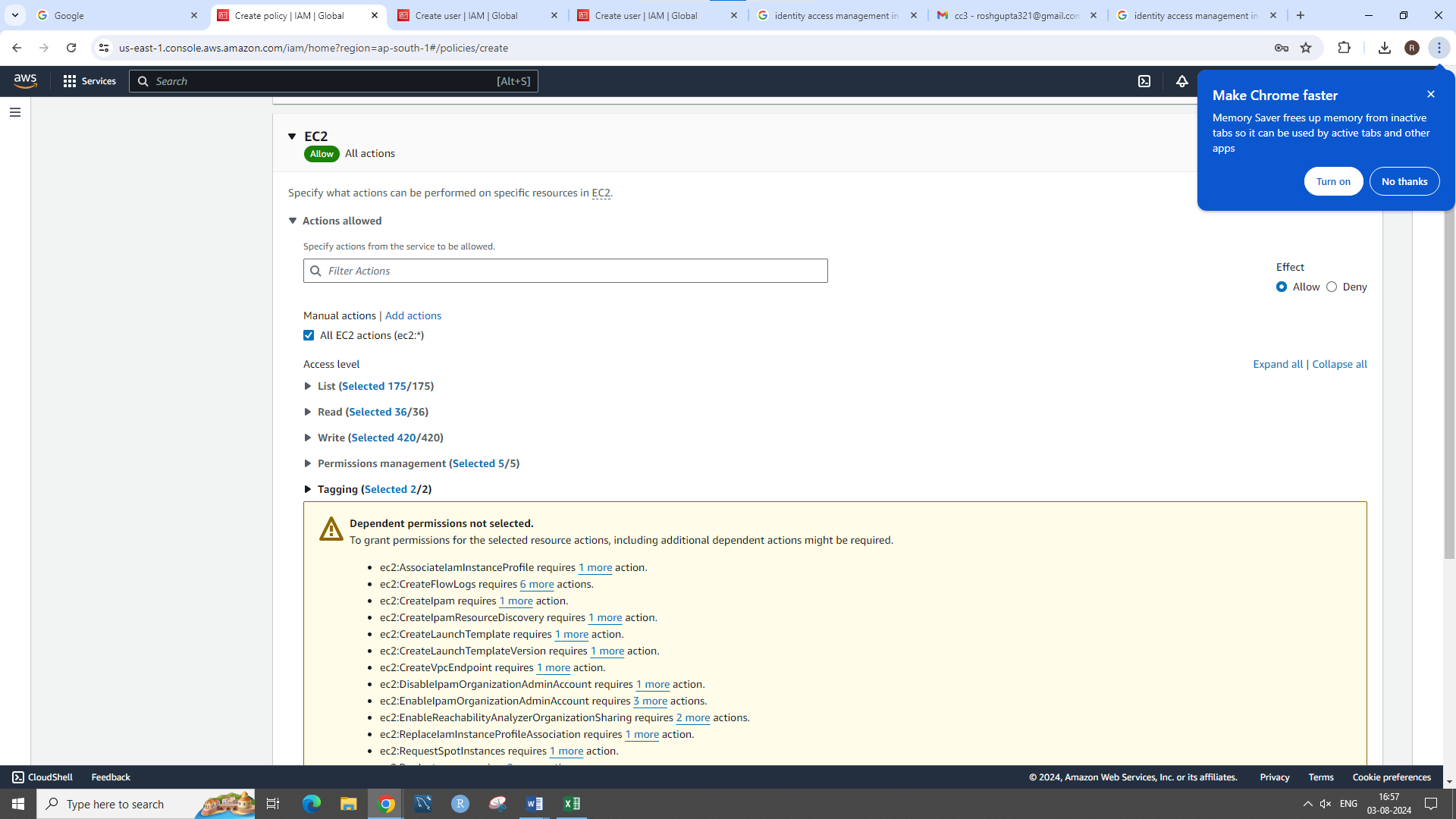


