

AWS IAM Hands-On

Shivam9joshi@gmail.com

<https://www.linkedin.com/in/shivamgjoshi>

What is AWS IAM?

AWS IAM, a web service, facilitates secure control over access to AWS resources by centrally managing permissions for users. IAM enables the management of authentication (signing in) and authorization (permissions) for accessing AWS resources.

1. **Purpose:** IAM allows centralized management of permissions for controlling user access to AWS resources.

2. **Initial Access:** Upon creating an AWS account, a single sign-in identity, known as the AWS account root user, is provided with complete access to all AWS services and resources. Access is granted using the email address and password used during the account creation.

3. **Root User Caution:** It is advised against using the root user for routine tasks due to security reasons. The root user's credentials should be safeguarded and utilized only for tasks exclusive to the root user.

AWS IAM streamlines access control to AWS resources by providing a centralized platform for managing permissions. While the root user initially possesses complete access, it is recommended to avoid regular usage of the root user credentials for enhanced security measures. Instead, IAM allows for the creation and management of user identities with tailored permissions, ensuring secure and controlled access to AWS resources.

IAM features

- **Shared access to your AWS account:** You can grant other people permission to administer and use resources in your AWS account without having to share your password or access key.
- **Granular permissions:** You can grant different permissions to different people for different resources. For example, you might allow some users complete access to Amazon Elastic Compute Cloud (Amazon EC2), Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB, Amazon Redshift, and other AWS services. For other users, you can allow read-only access to just some S3 buckets, or permission to administer just some EC2 instances, or to access your billing information but nothing else.
- **Secure access to AWS resources for applications that run on Amazon EC2:** You can use IAM features to securely provide credentials for applications that run on EC2 instances. These credentials provide permissions for your application to access other AWS resources. Examples include S3 buckets and DynamoDB tables.
- **Multi-factor authentication (MFA):** You can add two-factor authentication to your account and to individual users for extra security. With MFA you or your users must provide not only a password or access key to work with your account, but also a code from a specially configured device. If you already use a FIDO security key with other services, and it has an AWS supported

configuration, you can use WebAuthn for MFA security. For more information, see [Supported configurations for using FIDO security keys](#).

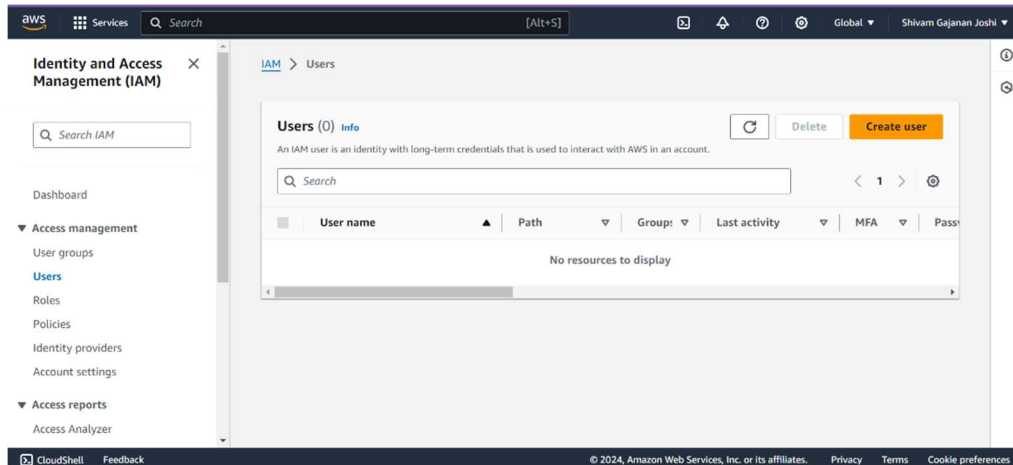
- **Identity federation:** You can allow users who already have passwords elsewhere—for example, in your corporate network or with an internet identity provider—to get temporary access to your AWS account.
- **Identity information for assurance:** If you use AWS CloudTrail, you receive log records that include information about those who made requests for resources in your account. That information is based on IAM identities.
- **PCI DSS Compliance:** IAM supports the processing, storage, and transmission of credit card data by a merchant or service provider, and has been validated as being compliant with Payment Card Industry (PCI) Data Security Standard (DSS). For more information about PCI DSS, including how to request a copy of the AWS PCI Compliance Package, see [PCI DSS Level 1](#).
- **Integrated with many AWS services:** For a list of AWS services that work with IAM, see [AWS services that work with IAM](#).
- **Eventually Consistent:** IAM, like many AWS services, is eventually consistent, ensuring high availability by replicating data across multiple servers globally. However, changes such as creating or updating users, groups, roles, or policies may take time to propagate. To mitigate this, avoid including IAM changes in critical code paths; instead, execute them in separate setup routines and verify their propagation before production use.
- **Free to use:** AWS Identity and Access Management (IAM) and AWS Security Token Service (AWS STS) are features of your AWS account offered at no additional charge. You are charged only when you access other AWS services using your IAM users or AWS STS temporary security credentials. For information about the pricing of other AWS products, see the [Amazon Web Services pricing page](#).

