

ACF Module 4: AWS Cloud Security

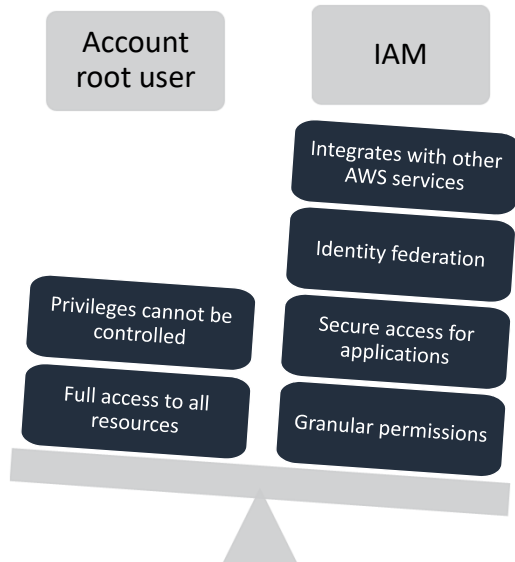
# Securing a new AWS account

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.



Introducing Section 3: Securing a new AWS account.

# AWS account root user access versus IAM access



© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

- **Best practice:** Do not use the AWS account root user except when necessary.

- Access to the **account root user** requires logging in with the *email address* (and password) that you used to create the account.

- Example actions that can only be done with the account root user:

- Update the account root user password
- Change the AWS Support plan
- Restore an IAM user's permissions
- Change account settings (for example, contact information, allowed Regions)

When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the **AWS account root user** and it is accessed by signing into the AWS Management Console with the email address and password that you used to create the account. AWS account root users have (and retain) **full** access to all resources in the account.

Therefore, AWS strongly recommends that you do not use account root user credentials for day-to-day interactions with the account.

Instead, AWS recommends that you use IAM to create additional users and assign permissions to these users, following the principle of least privilege. For example, if you require administrator-level permissions, you can create an IAM user, grant that user full access, and then use those credentials to interact with the account. Later, if you need to revoke or modify your permissions, you can delete or modify any policies that are associated with that IAM user.

Additionally, if you have multiple users that require access to the account, you can create unique credentials for each user and define which user will have access to which resources. For example, you can create IAM users with read-only access to resources in your AWS account and distribute those credentials to users that require read access. You should avoid sharing the same credentials with multiple users.

While the account root user should not be used for routine tasks, there are a few tasks that can only be accomplished by logging in as the account root user. A full list of these tasks is detailed on the [AWS Tasks that Require AWS Account Root User Credentials](#) AWS documentation page.

## Step 1: Stop using the account root user as soon as possible.

- The account root user has unrestricted access to all your resources.
- To stop using the account root user:
  1. While you are logged in as the account root user, **create an IAM user** for yourself. Save the access keys if needed.
  2. Create an IAM group, give it full administrator permissions, and add the IAM user to the group.
  3. Disable and **remove your account root user access keys**, if they exist.
  4. **Enable a password policy** for users.
  5. Sign in with your new IAM user credentials.
  6. Store your account root user credentials in a secure place.

To stop using the account root user, take the following steps:

1. While you are logged into the account root user, create an IAM user for yourself with AWS Management Console access enabled (but do not attach any permissions to the user yet). Save the IAM user access keys if needed.
2. Next, create an IAM group, give it a name (such as *FullAccess*), and attach IAM policies to the group that grant full access to at least a few of the services you will use. Next, add the IAM user to the group.
3. Disable and remove your account root user access keys, if they exist.
4. Enable a password policy for all users. Copy the **IAM users sign-in link** from the IAM Dashboard page. Then, sign out as the account root user.
5. Browse to the IAM users sign-in link that you copied, and sign in to the account by using your new IAM user credentials.
6. Store your account root user credentials in a secure place.

To view detailed instructions for how to set up your first IAM user and IAM group, see [Creating Your First IAM Admin User and Group](#).