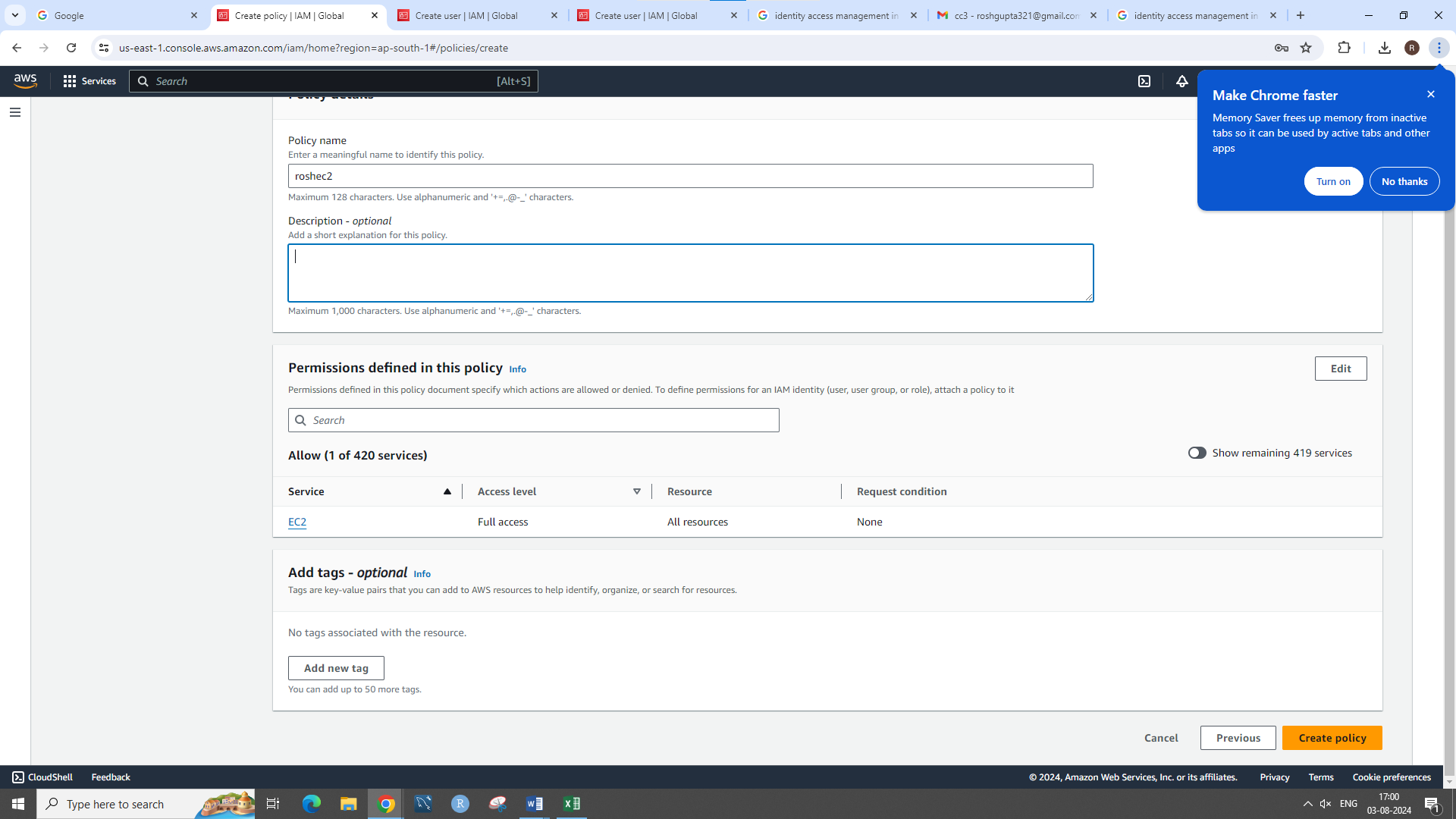


