

# Lab 1: AWS Identity and Access Management | Monarch Nigam

## Exercise 6.1: Lock Down the Root User -

1. I have created a regular user named AdminUserMonarchy and then assigned it the AdministratorAccess policy as shown in the below screenshot. My email is [cloud.class.service@gmail.com](mailto:cloud.class.service@gmail.com) and my account name is monarch.smuclass.2025

The screenshot shows the AWS IAM User details page for 'AdminUserMonarchy'. The 'Summary' section displays the ARN (arn:aws:iam::160885292183:user/AdminUserMonarchy), which is enabled without MFA. It also shows the creation date (February 02, 2025) and the last console sign-in (Never). The 'Permissions' tab is selected, showing one attached policy: 'AdministratorAccess'. A note indicates that no permissions boundary is set.

2. I have made sure that there are no active access keys associated with my root account, as shown in the below screenshot

The screenshot shows the AWS IAM Account details page for the account 'monarch.smuclass.2025'. The 'Account details' section shows the account name, email address (cloud.class.service@gmail.com), and canonical user ID. The 'Multi-factor authentication (MFA)' section indicates no MFA devices are assigned. The 'Access keys' section shows that there are no active access keys.

3. I have Enabled MFA for the root account to confirm a user's identity, as you can see in the below screenshot

The screenshot shows the 'My security credentials' page for a 'Root user'. At the top, there are navigation icons and a dropdown for 'Global'. Below that, a sub-navigation bar includes 'Edit account name, email, and password'.

**Account details:**

- Account name: monarch.smiclass.2025
- Email address: cloud.class.service@gmail.com
- AWS account ID: 160885292183
- Canonical user ID: 9faef398b9bcee233287acdae78382a9f39cdba33b52f5caf3f4e41fef3fb618

**Multi-factor authentication (MFA) (1)**

Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. [Learn more](#)

Type	Identifier	Certifications	Created on
Virtual	arn:aws:iam::160885292183:mfa/monarch_iphone15	Not Applicable	Sun Feb 02 2025

4. I Updated my root login to a password that's long and complex and that includes nonalphanumeric characters

The screenshot shows the 'Account Details' page for the root user. The 'Edit' button is visible at the top right.

**Account Details:**

Edit your AWS account name, root user password, or root user email address [Learn More](#)

Account name	monarch.smiclass.2025
Email address	cloud.class.service@gmail.com
Password	*****

[Return to console](#)

5. The below screenshot is a proof that I am able to login successfully-

The screenshot shows the AWS Console Home page. In the top left, under 'Recently visited', the 'IAM' service is highlighted with a blue icon. Other services like 'Billing and Cost Management', 'IAM Identity Center', and 'CloudWatch' are also listed. The top right features a search bar, a help icon, and navigation links for 'United States (N. Virginia)' and 'monarch.smiclass.2025'. Below the header, there are several cards: 'Welcome to AWS' (Getting started with AWS), 'AWS Health' (Open issues 0, Past 7 days), 'Cost and usage' (Current month costs: Data unavailable, Forecasted month end costs: Data unavailable), and 'Applications' (0). A large orange button at the bottom right says 'Create application'.

So to summarise in Exercise 6.1: I am locking down the root user by creating a regular user with **AdministratorAccess**. I am ensuring no active access keys exist and enabling **MFA** for authentication. I am updating the root password to a strong one and confirming login before securely storing it.

### Exercise 6.2: Assign and Implement an IAM Policy

- Create a new user in the IAM Dashboard- The New User is S3\_IAM\_User

The screenshot shows the IAM User details page for 'S3\_IAM\_User'. The 'Summary' section includes the ARN (arn:aws:iam::160885292183:user/S3\_IAM\_User), which is highlighted in blue. It also shows 'Console access' (Enabled without MFA) and 'Last console sign-in' (Today). The 'Permissions' tab is selected, showing 'Permissions policies (1)' attached via 'AmazonS3FullAccess'. The 'Groups' tab is also visible. At the bottom, there is a 'Permissions boundary (not set)' section.