## Step 2: Enable multi-factor authentication (MFA).

- Require MFA for your **account root user** and for **all IAM users**.
- You can also use MFA to control access to AWS service APIs.
- Options for retrieving the MFA token –
  - Virtual MFA-compliant applications:
    - Google Authenticator.
    - Authy Authenticator (Windows phone app).
  - U2F security key devices:
    - For example, YubiKey.
  - Hardware MFA options:
    - Key fob or display card offered by [Gemalto](#).



MFA token

Another recommended step for securing a new AWS account is to require multi-factor authentication (MFA) for the account root user login and for all other IAM user logins. You can also use MFA to control programmatic access. For details, see [Configuring MFA-Protected API Access](#).

You have a few options for retrieving the MFA token that is needed to log in when MFA is enabled. Options include virtual MFA-compliant applications (such as Google Authenticator and Authy Authenticator), U2F security key devices, and hardware MFA options that provide a key fob or display card.

## Step 3: Use AWS CloudTrail.

- CloudTrail tracks user activity on your account.
  - Logs all API requests to resources in all supported services your account.
  - **Basic AWS CloudTrail event history is enabled by default** and is free.
    - It contains all management event data on latest 90 days of account activity.
- To access CloudTrail –
  1. Log in to the **AWS Management Console** and choose the **CloudTrail** service.
  2. Click **Event history** to view, filter, and search the last 90 days of events.
- **To enable logs beyond 90 days and enable specified event alerting, create a trail.**
  1. From the CloudTrail Console trails page, click **Create trail**.
  2. Give it a name, apply it to all Regions, and create a new Amazon S3 bucket for log storage.
  3. Configure access restrictions on the S3 bucket (for example, only admin users should have access).

AWS CloudTrail is a service that logs all API requests to resources in your account. In this way, it enables operational auditing on your account.

AWS CloudTrail is enabled on account creation by default on all AWS accounts, and it keeps a record of the last 90 days of account management event activity. You can view and download the last 90 days of your account activity for *create*, *modify*, and *delete* operations of [services that are supported by CloudTrail](#) without needing to manually create another trail.

To enable CloudTrail log retention beyond the last 90 days and to enable alerting whenever specified events occur, create a new trail (which is described at a high level on the slide). For detailed step-by-step instructions about how to create a trail in AWS CloudTrail, see [creating a trail](#) in the AWS documentation.

### Step 4: Enable a billing report, such as the AWS Cost and Usage Report.

- Billing reports provide information about your use of AWS resources and estimated costs for that use.
- AWS delivers the reports to an Amazon S3 bucket that you specify.
  - Report is updated at least once per day.
- The **AWS Cost and Usage Report** tracks your AWS usage and provides estimated charges associated with your AWS account, either by the hour or by the day.

An additional recommended step for securing a new AWS account is to enable billing reports, such as the **AWS Cost and Usage Report**. Billing reports provide information about your use of AWS resources and estimated costs for that use. AWS delivers the reports to an Amazon S3 bucket that you specify and AWS updates the reports at least once per day.

The AWS Cost and Usage Report tracks usage in the AWS account and provides estimated charges, either by the hour or by the day.

See the AWS documentation for details about [How to Create an AWS Cost and Usage report](#).