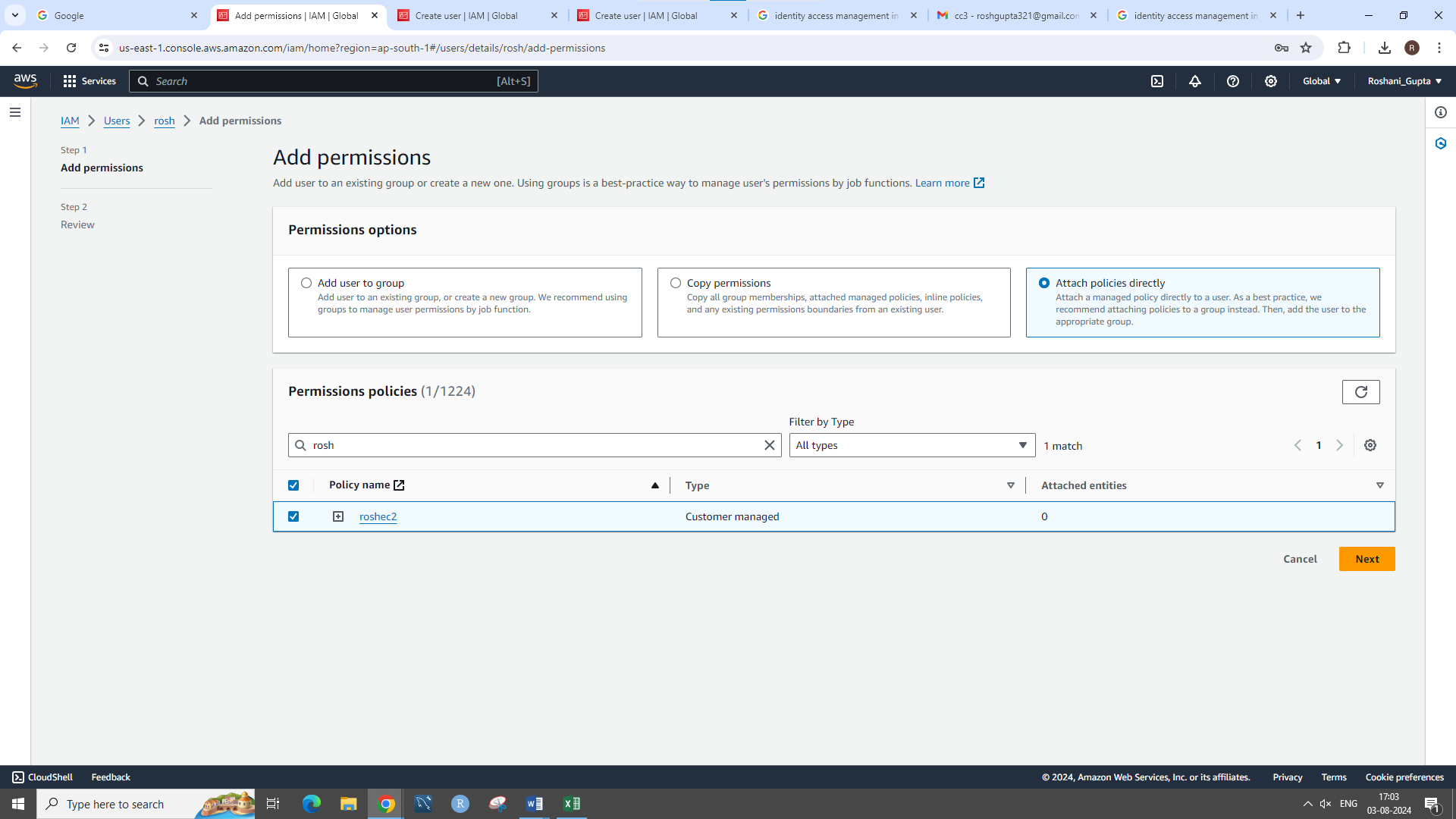