AWS Root User: The AWS Root User is the first cloud service identity created by default when you create your cloud service provider account.

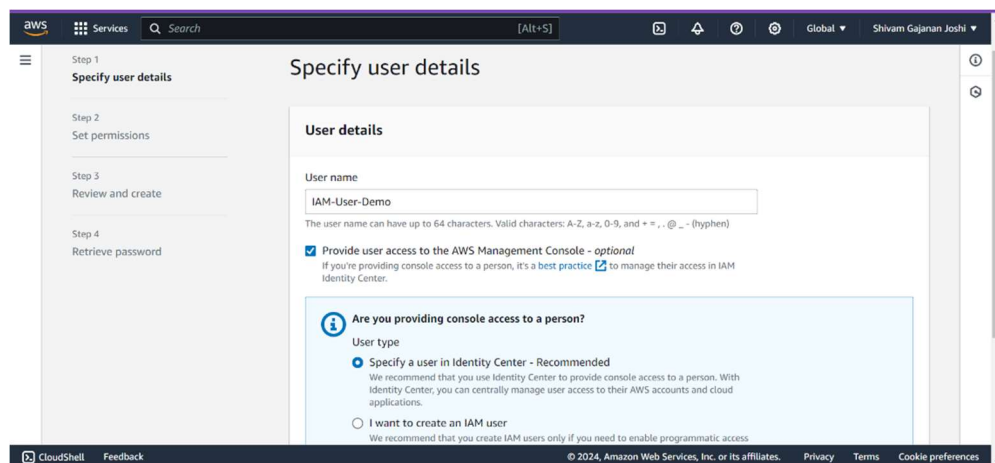
AWS IAM User: An AWS IAM User can be created by a root user or another IAM user who has entitlements to create additional IAM users.

HANDS-ON

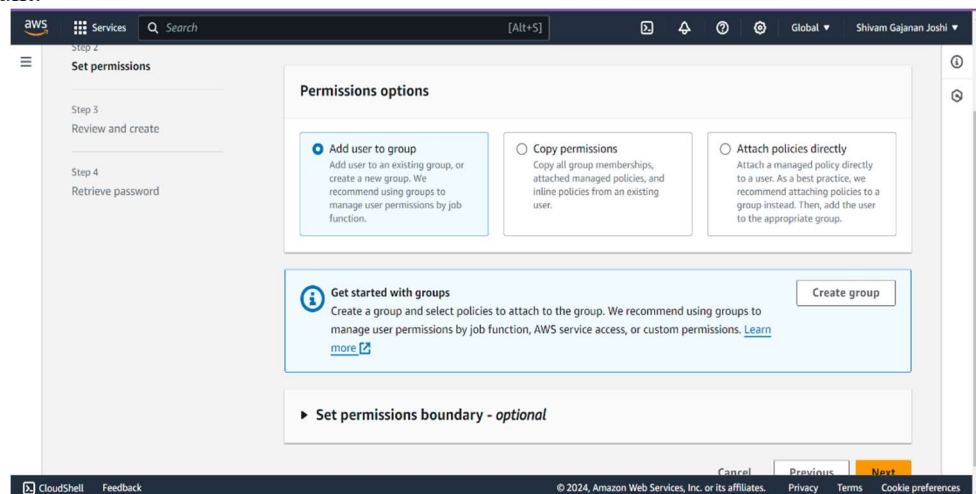
1. Go to AWS Management console
2. Search for IAM role in the search bar



