

IAM-Identity and Access Management

Contents

IAM-Identity and Access Management	1
Functionality [AWS video https://youtu.be/yv0RByTsYf4]	1
IAM Roles.....	2
Pricing of IAM	2
Creating an IAM User (AWS Management Console)	3
How IAM Users Sign In to Your AWS Account.....	4
Creating IAM Groups	5
Creating an Administrators Group Using the Console.....	5
Understand the construct of a JSON Policy:	6
IAM Roles (Delegation and Federation).....	8
Creating a role for a service using the console	9
EC2-INSTANCE CREATION and USING THE ROLE	10
Sample: Creating a role for delegating user permissions using the AWS Management Console.....	11
Examples of Policies for Delegating Access.....	12
Example: Using a resource-based policy to delegate access to an Amazon S3 bucket in another account	12
FAQs:.....	13
Temporary Security Credentials	15
Identity Federation	16
Additional Questions	16
Troubleshooting IAM.....	16
LIMITATIONS:.....	17

Functionality [AWS video <https://youtu.be/yv0RByTsYf4>]

AWS IAM allows you to:

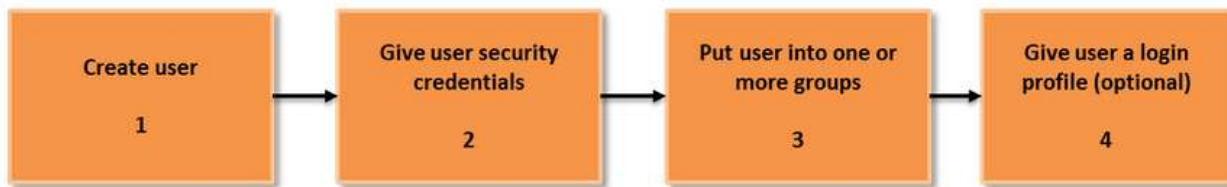
Manage IAM users and their access – You can create users in IAM, assign them individual security credentials (in other words, access keys, passwords, and **multi-factor authentication** devices), or request temporary security credentials to provide users access to AWS services and resources. You can manage permissions in order to control which operations a user can perform.

Manage IAM roles and their **permissions** – You can create roles in IAM and manage permissions to control which operations can be performed by the entity, or AWS service, that assumes the role. You can also define which entity is allowed to assume the role.

Manage federated users and their **permissions** – You can enable identity federation to allow existing identities (e.g. users) in your enterprise to access the AWS Management Console, to call AWS APIs, and to access resources, without the need to create an IAM user for each identity.

Protect your AWS environment by using AWS Multi-Factor Authentication (MFA), a security feature available at no extra cost that augments user name and password credentials. MFA requires users to prove physical possession of a hardware or virtual MFA device by providing a valid MFA code.

User Creation:



