AWS IAM

**What is AWS Security?**

Cloud security is the highest priority in AWS. When you host your environment in the cloud, you can be assured that it’s hosted in a data center or in a network architecture that’s built to meet the requirements of the most security-sensitive organization. Additionally, this high level of security is available on a pay-as-you-go basis, meaning there is really no upfront cost, and the cost for using the service is a lot cheaper compared to an on-premises environment.

**There are many types of security services available but some of them are widely used by AWS, such as**:

IAM

Key Management System (KMS)

Cognito

Web Access Firewall (WAF)

**What is IAM?**

AWS Identity and Access Management (IAM) is a web service for securely controlling access to AWS resources. It enables you to create and control services for user authentication or limit access to a certain set of people who use your AWS resources. You can create groups and allow those users or groups to access some servers, or you can deny them access to the service.

**With IAM**, Organizations can centrally manage users, security credentials such as access keys, and permissions that control which AWS resources users can access.

**Without IAM**, Organizations with multiple users must either create multiple user accounts, each with its own billing and subscriptions to AWS products or share an account with a single security credential. Without IAM, you also don't have control about the tasks that the users can do.

**How Does IAM Work?**

The IAM workflow includes the following six elements:

🡪**A principal** is an entity that can perform actions on an AWS resource. A user, a role or an application can be a principal.

**🡪Authentication** is the process of confirming the identity of the principal trying to access an AWS product. The principal must provide its credentials or required keys for authentication.

**🡪Request:** A principal sends a request to AWS specifying the action and which resource should perform it.

**🡪Authorization**: By default, all resources are denied. IAM authorizes a request only if all parts of the request are allowed by a matching policy. After authenticating and authorizing the request, AWS approves the action.

🡪**Actions** are used to view, create, edit or delete a resource.

**🡪Resources**: A set of actions can be performed on a resource related to your AWS account.

**Components of IAM**

There are other basic components of IAM. First, we have the user; many users together form a group. Policies are the engines that allow or deny a connection based on policy. Roles are temporary credentials that can be assumed to an instance as needed.

**Users**

An IAM user is an identity with an associated credential and permissions attached to it. This could be an actual person who is a user, or it could be an application that is a user. With IAM, you can securely manage access to AWS services by creating an IAM user name for each employee in your organization. Each IAM user is associated with only one AWS account. By default, a newly created user is not authorized to perform any action in AWS. The advantage of having one-to-one user specification is that you can individually assign permissions to each user.

**Groups**

A collection of IAM users is an IAM group. You can use IAM groups to specify permissions for multiple users so that any permissions applied to the group are applied to the individual users in that group as well. Managing groups is quite easy. You set permissions for the group, and those permissions are automatically applied to all the users in the group. If you add another user to the group, the new user will automatically inherit all the policies and the permissions already assigned to that group. This lessens the administrative burden.

**Policies**

An IAM policy sets permission and controls access to AWS resources. Policies are stored in AWS as JSON documents. Permissions specify who has access to the resources and what actions they can perform. For example, a policy could allow an IAM user to access one of the buckets in Amazon S3.

The policy would contain the following information:

🡪Who can access it

🡪What actions that user can take

🡪Which AWS resources that user can access

🡪When they can be accessed

There are two types of policies: managed policies and inline policies.