3. Select IAM and create a user by filling up all the fields like Username

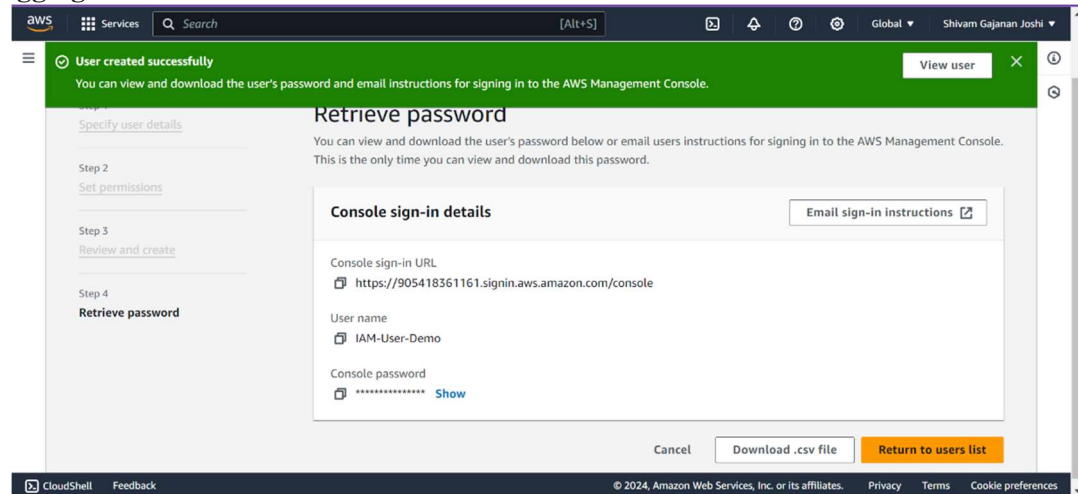


4. At this time we won't set any permission just to check what happens when we access the user account.