## Module 4: AWS Cloud Security

### Optional: Securing a new AWS account – Full walkthrough

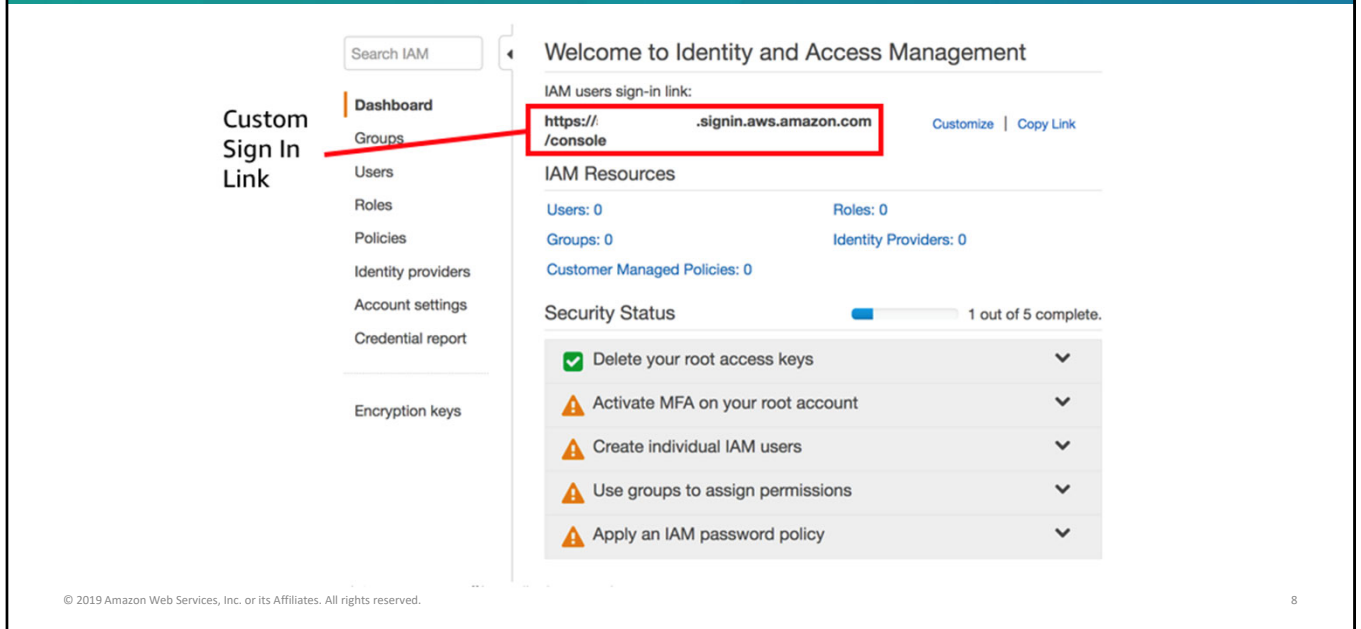
© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.



The educator might optionally choose to show a full walkthrough of the first two major steps that you must complete to secure a new AWS account. (These steps were described in the previous slides.) The slides in this section provide screen captures of what it looks like to go through the process in detail.



# IAM security status review



The screen capture shows an example of what the IAM Console Dashboard looks like when you are logged in as the AWS account root user. To access this screen in an account:

1. Log in to the **AWS Management Console** as the AWS account root user.
2. Go to the **IAM** service page and click the **Dashboard** link.
3. Review the information in the **Security Status** panel.

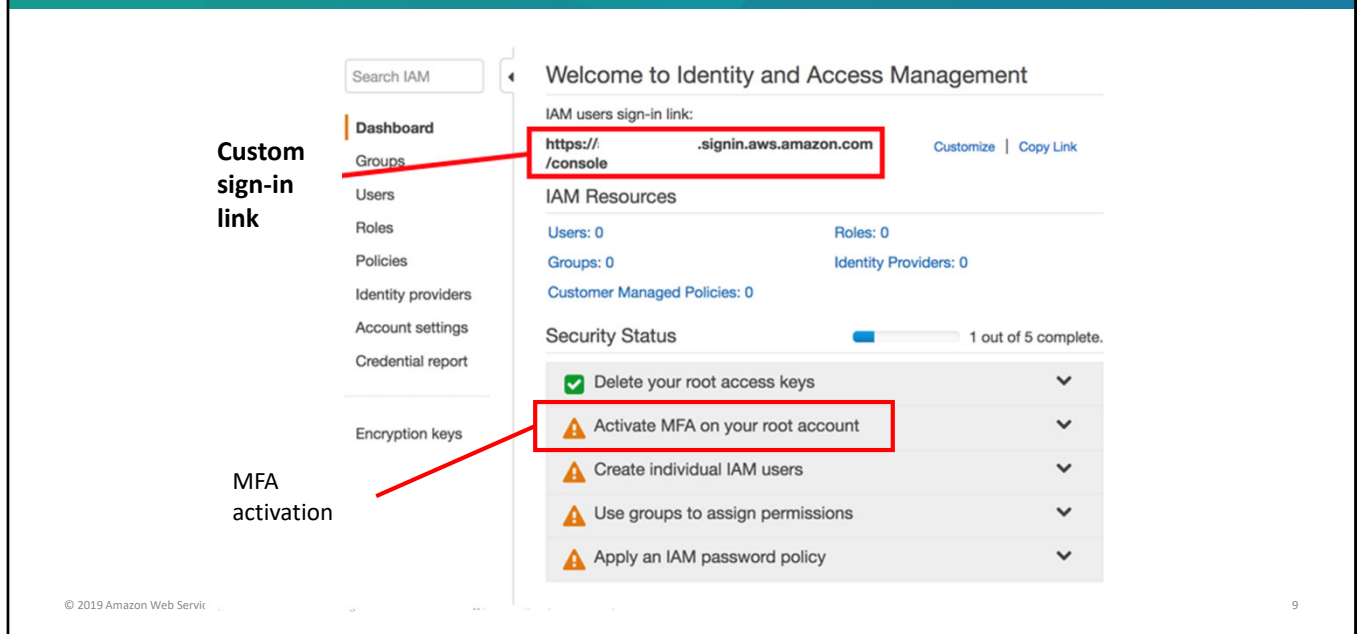
In the screen capture, only one of the five security status checks has been completed (*Delete your root access keys*). The goal of a person who completes the steps to secure the account is to receive green checks next to each security status item.