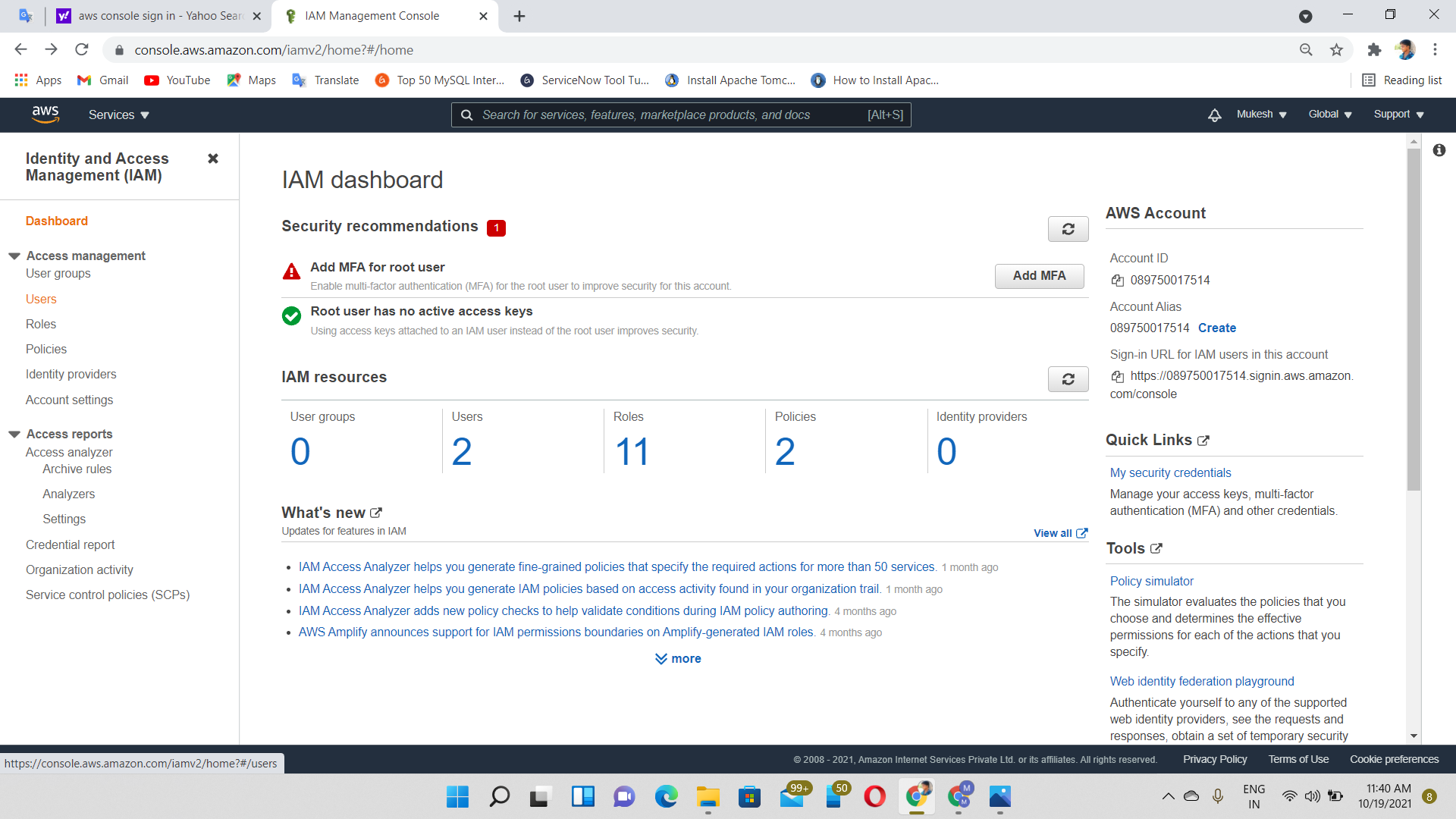
**A managed policy** is a default policy that you attach to multiple entities (users, groups, and roles) in your AWS account. Managed policies, whether they are AWS-managed or customer-managed, are stand-alone identity-based policies attached to multiple users and/or groups.