2. Attach the AmazonS3FullAccess policy that will permit your user to create, edit, and delete S3 buckets. (In the above screenshot we can see that the AmazonS3FullAccess policy has been implemented)
3. Note the user login instructions that will be displayed - the user login is <https://1608XXXXXX3.signin.aws.amazon.com/console>, FOR S3\_IAM\_User
4. Log in as your new user and try creating a new S3 bucket. As we can see the login is successful and the S3 Bucket is created successfully with the bucket name **monarchbuckets3demo**

The screenshot shows the AWS Console Home page. On the left, under 'Recently visited', there are links to CloudWatch, IAM Identity Center, Billing and Cost Management, and IAM. In the center, the 'Applications' section shows 0 applications with a note: 'Access denied to servicecatalog>ListApplications'. Below it, the 'Cost and usage' section shows current month costs and forecasted month end costs, both with 'Access denied' status. At the bottom, there are links for CloudShell, Feedback, and cookie preferences.

## Login S3\_IAM\_USER Success

The screenshot shows the AWS Buckets page. A green banner at the top says: 'Successfully created bucket "monarchbuckets3demo". To upload files and folders, or to configure additional bucket settings, choose View details.' Below it, the 'Account snapshot' section provides visibility into storage usage and activity trends. The 'General purpose buckets' section shows one bucket named 'monarchbuckets3demo' from the 'us-east-2 (Ohio)' region, created on February 2, 2025. There are buttons for Copy ARN, Empty, Delete, and Create bucket.

5. Just to prove everything is working, try launching an EC2 instance. Your request should be denied. - The below screenshots depicts that I am not able to create the EC2 Instance which clearly means the IAM Permissions are working.

The screenshot shows the 'Launch an instance' wizard. In the 'Amazon Machine Image (AMI)' section, a specific AMI is selected: 'Amazon Linux 2023 AMI'. A warning message is displayed: 'You are not authorized to perform the operation ec2:GetAllowedImagesSettings. Without this operation we will be unable to validate your allowed AMIs. You need to grant your IAM user permission to use the GetAllowedImagesSettings API. For more information, see [Changing Permissions for an IAM User](#). Contact your AWS administrator if you need help.' Below this, another message states: 'You are not authorized to perform this operation. User: arn:aws:iam::160885292183:user/S3\_IAM\_User is not authorized to perform: ec2:GetAllowedImagesSettings because no identity-based policy allows the ec2:GetAllowedImagesSettings action'.

The screenshot shows the main EC2 dashboard. On the left, a sidebar lists navigation options like Dashboard, Instances, Images, and Network & Security. The main area features several cards: 'Resources' (listing 0 instances, 0 auto scaling groups, etc.), 'Launch instance' (with 'Launch instance' and 'Migrate a server' buttons), 'Service health' (showing an error: 'An error occurred An error occurred retrieving service health information'), 'Zones' (Zone name and Zone ID fields), and 'Account attributes' (with a 'View all AWS Free Tier offers' link). A prominent red box highlights the 'Service health' card's error message, which is identical to the one shown in the launch wizard above.

## EXERCISE 6.3 : Create, Use, and Delete an AWS Access Key

1. Create a new AWS access key, and save both the access key ID and secret access key somewhere secure- the new Access key has been created:

The screenshot shows the AWS IAM 'Create access key' page. At the top, there's a green banner with the message 'Access key created' and a note: 'This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.' On the left, a vertical navigation bar lists three steps: Step 1 (Access key best practices & alternatives), Step 2 - optional (Set description tag), and Step 3 (Retrieve access keys). Step 3 is currently selected and highlighted with a blue circle. The main content area is titled 'Retrieve access keys'. It shows an 'Access key' section with the key ID 'AKIASK5MCXSLWOWU3F4M' and a 'Secret access key' section with a redacted value. Below this is an 'Access key best practices' section with a bulleted list: 'Never store your access key in plain text, in a code repository, or in code.', 'Disable or delete access key when no longer needed.', 'Enable least-privilege permissions.', and 'Rotate access keys regularly.' At the bottom right are 'Download .csv file' and 'Done' buttons.

2. Enter aws configure at your local command line to add the key to your AWS CLI configuration- as shown in the screenshot I have configured AWS CLI in my terminal.