A review of the current **Security Status** list indicates that:

- MFA has *not* been activated on the AWS account root user.
- No individual IAM users have been created.
- No permissions have been assigned to groups.
- No IAM password policy has been applied.

There is a custom IAM user sign-in link for the account. Note that the account number was hidden in this screen capture. Optionally, you can use the **Customize** link to the right of the IAM user sign-in link to change the name of the account so that it does not display the account number. This link is used to sign in to the account, and it can be sent to users after their accounts are created.

# Activate MFA on the account root user

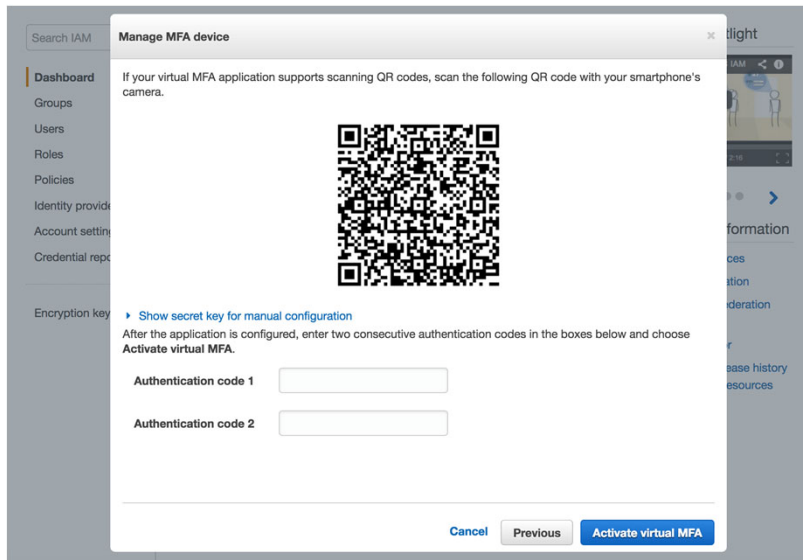


Before you create IAM users in the account, activate MFA on the account root user. To log in as the account root user, use the email address that you used to create the account. The account root user has access to everything, which is why it is important to secure this account with restrictions.

To configure MFA:

1. Click the **Activate MFA on your root account** link.
2. Click **Manage MFA**.
3. Click **Assign MFA device**. You have three options: **Virtual MFA device**, **U2F security key**, and **Other hardware MFA device**. A hardware device is an actual hardware device.
4. For purposes of this demonstration, select **Virtual MFA device** and then click **Continue**.
5. A new dialog box appears and asks you to configure a virtual MFA device. An app (such as Google Authenticator) must be downloaded for this task. After the download is complete, click **Show QR code**.

# Activate MFA on account root user



6. In the authenticator application, choose the **plus sign (+)**.
7. Scan the barcode, and enter the first authentication code.
8. Wait a moment for the second code to display, and enter the second code.
9. Click the **Assign MFA** button.