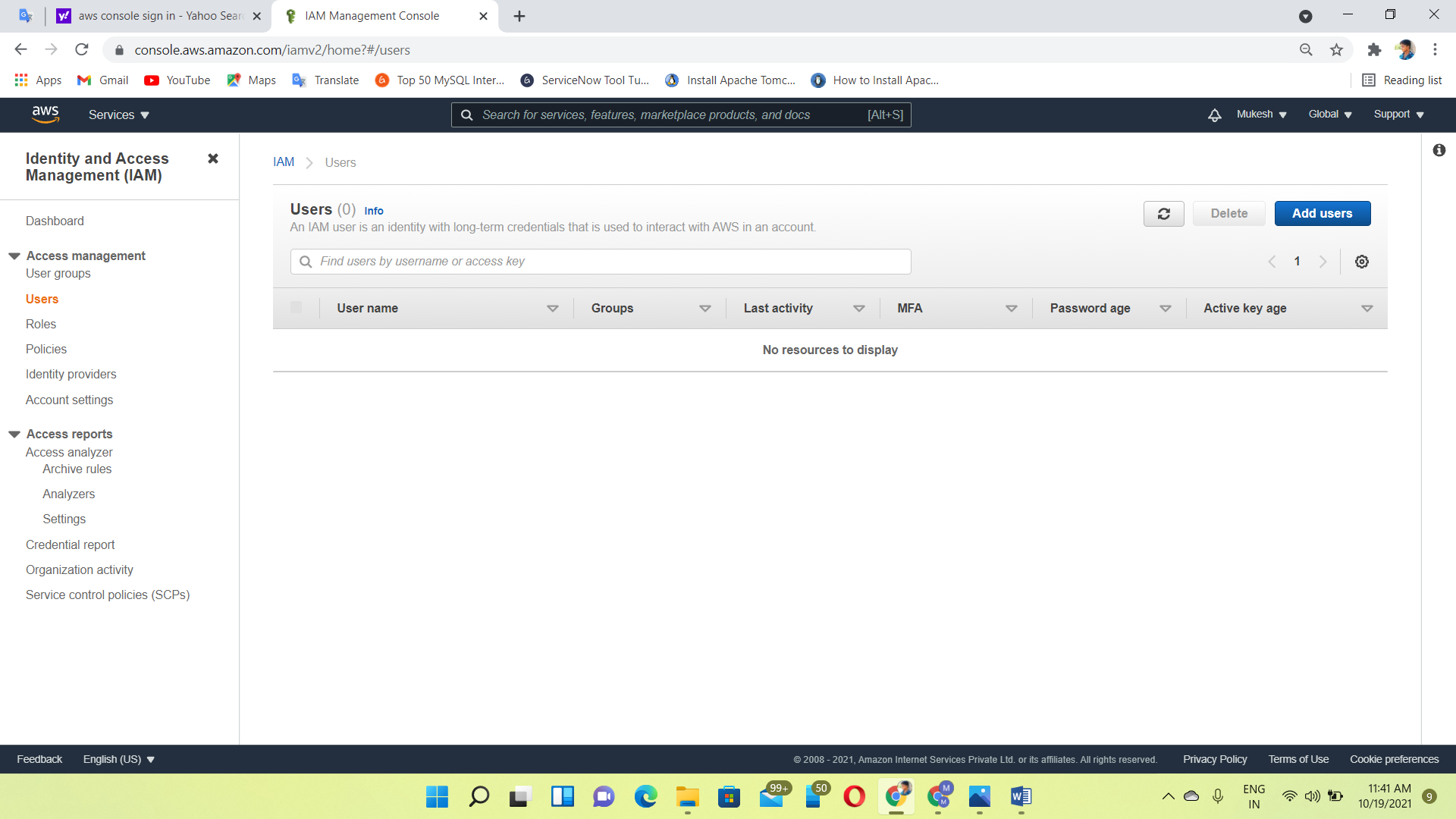
**Inline policies** are policies that you create that are embedded directly into a single entity (user, group or role).