The screenshot shows a macOS terminal window titled 'monarchnigam — zsh — 80x34'. The user runs the command 'aws configure'. The output shows the configuration process:  
Last login: Sun Feb 2 20:13:12 on ttys000  
[monarchnigam@Monarchs-Air ~ % aws configure  
AWS Access Key ID [\*\*\*\*\*3]: AKIASK5MCXSLWOWU3F4M  
AWS Secret Access Key [\*\*\*\*\*3F4M]: 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  
Default region name [us-east-1]: us-east-1  
Default output format [json]: json  
monarchnigam@Monarchs-Air ~ % aws s3 ls  
2025-02-02 20:36:01 monarchs3bucket  
monarchnigam@Monarchs-Air ~ %

3. Try performing some operation—such as listing your S3 buckets - in the above screen shot, I have listed the s3 buckets, since there is only one bucket named monarch3bucket, it gets listed here.

4. In the console, disable (select Make Inactive) or delete the key you just created from the IAM Dashboard.- as shown in the below screenshot the access key has been deactivated

The screenshot shows the AWS IAM Access Key Deactivated page. At the top, there are two green success notifications: "Access Key deactivated" and another "Access Key deactivated". Below these, there is a table with two rows of information:

ARN arn:aws:iam::160885292183:user/S3_IAM_User	Console access ⚠️ Enabled without MFA	Access key 1 AKIASK5MCXSLWOWU3F4M - Inactive 🕒 Never used. Created today.
Created February 02, 2025, 18:15 (UTC-06:00)	Last console sign-in 🕒 Today	Access key 2 <a href="#">Create access key</a>

Below the table, there are tabs for "Permissions", "Groups", "Tags (1)", "Security credentials", and "Last Accessed". The "Permissions" tab is currently selected.

5. Confirm that you are now unable to administer your S3 buckets using the key- in the below screenshot we can see the Access key not exist error is thrown-

The screenshot shows a terminal window titled "monarchnigam -- zsh -- 80x34". The user has run the "aws configure" command, which has stored their AWS Access Key ID and Secret Access Key. The user then attempts to list S3 buckets with the command "aws s3 ls", but receives an error message:

```
Last login: Sun Feb  2 20:13:12 on ttys000
[monarchnigam@Monarchs-Air ~ % aws configure
AWS Access Key ID [*****3]: AKIASK5MCXSLWOWU3F4M
AWS Secret Access Key [*****3F4M]: gPDHiCtofEYZV6s+kBJ1c4u3xSGCrv0I2R
7/8L3i
Default region name [us-east-1]: us-east-1
Default output format [json]: json
monarchnigam@Monarchs-Air ~ % aws s3 ls

2025-02-02 20:36:01 monarchs3bucket
monarchnigam@Monarchs-Air ~ % aws s3 ls

An error occurred (InvalidAccessKeyId) when calling the ListBuckets operation: The AWS Access Key Id you provided does not exist in our records.
monarchnigam@Monarchs-Air ~ %
```

## Exercise 6.2: Create and Configure an IAM Group

1. Make sure you have at least two IAM users in your account.

The screenshot shows the AWS IAM 'Users' page with three entries:

User name	Path	Groups	Last activity	MFA	Password age	Console last sign-in
<a href="#">AdminUserMonarchy</a>	/	0	-	-	3 hours	-
<a href="#">S3_IAM_User</a>	/	0	17 minutes ago	-	20 minutes	February 02, 2025, 20:58
<a href="#">S3_IAM_USER_2</a>	/	0	-	-	29 minutes	-

2. Create a new IAM group and attach at least one policy—perhaps IAMUserChangePassword.

The screenshot shows the AWS IAM 'Groups' page with a single entry:

User group name	Creation time	ARN
S3_User_Group_Monarch	February 02, 2025, 20:58 (UTC-06:00)	arn:aws:iam::160885292183:group/S3_User_Group_Monarch

Below the table, there are tabs for 'Users' (2), 'Permissions', and 'Access Advisor'. The 'Permissions' tab is selected, showing the 'Permissions policies' section with one policy attached:

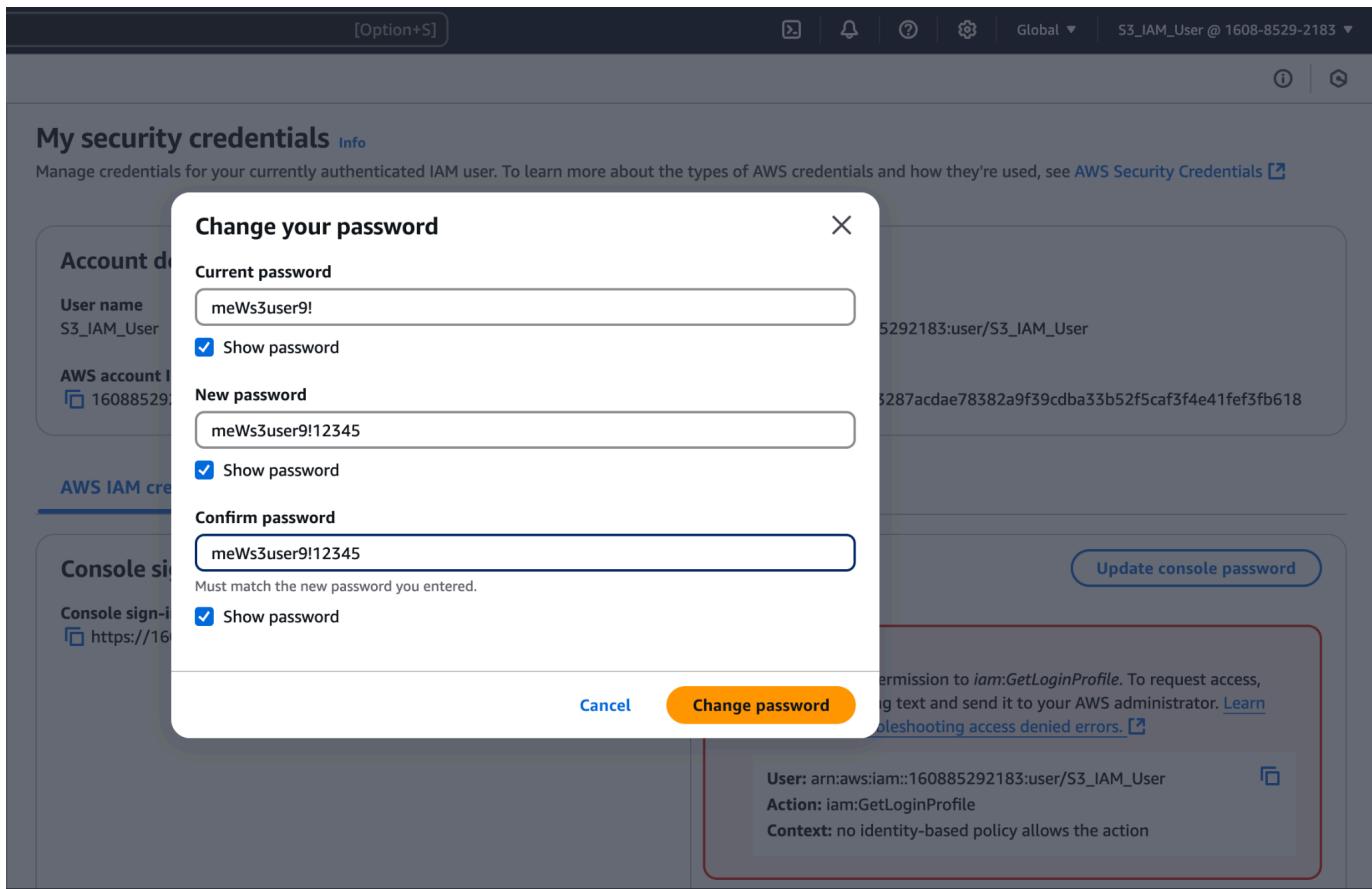
**Permissions policies (1) [Info](#)**

You can attach up to 10 managed policies.