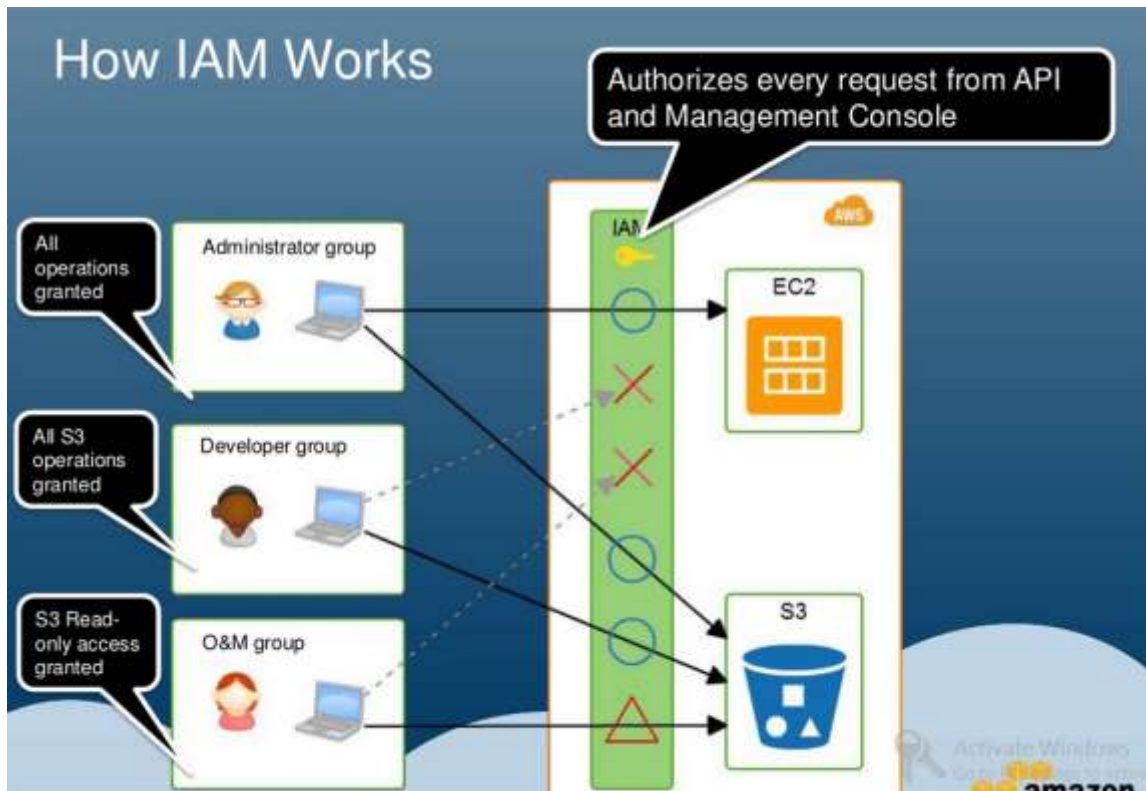
A group is a collection of IAM users. Groups let you assign permissions to a collection of users, which can make it easier to manage the permissions for those users. For example, you could have a group called Admins and give that group the types of permissions that administrators typically need. Any user in that group automatically has the permissions that are assigned to the group. If a new user joins your organization and should have administrator privileges, you can assign the appropriate permissions by adding the user to that group. Similarly, if a person changes jobs in your organization, instead of editing that user's permissions, you can remove him or her from the old group and add him or her to the new group.

IAM Roles

IAM roles allow you to delegate access to users or services that normally don't have access to your organization's AWS resources. IAM users or AWS services can assume a role to obtain temporary security credentials that can be used to make AWS API calls. Consequently, you don't have to share long-term credentials or define permissions for each entity that requires access to a resource.

Pricing of IAM

AWS Identity and Access Management is a feature of your AWS account offered at no additional charge.



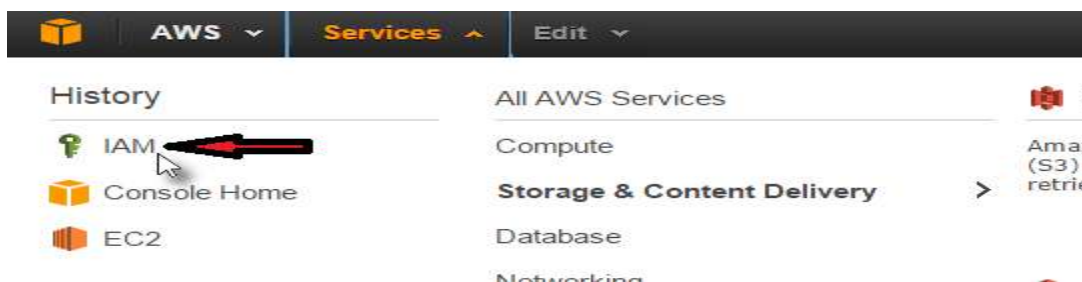
Operations and Configuration

- Two Ways for Managing Users and Groups
 - AWS Management Console
 - IAM API
- "Access Policy Language" for describing policies
 - JSON format

Creating an IAM User (AWS Management Console)

To create a user using the AWS Management Console

1. Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.