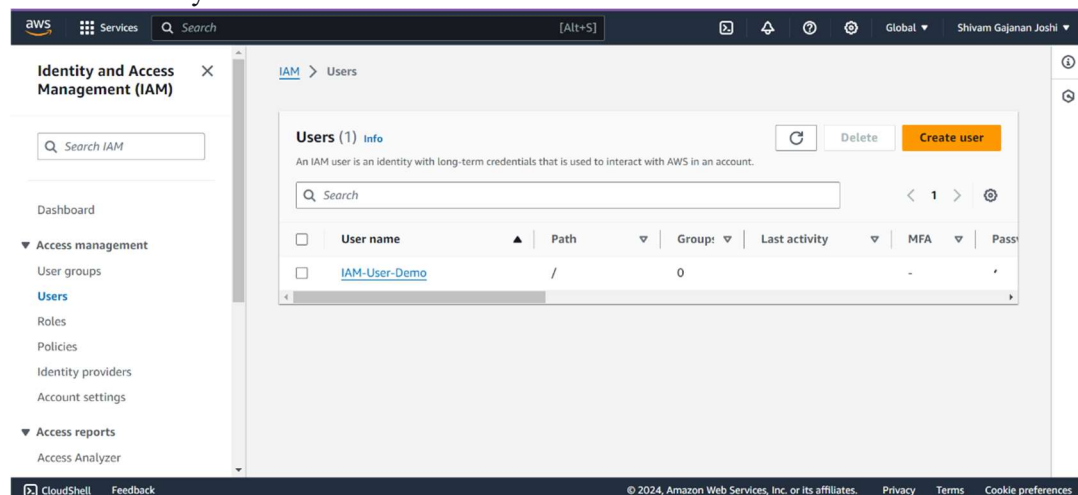


5. Review and create the user.

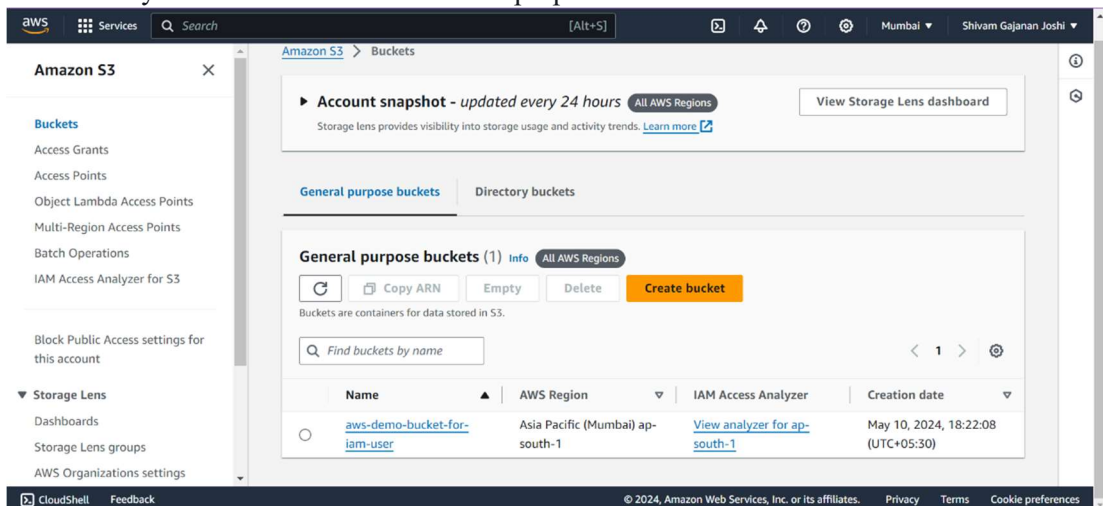
Note: Please copy the IAM username, Password and Account ID. We will need it while logging in.