# MFA on account root user is activated

The screenshot shows the AWS IAM console interface. On the left is a navigation sidebar with a search bar and a list of menu items: Dashboard, Groups, Users, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The 'Dashboard' item is selected. Below the menu items, the text 'MFA activated' is displayed. A red arrow points from this text to the 'Activate MFA on your root account' item in the 'Security Status' panel. The main content area is titled 'Welcome to Identity and Access Management'. It includes a sign-in link for IAM users, a section for IAM Resources (Users: 0, Roles: 0, Groups: 0, Identity Providers: 0, Customer Managed Policies: 0), and a 'Security Status' section. The 'Security Status' section shows a progress bar at '2 out of 5 complete' and a list of five items: 'Delete your root access keys' (green checkmark), 'Activate MFA on your root account' (green checkmark, highlighted with a red box), 'Create individual IAM users' (yellow warning triangle), 'Use groups to assign permissions' (yellow warning triangle), and 'Apply an IAM password policy' (yellow warning triangle).

Search IAM

Dashboard

Groups

Users

Roles

Policies

Identity providers

Account settings

Credential report

Encryption keys

MFA activated

Welcome to Identity and Access Management

IAM users sign-in link:  
<https://raysinut.signin.aws.amazon.com/console> [Customize](#) | [Copy Link](#)

IAM Resources

Users: 0 Roles: 0

Groups: 0 Identity Providers: 0

Customer Managed Policies: 0

Security Status 2 out of 5 complete.

- ✓ Delete your root access keys
- ✓ Activate MFA on your root account
- ⚠ Create individual IAM users
- ⚠ Use groups to assign permissions
- ⚠ Apply an IAM password policy

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

11

10. Click **Finish** and refresh your browser.

In the **Security Status** panel, it should now show a green checkmark icon, which indicates that MFA is now activated on the account root user.

# Create an individual IAM user (1)

The screenshot shows the AWS IAM console dashboard. On the left is a navigation menu with options: Dashboard, Groups, Users, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The main content area is titled 'Welcome to Identity and Access Management'. It includes a sign-in link for IAM users, a summary of IAM Resources (0 Users, 0 Roles, 0 Groups, 0 Identity Providers, 0 Customer Managed Policies), and a 'Security Status' section. The Security Status section shows a progress bar at '2 out of 5 complete' and a list of tasks: 'Delete your root access keys' (checked), 'Activate MFA on your root account' (checked), 'Create individual IAM users' (highlighted with a red box and a red arrow pointing from the text 'IAM user creation' in the left margin), 'Use groups to assign permissions' (warning icon), and 'Apply an IAM password policy' (warning icon). The footer contains the copyright notice '© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.' and the page number '12'.

Most AWS accounts are shared by multiple users in an organization. To support this practice, you can set up each user with individually assigned permissions, or you can add users to the appropriate IAM group that grants them specific permissions.

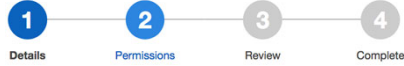
An AWS best practice is to provide each user with their own IAM user login so that they do not log in as the account root user with global privileges, or use the same credentials as someone else to log in to the account.

To configure this setup:

1. Click **Create individual IAM users** and then select **Manage Users**.

# Create an individual IAM user (2)

## Add user



### Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\*

[Add another user](#)

### Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

- Access type\* ☒ **Programmatic access**  
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.
- ☒ **AWS Management Console access**  
Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password\* ☒ Autogenerated password  
☐ Custom password

Require password reset ☒ User must create a new password at next sign-in  
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.


2. Select **Add user** and specify a new user name. Note that user names cannot have spaces.
3. Select the **Access type**. There are two access types (you can grant either type or both types to the user, but for the purposes of this demonstration, grant both types):
  - **Programmatic access** enables the user to have AWS CLI access to provision resources. This option will generate an access key one time. This access key must be saved because it will be used for all future access.
  - **AWS Management Console access** enables the user to log in to the console.
4. If you chose to grant console access, either choose **Autogenerate password**, or select **Custom password** and enter one.
5. Click **Next: Permissions**.