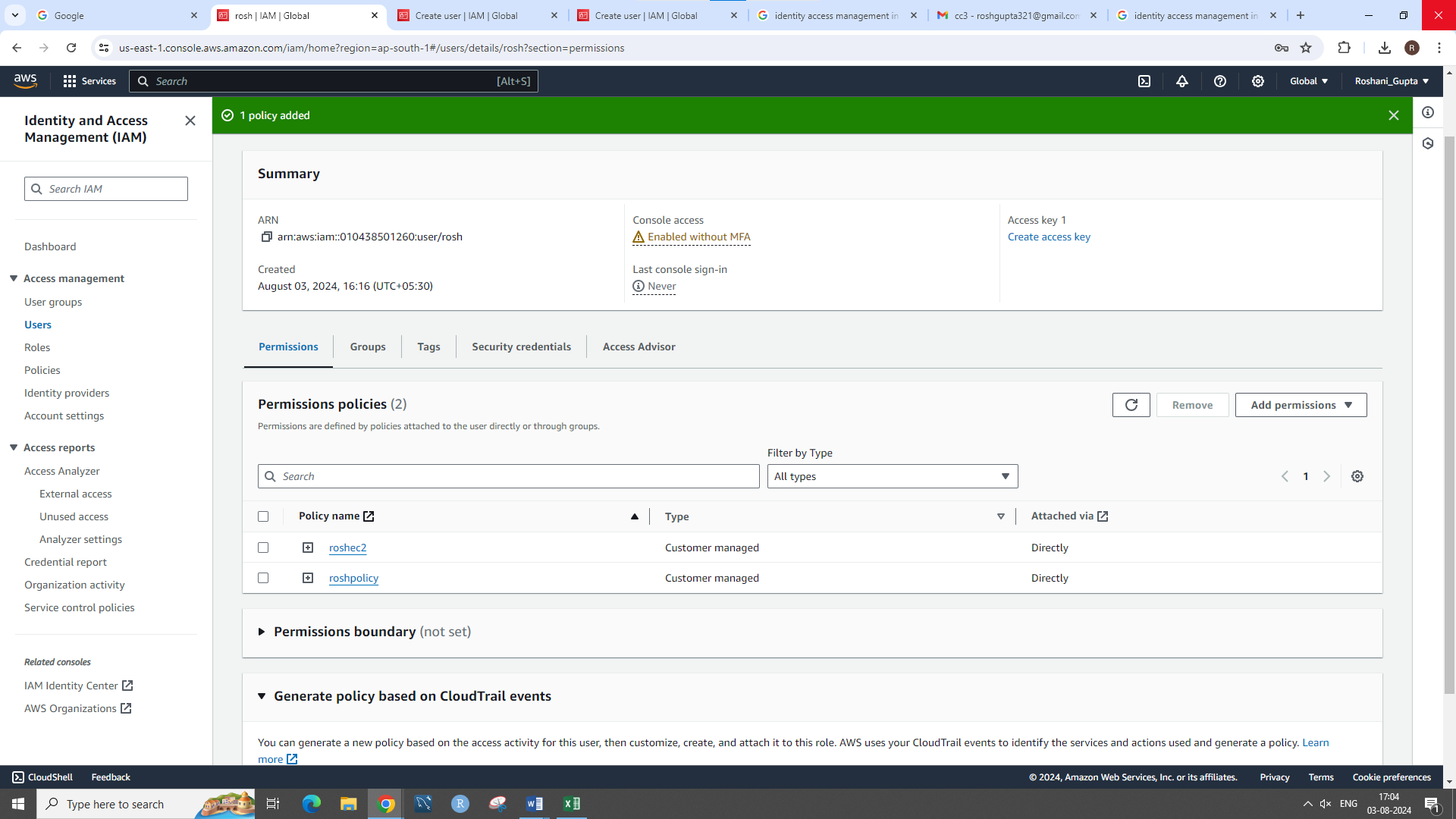