- In the navigation pane, click **Users** and then click **Create New Users**.
- Enter the user names for the users you want to create. You can create up to five users at one time.

Note

User names can use only alphanumeric characters plus these characters: plus (+), equal (=), comma (,), period (.), at (@), and hyphen (-).

For more information about limitations on IAM entities , refer the section [LIMITATIONS](#)

- If you want to generate an access key ID and secret access key for new users, select **Generate an access key for each user**. Users must have keys if they need to work with the AWS CLI or with the AWS SDKs or APIs. Click **Create**.

Note

If you have users who will work with the AWS Management Console, you must create passwords for each of them. Creating passwords is described in a later step.

- A page appears that enables you to download the access key IDs for the new user or users. To save the access keys for the new user or users, click **Download Credentials**. This lets you save the access key IDs and secret access keys to a CSV file on your computer.

Important

This is your only opportunity to view or download the keys, and you must provide this information to your users before they can begin using an AWS API. If you don't download and save them now, you will need to create new access keys for the users later. Save the user's new access key ID and secret access key in a safe and secure place. **You will not have access to the secret access keys again after this step.**

- (Optional) Give the user permission to manage his or her own security credentials.
- (Optional) Create a password if the user needs to access the console. If the user will only access AWS by using the CLI or the API, then you only need the access keys from step 5
- (Optional) Attach a policy to the user (or to a group the user is a member of) to grant the user permissions to access AWS resources .
- Provide the information that the user needs to sign in. This includes the user name and password or access keys, and the URL to the sign-in page for the account, substituting the correct account ID number or account alias for *AWS-account-ID*:

```
https://AWS-account-ID.signin.aws.amazon.com/console
```

How IAM Users Sign In to Your AWS Account

After you have created IAM users and created passwords for them, users can sign in to the AWS Management Console for your AWS account by using a special URL, which has this format:

```
https://AWS-account-ID.signin.aws.amazon.com/console
```

For example, the URL might look like this:

```
https://123456789012.signin.aws.amazon.com/console
```

By default, the sign-in URL for your account includes your account ID. You can create a unique sign-in URL for your account so that the URL includes a name instead of an account ID.

You can also find the sign-in URL for an account on the IAM console dashboard.

IAM users sign-in link:

<https://my-account.signin.aws.amazon.com/console>

[Customize](#) | [Copy Link](#)

Creating IAM Groups

Creating an Administrators Group Using the Console

This procedure describes how to create an IAM group named Administrators, grant the group full permissions for all AWS services, create an IAM user for yourself, and add the user to the Administrators group.

To create the Administrators group

1. Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, click Groups, then click Create New Group.



3. In the Group Name box, type **Administrators** and then click Next Step.
4. In the list of policies, select the check box next to the AdministratorAccess policy. You can use the Filter menu and the Search box to filter the list of policies.
5. Click Next Step, then click Create Group.
6. Your new group is listed under Group Name.

To create an IAM user for yourself, add the user to the Administrators group, and create a password for the user

1. In the navigation pane, click Users and then click Create New Users.
2. In box 1, enter a user name. Clear the check box next to Generate an access key for each user, then click Create.
3. In the list of users, click the name (not the check box) of the user you just created. You can use the Search box to search for the user name.
7. In the Groups section, click Add User to Groups.
8. Select the check box next to the Administrators group, then click Add to Groups.
9. Scroll down to the Security Credentials section. Under Sign-In Credentials, click Manage Password.

10. Select Assign a custom password, then enter a password in the Password and Confirm Password boxes. When you are finished, click Apply.

Attaching a Policy to an IAM Group (AWS Management Console)

The following procedure describes how to attach a managed policy to a group. To attach an AWS managed policy—that is, a pre-written policy provided by AWS—to a group, follow the steps in the following procedure now. To attach a customer managed policy—that is, a policy with custom permissions that you create—you must first create the policy

To attach a policy to a group

1. Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, select **Policies**.
3. In the list of policies, select the check box next to the name of the policy to attach. You can use the **Filter** menu and the **Search** box to filter the list of policies.
4. Click **Policy Actions**, then click **Attach**.
5. Click **All Types** in the **Filter** menu, then click **Groups**.
6. Select the check box next to the name of the group to attach the policy to, then click **Attach Policy**.