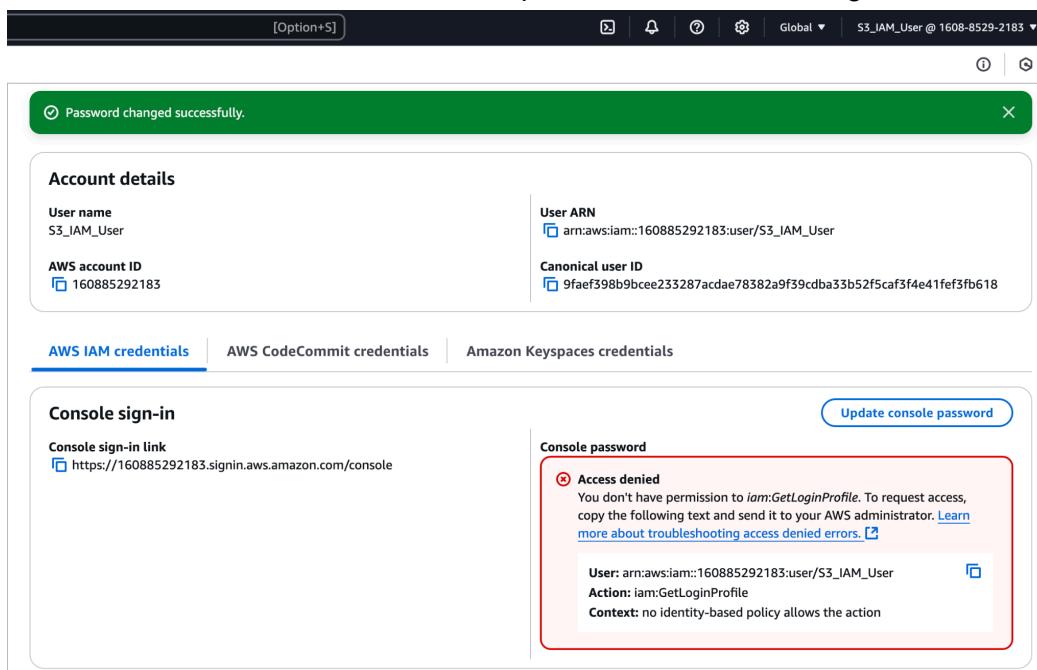
Filter by Type		
Search	All types	
<input type="checkbox"/> Policy name <a href="#">IAMUserChangePassword</a>	Type	Attached entities
<input type="checkbox"/> <a href="#">IAMUserChangePassword</a>	AWS managed	1

3. Add your two users to the group. - I have added the last two Users to the group.

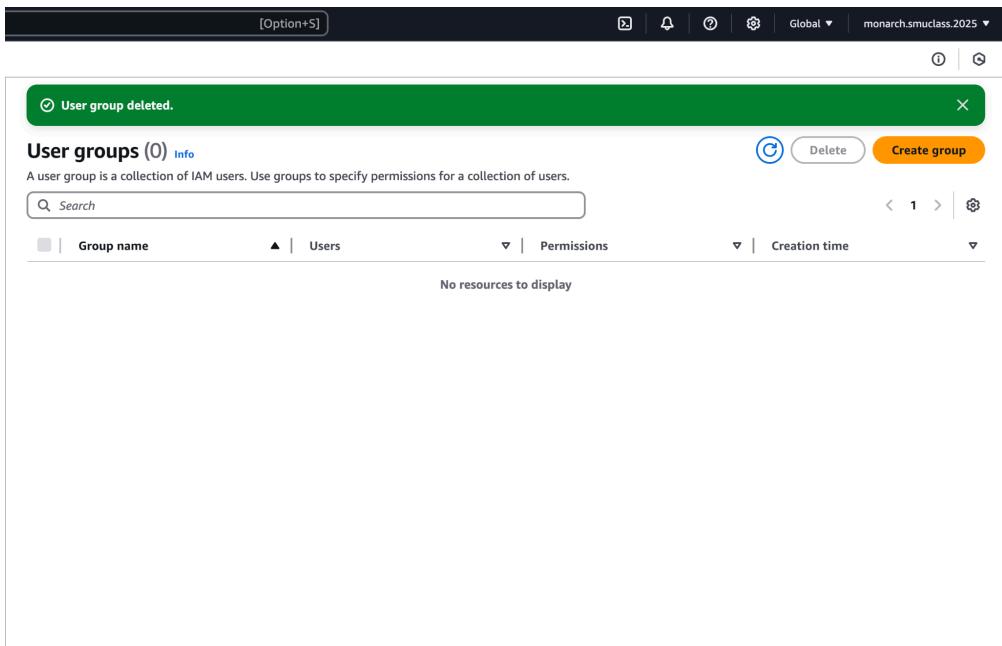
4. Confirm that your users can now change their own passwords- As you can see in the below screenshot I am able to change password as one of the user S3\_IAM\_User -



In the below screenshot we can see the password has been changed successfully



5. Delete the group or change its policies and then confirm that your users can no longer update their passwords- In the below screenshot i have deleted the USER GROUP



Now I am trying to change the password again in one of the user from the group and we can see that access denied is showing, because with the group, the permission is also deleted.