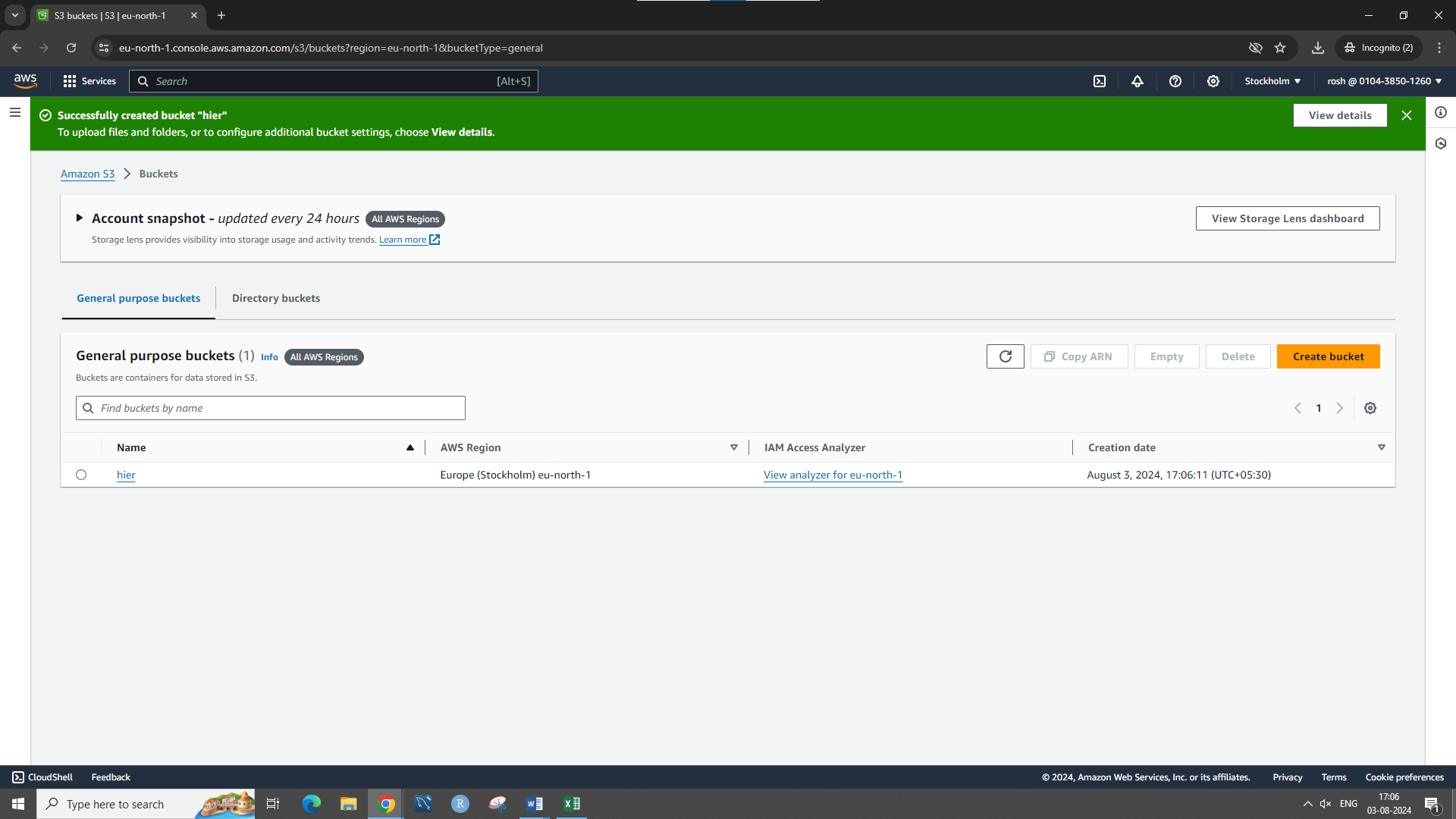


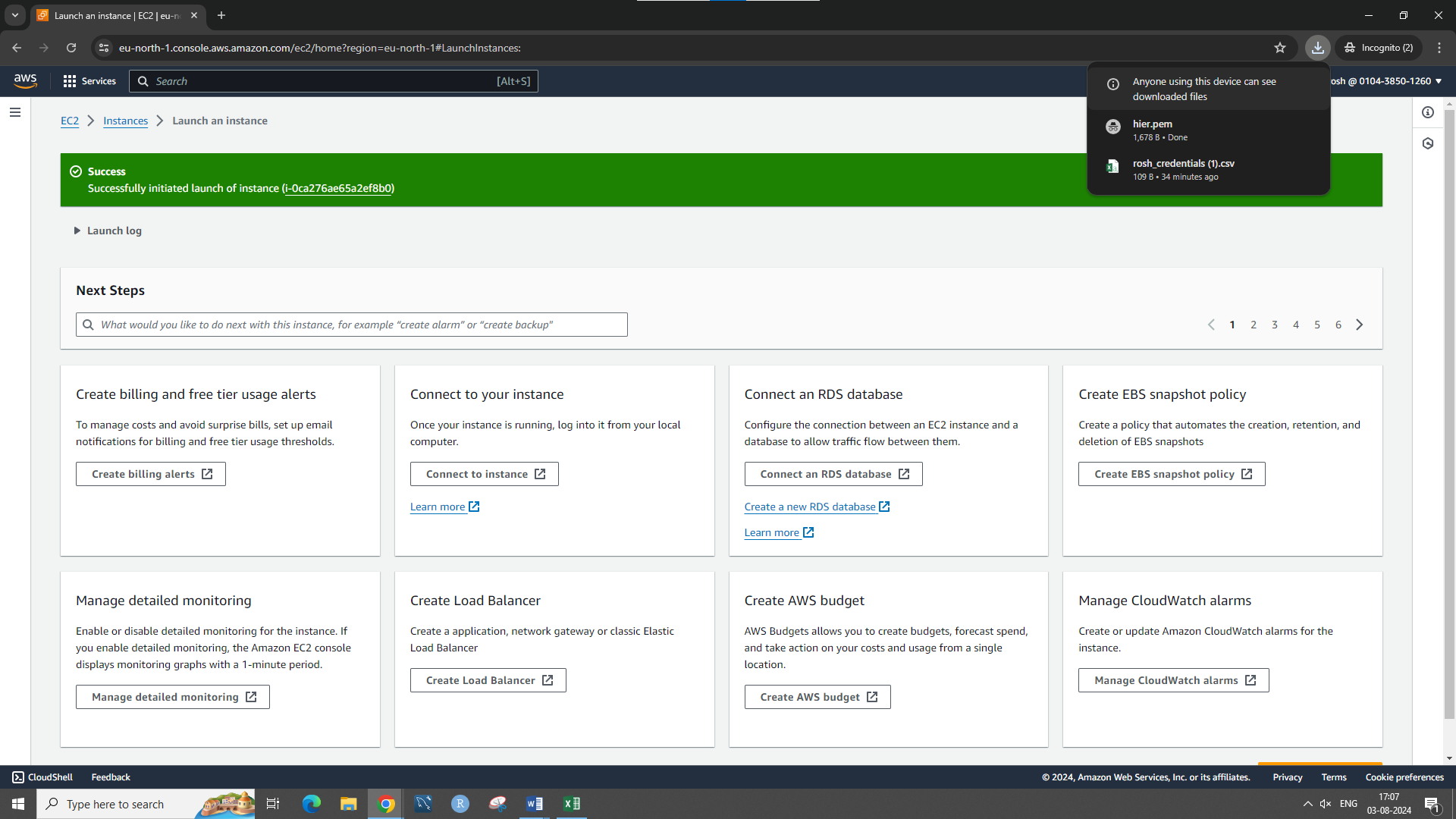


****

****

****

****

****