Understand the construct of a JSON Policy:



Access Policy Language

```

{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBuckets",
        "s3:Get *"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceIP": ["176.32.92.49/32"]
        }
      }
    }
  ]
}

```

Access is granted or rejected according to the statement

Access Policy Configuration

```

{
  "Effect": "Allow",
  "Action": [
    "s3:ListBuckets",
    "s3:Get *"
  ],
  "Resource": [
    "*"
  ],
  "Condition": {
    "StringEquals": {
      "aws:SourceIP": ["176.32.92.49/32"]
    }
  }
}

```

- "Allow" for granting access
"Deny" for rejecting
- Specifies target operations
* Wildcard is allowed
- Specifies target resources with Amazon Resource Name (ARN)
* Wildcard is allowed
- Specifies condition to enable this policy

This example means
"If the request is from 176.32.92.49, S3 ListBuckets and Get related operations would be allowed"

Action & Resource

- ❗ "Action" specifies right for **operations**, e.g.
 - RunInstances
 - AttachVolume
 - CreateBucket
 - DeleteObject
- ❗ "Resource" specifies right for **targets** of operations, e.g.
 - EC2 Instances
 - EBS Volumes
 - S3 Buckets
 - S3 Objects

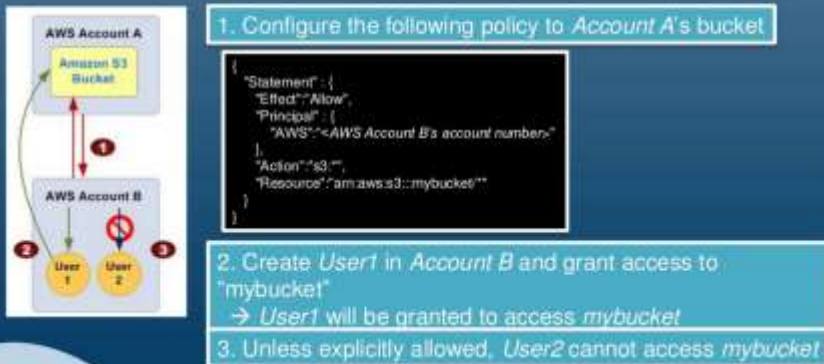
User based and Resource based

- Besides Users and Groups, Policies can be Assigned to Resources
- E.g. S3 Buckets and SQS queues can be applied policies
 - Configuring a bucket to be only accessible from a certain IP address(es)



Cross-Account Access

- Granting Access from an AWS account to Another

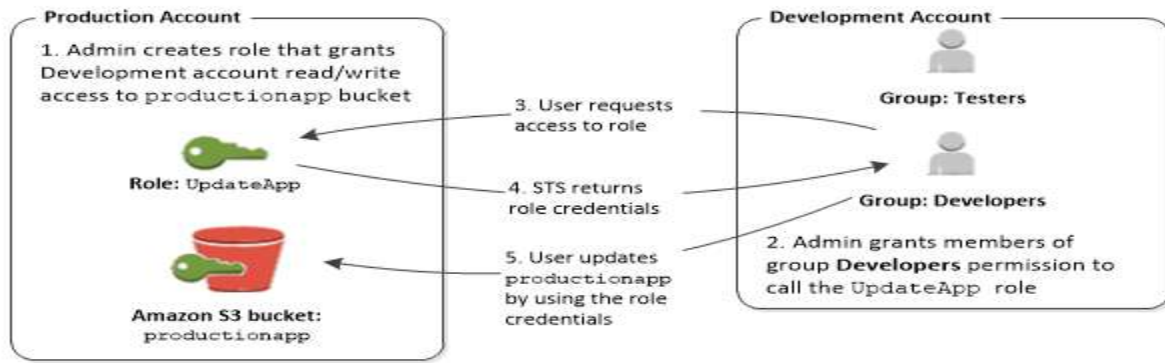


Also refer : <https://www.youtube.com/watch?v=ZhvXW-ILyPs>

IAM Roles (Delegation and Federation)