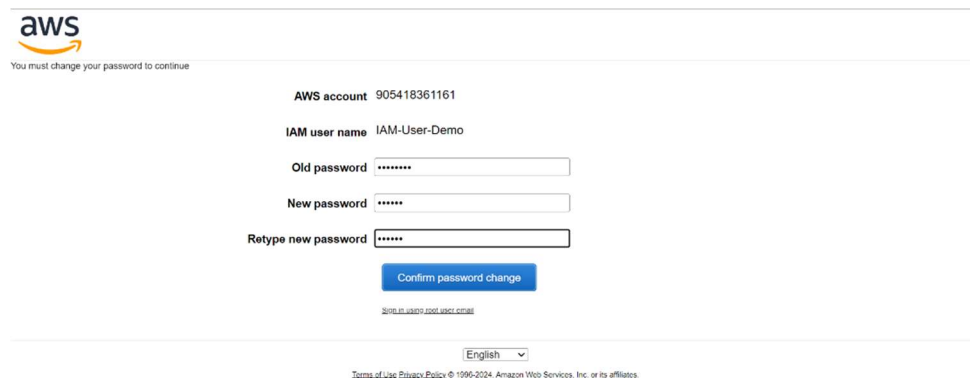
6. Review the newly created user



7. I had already created a S3 bucket for demo purpose.

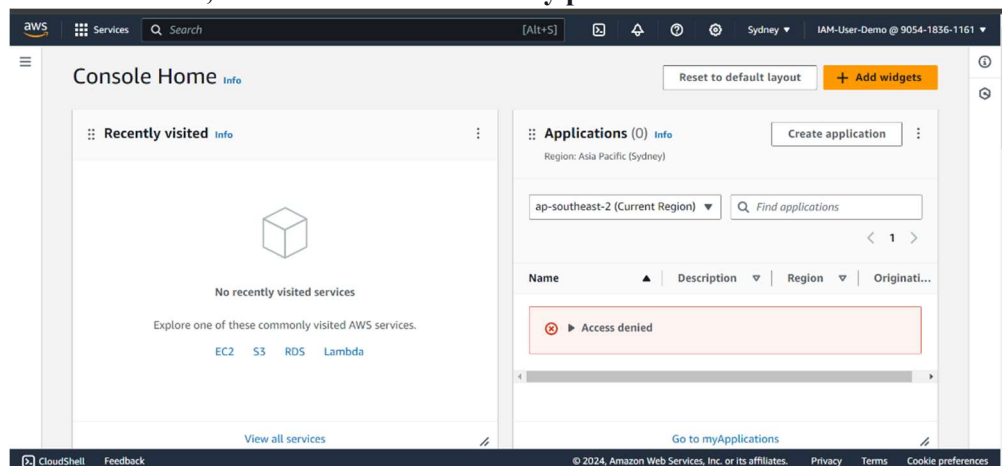


8. Now, open incognito window and login into the IAM User account using the credentials

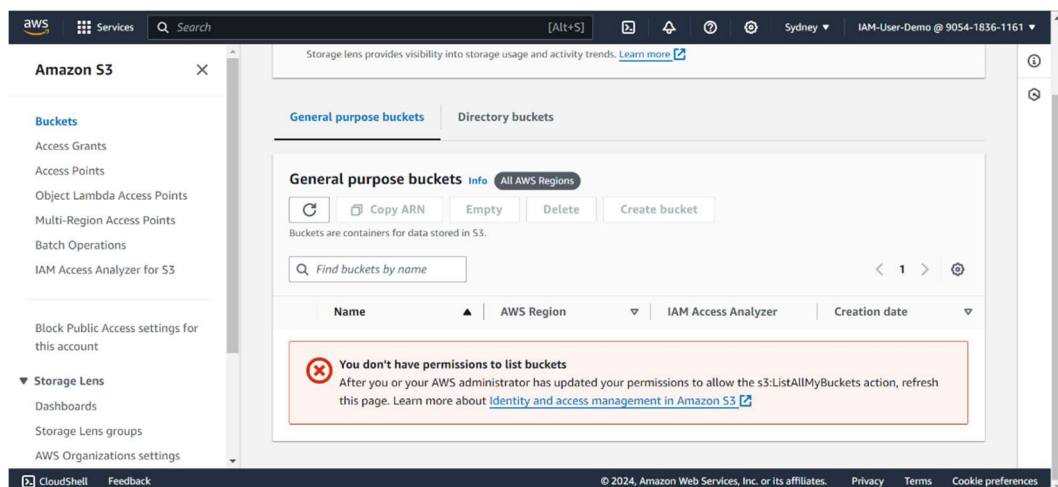


The screenshot shows the AWS IAM console login page. At the top, the AWS logo is followed by the text "You must change your password to continue". Below this, the "AWS account" is listed as 905418361161. The "IAM user name" is IAM-User-Demo. There are three password fields: "Old password", "New password", and "Retype new password", each with a masked input field. A blue "Confirm password change" button is located below the password fields. At the bottom, there is a link to "Sign in with your MFA device" and a language dropdown menu set to "English".

9. When we access the IAM user account, we can see that we don't have any permission to access the resources, since we did not attach any policies.