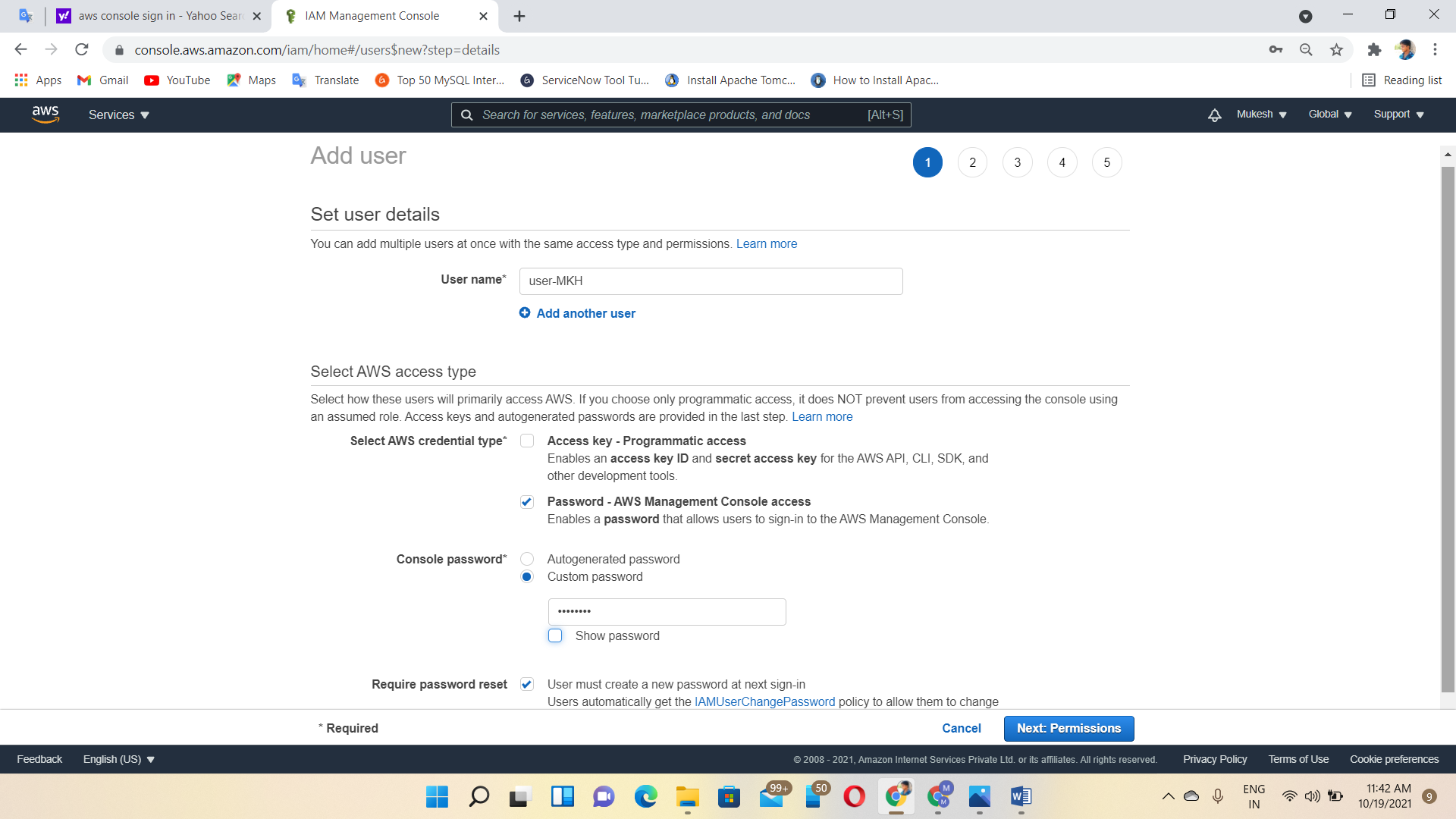
**Roles**

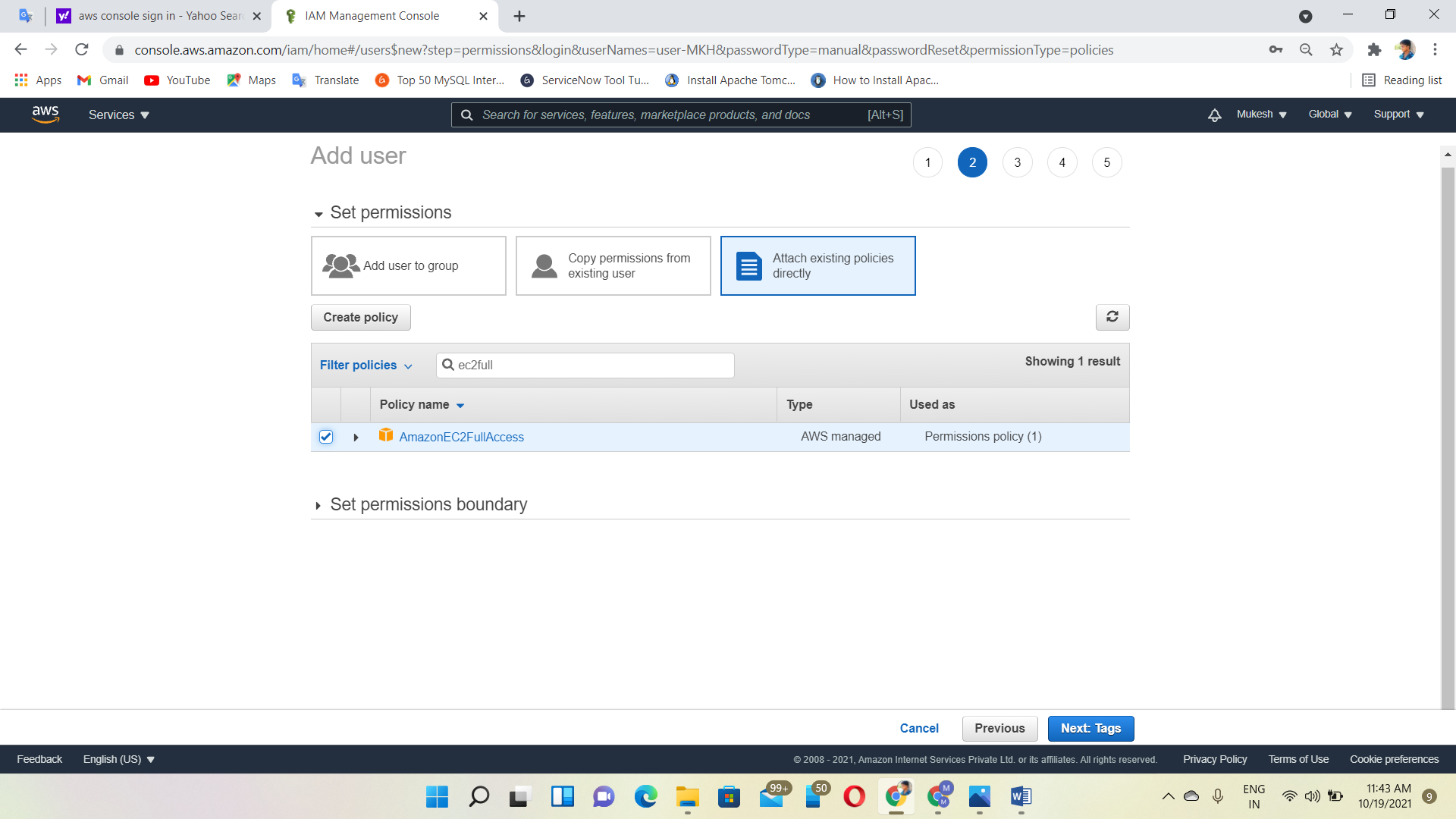
An IAM role is a set of permissions that define what actions are allowed and denied by an entity in the AWS console. It is similar to a user in that it can be accessed by any type of entity (an individual or AWS service). Role permissions are temporary credentials.