Sometimes you want to delegate access to users, applications, or services that don't normally have access to your AWS resources. For example, you might want to grant a user in your AWS account access to resources they don't usually have, or grant a user in one AWS account access to resources in another account. Or you might want to allow a mobile app to use AWS resources, but not want to store AWS keys within the app (where they can be difficult to rotate and where users can potentially extract them). Sometimes you want to give users who already have identities outside of AWS, such as through your corporate directory, access to AWS resources, that is, create federated identities. Or, you might want to grant access to your account to a third party, for example, so that they can perform an audit on your resources.

Use a role to delegate permissions to a user in a different account



Creating a Role to Delegate Permissions to an AWS Service

Creating a role for a service using the console

To create a role for an AWS service using the AWS Console

1. In the navigation pane of the console, click **Roles**, and then click **Create New Role**.
2. In the **Role name** box, enter a role name that can help you identify the purpose of this role. Role names must be unique within your AWS account. After you enter the name, click **Next Step** at the bottom of the page.

Because various entities might reference the role, you cannot edit the name of the role after it has been created.

3. Expand the **AWS Service Roles** section, and then select the service that you will allow to assume this role.
4. Select the managed policy that enables the permissions you want the service to have.
5. Click **Next Step** to review the role and then click **Create Role**.

Create Role

Step 1: Set Role Name

Step 2: Select Role Type

Step 3: Establish Trust

Step 4: Attach Policy

Set Role Name

Enter a role name. You cannot edit the role name after the role is created.

Role Name

ec2-s3

Maximum 64 characters. Use alphanumeric and '+', '@', '_' characters.

Select Role Type

AWS Service Roles

Amazon EC2


Allows EC2 instances to call AWS services on your behalf.

Select

Attach Policy

Select up to two policies to attach to the role.

Filter: Policy Type Showing 124 results

	Policy Name	Attached Entities	Creation Time	Edited Time
	AmazonS3FullAccess	4	2015-02-07 00:10 UTC+0530	2015-02-07 00:10 UTC+0530

Review

Review the following role information. To edit the role, click an edit link, or click **Create Role** to finish.

[Cancel](#)
[Previous](#)
[Create Role](#)

EC2-INSTANCE CREATION and USING THE ROLE

Step1 : Select **Amazon Linux AMI 2015.03 (HVM), SSD Volume Type - ami-68d8e93a**

Step 2: Choose an Instance Type -> t2.micro

Step 3: Instance details section: Select the role.

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instance, request an Elastic IP, assign an IAM management role to the instance, and more.


Number of instances ⓘ

Purchasing option ⓘ ☐ Request Spot Instances

Network ⓘ [Create new VPC](#)

Subnet ⓘ [Create new Subnet](#)

Auto-assign Public IP ⓘ

 IAM role ⓘ [Create new IAM role](#)

Complete Instance creation with chosen security and keypair as usual

SSH to the instance using public IP

Test the AWS s3 access by typing the following in the '\$' prompt

```
aws s3 ls
```

(you will be listed with the s3 buckets)

```
[ec2-user@ip-172-31-45-215 ~]$ aws s3 ls
2015-04-04 04:15:15 cloud-b-lab.com
2015-04-07 12:41:29 jaikrishjai
2015-04-16 04:09:05 jmsapr16
2015-04-17 04:25:57 jpalogbkt2015
2015-04-04 04:09:59 rajivdayal
2015-04-22 04:05:16 testjpa
```

Sample: Creating a role for delegating user permissions using the AWS Management Console

To create a role that an IAM user can switch to using the AWS Management Console

1. In the navigation pane of the console, click **Roles** and then click **Create New Role**.
2. In the **Role name** box, enter a role name that can help you identify the purpose of this role. Role names must be unique within your AWS account. After you enter the name, click **Next Step** at the bottom of the page.