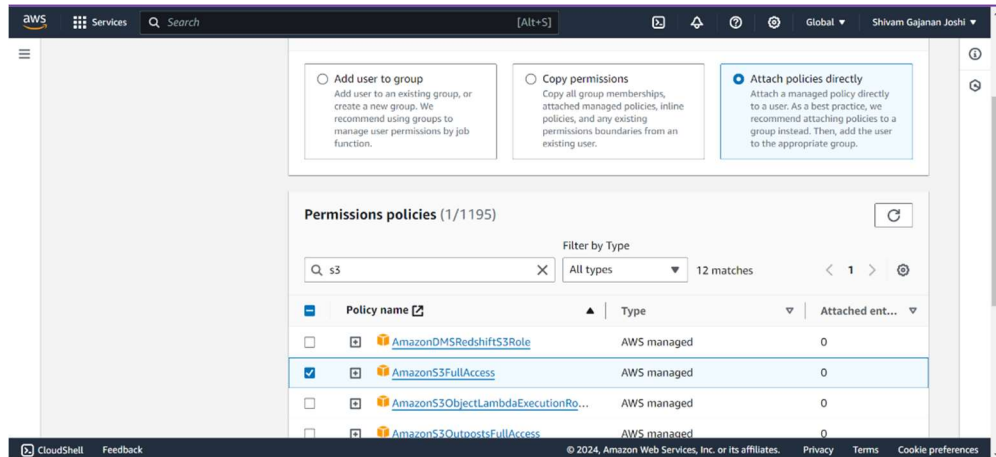


10. Same with S3 bucket.

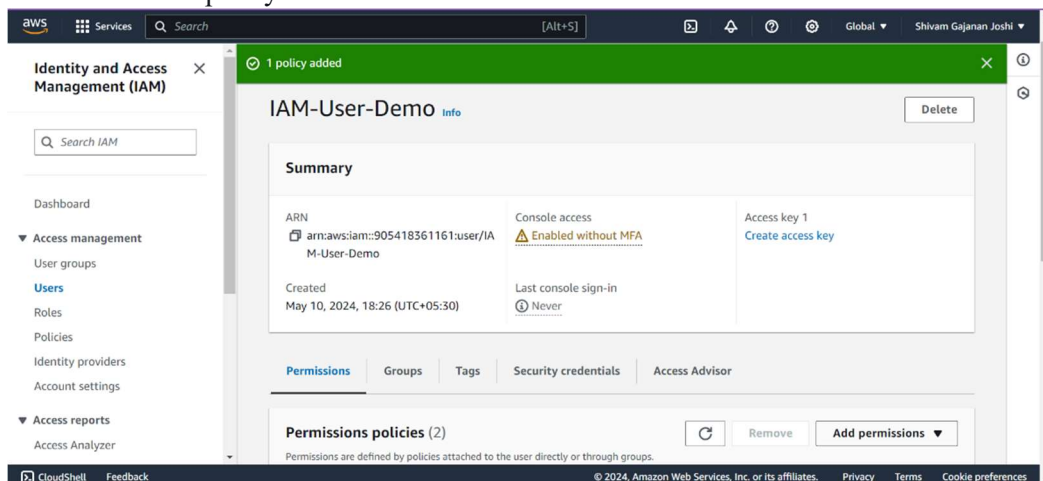


11. Now, go back to your root account and go to IAM, select the newly created user, and click on add policies.

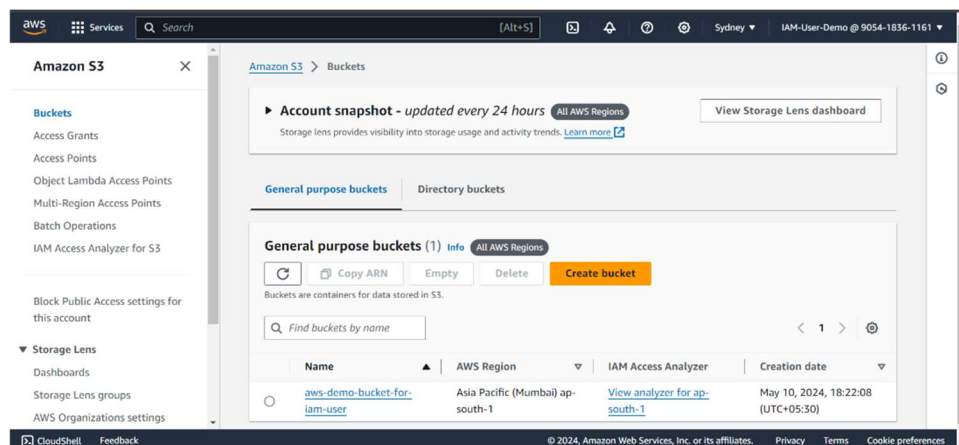
We need to add policies for S3, so search S3 and select “**AmazonS3FullAccess**” to give full access to the IAM user



12. We have added the policy. Now it's time to check the IAM user account.



13. Hurray! Our IAM User has now got the access to S3 bucket.



14. Our lab is now completed, so we can now delete the resource.

Note that we can also see the last activity of the IAM user, we can integrate it with other AWS services to get logs and monitor the user activity.