# Create an individual IAM user (3)


## Add user




### Set permissions for Mi



Add user to group



Copy permissions from existing user



Attach existing policies directly

**Get started with groups**

You haven't created any groups yet. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. Get started by creating a group. [Learn more](#)

Create group

Cancel Previous Next: Review

Next, you will assign permissions. You have three options for assigning permissions:

- Add user to group
- Copy permissions from an existing user
- Attach existing policies directly

6. You want to add the user to a group, so select **Add user to group** and then choose **Create group**.

Note: A group is where you put users to inherit the policies that are assigned to the group.











# Create an individual IAM user (4)

## Create group

Create a group and select the policies to be attached to the group. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. [Learn more](#)

Group name

Filter: Policy type  Showing 313 results

	Policy name	Type	Attachments	Description
<input checked="" type="checkbox"/>	 AdministratorAccess	Job function	0	Provides full access to AWS services and resources.
<input type="checkbox"/>	 AlexaForBusinessDeviceSetup	AWS managed	0	Provide device setup access to AlexaForBusiness services
<input type="checkbox"/>	 AlexaForBusinessFullAccess	AWS managed	0	Grants full access to AlexaForBusiness resources and access to relat...
<input type="checkbox"/>	 AlexaForBusinessGatewayEx...	AWS managed	0	Provide gateway execution access to AlexaForBusiness services
<input type="checkbox"/>	 AlexaForBusinessReadOnlyA...	AWS managed	0	Provide read only access to AlexaForBusiness services
<input type="checkbox"/>	 AmazonAPIGatewayAdminist...	AWS managed	0	Provides full access to create/edit/delete APIs in Amazon API Gatew...
<input type="checkbox"/>	 AmazonAPIGatewayInvokeFu...	AWS managed	0	Provides full access to invoke APIs in Amazon API Gateway.
<input type="checkbox"/>	 AmazonAPIGatewayPushToC...	AWS managed	0	Allows API Gateway to push logs to user's account.
<input type="checkbox"/>	 AmazonAppStreamFullAccess	AWS managed	0	Provides full access to Amazon AppStream via the AWS Managemen...
<input type="checkbox"/>	 AmazonAppStreamReadOnly...	AWS managed	0	Provides read only access to Amazon AppStream via the AWS Mana...

7. Give the group a name. In this example, give the lead developer administrative access and then choose **Create group**.



# Create an individual IAM user (5)

Add user

1 Details 2 Permissions 3 Review 4 Complete

Set permissions for M



Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Create group Refresh

Search		Showing 1 result
Group	Attached policies	
✓ Administrators	AdministratorAccess	

Cancel Previous Next: Review

8. Select **Next Review** to review what will be created, and then choose **Create user**.

# IAM user creation successful

## Add user



### Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://raysinut.signin.aws.amazon.com/console>

Download .csv

	User	Access key ID	Secret access key	Password	Email login instructions
▶	✓ Mi	AKI	***** Show	***** Show	Send email <a href="#">↗</a>

Close

When a user is created—and assuming you enabled both programmatic and console access when you defined the **Access type** setting and created the user—several artifacts will be generated:

1. An **access key ID** that can be used to sign AWS API calls when the user uses the AWS CLI or AWS SDKs.
2. A **secret access key** that is also used to sign AWS API calls when the user uses the AWS CLI or AWS SDKs.
3. A **password** that can be used to log in to the AWS Management Console.

Choose **Show** to display the values in each field. The credentials can also be downloaded by choosing **Download .csv**. This time is the only time when you have the option to download these credentials. You will not have an opportunity to retrieve the secret access key after this screen. Thus, you should either download the credentials, or—at the minimum—copy the secret access key, and paste it in a safe location.

**Important:** Never store these credentials in a public place (for example, never embed these credentials in code that you upload to GitHub or elsewhere). This information can be used to access your account. If you ever have a concern that your credentials have been compromised, log in as a user with IAM administrator access permissions and delete the existing access key. You can then optionally create a new access key.

# IAM Dashboard security status

Search IAM

Dashboard

Groups

Users

Roles

Policies

Identity providers

Account settings

Credential report

Encryption keys

Password policy creation

## Welcome to Identity and Access Management

IAM users sign-in link:  
<https://raysinut.signin.aws.amazon.com/console> [Customize](#) | [Copy Link](#)

### IAM Resources

Users: 1 Roles: 0  
Groups: 1 Identity Providers: 0  
Customer Managed Policies: 0

### Security Status

4 out of 5 complete.