Role names have character limitations. The number of roles in an AWS account, and the policy size for policies attached to roles are also limited. Because various entities might reference the role, you cannot edit the name of the role after it has been created.

3. Click **Roles for Cross-Account Access**, and then select the type of role that you want to create:
 - Select **Provide access between AWS accounts you own** if you are the administrator of both the user account and the resource account, or both accounts belong to the same company. This is also the option to select when the users, role, and resource to be accessed are all in the same account.
 - Select **Allows IAM users from a 3rd party AWS account to access this account** if you are the administrator of the account that owns the resource and you want to grant permissions to users from an account that you do not control. This option requires you to specify an external ID (which must be provided to you by the third party) to provide additional control over the circumstances in which the third party can use the role to access your resources.

Important

Selecting this option enables access to the role only through the AWS CLI and API. You cannot switch roles at the AWS console to a role that has an external ID condition in its trust policy. However, you can create this kind of access programmatically by writing a script or an application using the relevant SDK.

4. Specify the AWS account ID that you want to grant access to your resources.

Any IAM user from the trusted AWS account can assume this role if that user has a policy that grants permission for the AWS STS `AssumeRole` action and that specifies your role as the resource.

5. If you selected **Allows IAM users from a 3rd party AWS account to access this account**, enter the external ID provided by the administrator of the third party account. This automatically adds a condition to the trust policy that allows the user to assume the role only if the request includes the correct `sts:ExternalID`.
6. If you want to restrict the role to users who provide multi-factor authentication, select the **Require MFA** option. This adds an MFA condition to the role's trust policy. A user who wants to assume the role must provide a temporary one-time password (TOTP) from the configured device. Users without MFA authentication cannot assume the role.
7. Click **Next Step**.
8. Set the permissions for the role to specify what actions can be done on specific resources (similar to setting permissions for IAM groups). You specify permissions by selecting a policy.

The permissions that you specify are available to any entity that uses the role. By default, roles have no permissions.

Select the box next to the policy that assigns the permissions that you want the users to have, and then click **Attach Policy**.

9. Click **Next Step** to review the role. Note the link provided for you to give to users who can use the role. When the user clicks this link, the user is taken directly to the Switch Role page with the **Account ID** and **Role Name** already filled in. The user can optionally set a **Display Name** and can select a **Display Color**. When the user clicks **Switch Role**, the user immediately begins operating with the new permissions.

10. Click **Create Role** to complete the creation of the role.

Important

Remember that this is only the first half of the configuration required. You must also enable individual users in the trusted account with permissions to switch to the role.

Examples of Policies for Delegating Access

Example: Using a resource-based policy to delegate access to an Amazon S3 bucket in another account

In this example, Account A uses a resource-based policy (an Amazon S3 [bucket policy](#)) to grant Account B full access to Account A's Amazon S3 bucket. Then Account B creates an IAM user policy to delegate that access to Account A's bucket to one of the users in Account B.

The Amazon S3 bucket policy in Account A might look like the following policy. In this example, Account A's Amazon S3 bucket is named *mybucket*, and Account B's account number is 111122223333.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "AccountBAccess1",
    "Effect": "Allow",
    "Principal": {"AWS": "111122223333"},
    "Action": "s3:*",
    "Resource": [
      "arn:aws:s3:::mybucket",
      "arn:aws:s3:::mybucket/*"
    ]
  }
}
```

Alternatively, Account A can use Amazon S3 [Access Control Lists \(ACLs\)](#) to grant Account B access to an Amazon S3 bucket or a single object within a bucket. In that case, the only thing that changes is how Account A grants access to Account B. Account B still uses a policy to delegate access to a user in Account B, as described in the second part of this example. For more information about controlling access on Amazon S3 buckets and objects, go to [Access Control](#) in the *Amazon Simple Storage Service Developer Guide*.

The next example shows the IAM user (or group) policy that Account B might create to delegate read access to a user in Account B. In this policy, the `Action` element is explicitly defined to allow only `List` actions, and the `Resource` element of this policy matches the `Resource` for the bucket policy implemented by Account A.

Account B implements this policy by using IAM to attach it to the appropriate user (or group) in Account B.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "s3:List*",
    "Resource": [
      "arn:aws:s3:::mybucket",
      "arn:aws:s3:::mybucket/*"
    ]
  }
}
```

FAQs:

Q: What kinds of security credentials can IAM users have?