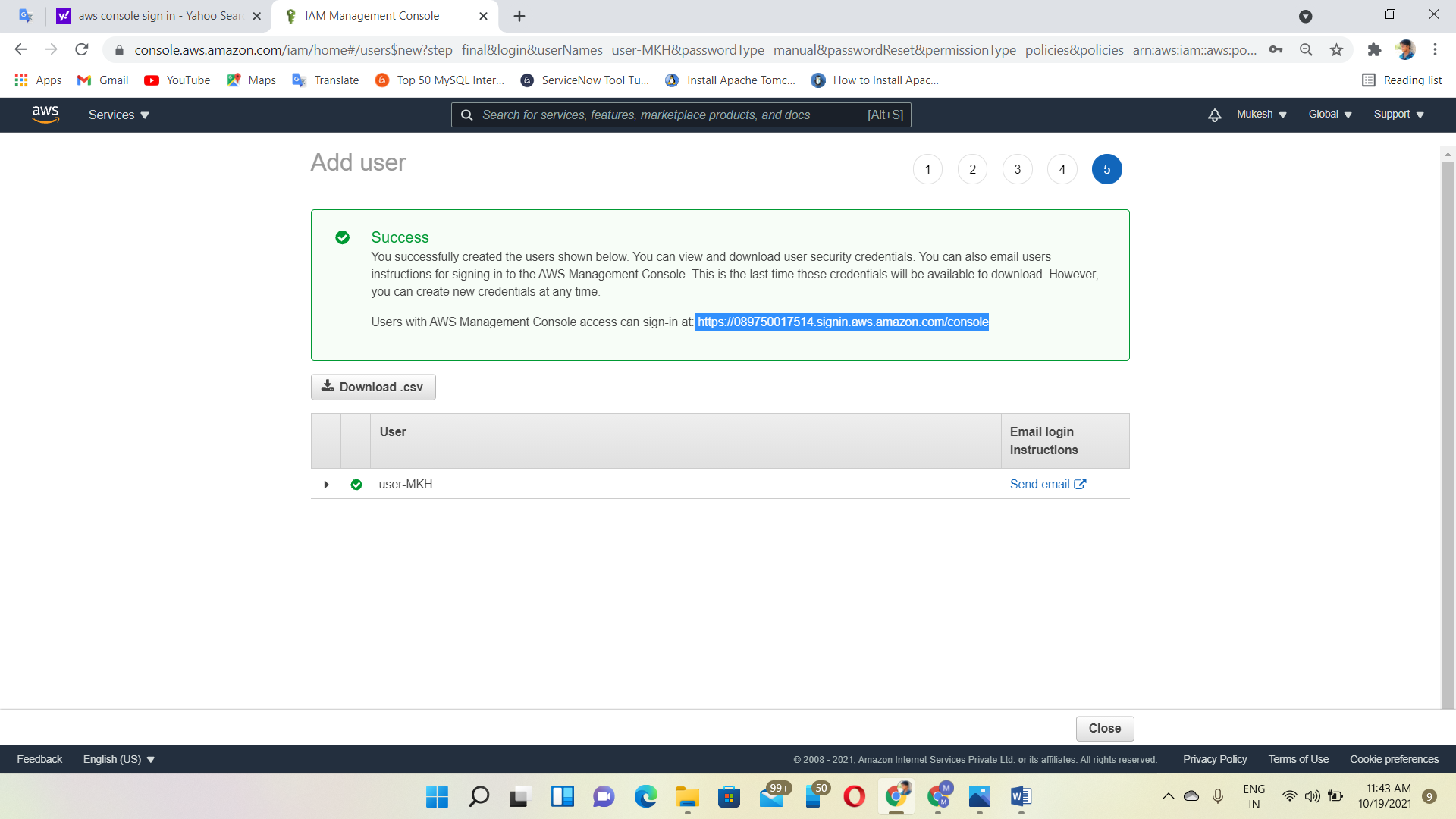
**IAM User Identity**

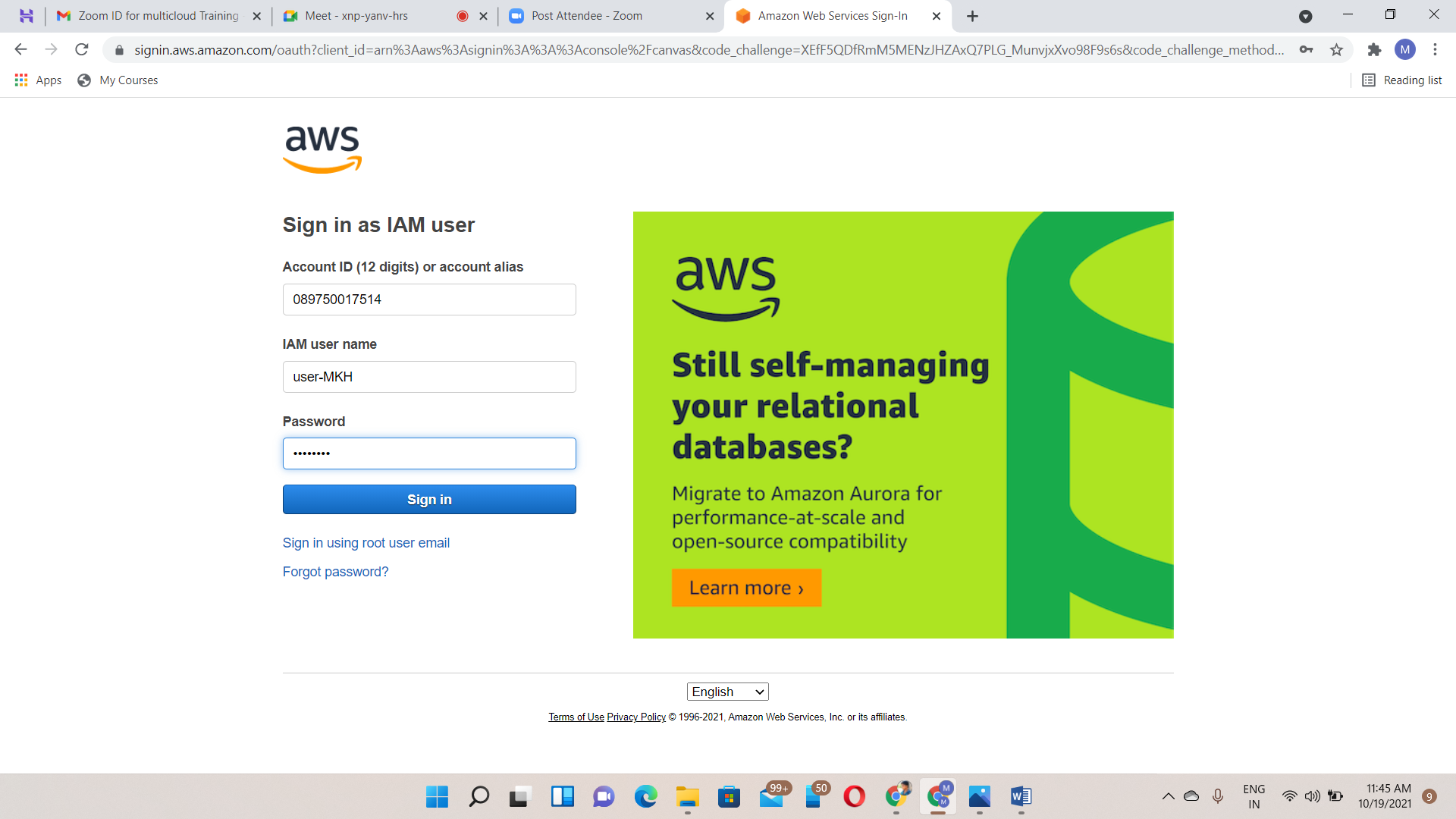












**Features of IAM**

Shared access to the AWS account

Granular permissions.

Multifactor authentication (MFA)

Identity Federation.

Free to use

Password policy

**Limitations**

User per Account 🡪 5000

Roles per Account 🡪 1000

Groups per Account 🡪 300

Policy

Inline Policy🡪No limitation. But, size is limited

User - 2048 characters

Group - 5120 characters

Roles - 10248 characters

Managed Policy 🡪 User (or) Group (or) Role 🡪 6144 characters.