✓	Delete your root access keys	▼
✓	Activate MFA on your root account	▼
✓	Create individual IAM users	▼
✓	Use groups to assign permissions	▼
⚠	Apply an IAM password policy	▼

When you return to the IAM Dashboard, the **Create individual IAM users** and **Use groups to assign permissions** security status items should show that they were addressed.

The remaining security item to address is to apply an IAM password policy.

# Set an IAM password policy

Search IAM

Dashboard  
Groups  
Users  
Roles  
Policies  
Identity providers  
**Account settings**  
Credential report  
Encryption keys

### Password Policy

You have unsaved changes to your password policy.

A password policy is a set of rules that define the type of password an IAM user can set. For more information about password policies, go to [Managing Passwords in Using IAM](#).

Currently, this AWS account does not have a password policy. Specify a password policy below.

Minimum password length:

- ☒ Require at least one uppercase letter ⓘ
- ☒ Require at least one lowercase letter ⓘ
- ☒ Require at least one number ⓘ
- ☒ Require at least one non-alphanumeric character ⓘ
- ☒ Allow users to change their own password ⓘ
- ☐ Enable password expiration ⓘ  
Password expiration period (in days):
- ☐ Prevent password reuse ⓘ  
Number of passwords to remember:
- ☐ Password expiration requires administrator reset ⓘ

**Apply password policy** **Delete password policy**

The IAM password policy is a set of rules that defines the type of password that an IAM user can set.

Select the rules that the passwords should comply with and then choose **Apply password policy**.

# Security status checks completed



Search IAM

Dashboard

Groups

Users

Roles

Policies

Identity providers

Account settings

Credential report

Encryption keys

## Welcome to Identity and Access Management

IAM users sign-in link:  
<https://raysinut.signin.aws.amazon.com/console> [Customize](#) | [Copy Link](#)

### IAM Resources

Users: 1      Roles: 0  
Groups: 1      Identity Providers: 0  
Customer Managed Policies: 0

### Security Status

5 out of 5 complete.

✓	Delete your root access keys	▼
✓	Activate MFA on your root account	▼
✓	Create individual IAM users	▼
✓	Use groups to assign permissions	▼
✓	Apply an IAM password policy	▼

All the security status checkmarks should now be green. Your account is now in compliance with the listed IAM security status checks. Congratulations!

In the shared responsibility model, AWS is responsible for providing security of the cloud. Encryption of data at rest and data in transit and security group configuration are examples of security in the cloud. Maintaining physical hardware is the responsibility of AWS under the shared responsibility model. When creating an IAM policy, a user can be granted AWS Management Console access and programmatic access. Managing access to AWS resources and defining fine-grained access rights are best practices when securing accounts with AWS IAM. Changing support plan can only be done by AWS root user. The other tasks are done with IAM. After initial login, AWS recommended deleting the access key of the AWS account root user as the best practice. To add an additional layer of login security to a user's AWS Management Console, enable multi-factor authentication. AWS KMS is a service that allows you to create and manage encryption keys and control the use of encryption across a wide range of AWS services in your application.

IAM user names only need to be unique within an account. By defining the permission in an IAM policy and putting the users in a group, you can set the same level of permissions to all users in that group. IAM roles do not provide permanent credentials. Roles can be assigned to multiple individuals, even if they are in different account. Resource-Based policies do not specify a user or group. They can control access to specific resources. IAM checks for explicit deny statements before it checks for explicit allow statements. Attaching policies to groups applies the same access rules to all member of the group. It also automatically applies the access rules to new users that are added to the group, and remove those access rules from users that are removed from the group. Identity federation does not address the security of application in AWS. AWS WAF helps protect your web applications against common web exploits. AWS KMS and AWSHSM are used to create and manage encryption keys. Using AWS Organizations, you can automate account creation, create groups of accounts to reflect your business needs, and apply policies for these groups for governance. The employees can use their root user access to override a policy that is attached to an IAM group.