IAM users can have any combination of credentials that AWS supports, such as an AWS access key, X.509 certificate, password for web app logins, or an MFA device. This allows users to interact with AWS in any manner that makes sense for them. An employee might have both an AWS access key and a password; a software system might have only an AWS access key to make programmatic calls; and an outside contractor might have only an X.509 certificate to use the EC2 command-line interface.

Q: Can user access be enabled and disabled?

Yes. An IAM user's access keys can be enabled and disabled via the IAM APIs, AWS CLI, or IAM console. Disabling the access keys means the user will not be able to programmatically access the AWS services.

Q: Can a collection of users be structured in a hierarchical way, such as in LDAP?

Yes. Users and groups can be organized under paths, similar to object paths in Amazon S3—for example `/mycompany/division/project/joe`.

Q: Can users be defined regionally?

Not initially. Users are global entities, like an AWS account is today. No region is required to be specified when defining user permissions. Users are able to use AWS services in any geographic region.

Q: How are MFA devices configured for IAM users?

The AWS account holder can order multiple MFA devices. These devices can then be assigned to individual IAM users via the IAM APIs, AWS CLI, or IAM console.

Q: What kind of key rotation is supported for IAM users?

User access keys and X.509 certificates can be rotated just as they are for an AWS account's root access identifiers. A user's access keys and X.509 certificates can be managed and rotated programmatically via the IAM APIs, AWS CLI, or IAM console.

Q: Can IAM users have individual EC2 SSH keys?

Not in the initial release. IAM does not affect EC2 SSH keys or Windows RDP certificates. This means that although each user has separate credentials for accessing web service APIs, they must share SSH keys that are common across the AWS account under which the user has been defined.

Q: Do IAM user names have to be email addresses?

No, but they can be. User names are just ASCII strings that are unique within a given AWS account. The AWS account holder can assign names using any naming convention they choose, including email addresses.

Q: What character sets can I use for IAM user names?

IAM entities support only ASCII characters.

Q: Are user attributes other than user name supported?

Not at this time.

Q: How are user passwords set?

An initial password can be set for an IAM user via the IAM console, [AWS CLI](#), or IAM APIs. User passwords never appear in clear text after the initial provisioning, and are never displayed or returned via an API call. IAM users can manage their passwords via the My Password page in the IAM console. Users access this page by selecting the Security Credentials option in the AWS Management Console dropdown list in the upper right-hand corner.

Q: Can I define a password policy for my user's passwords?

Yes, you can enforce strong passwords, like requiring minimum length or at least one number. You can also enforce automatic password expiration, prevent re-use of old passwords, and require a password reset upon next AWS sign in

Q: Can I set usage quotas on IAM users?

No. All limits are on the AWS account as a whole. For example, if your AWS Account has a limit of 20 Amazon EC2 instances, IAM users with EC2 permissions can start instances up to the limit. You cannot limit what an individual user can do.

Q: Who can use IAM roles?

Any AWS customer can use this feature.

Q: How much do IAM roles cost?

IAM roles are free of charge. You will continue to pay for any resources a role in your AWS account consumes.

Q: What is the difference between an IAM role and an IAM user?

An IAM user has permanent long-term credentials and is used to directly interact with AWS services. An IAM role does not have any credentials and cannot make direct requests to AWS services. IAM roles are meant to be assumed by authorized entities, such as IAM users, applications, or an AWS service like EC2.

Q: What is the difference between an IAM role and an IAM group?

An IAM group is a collection of IAM users that share the same permissions. An IAM group is primarily a management convenience to manage the same set of permissions for a set of IAM users. An IAM role is an IAM entity with permissions to make AWS service requests. IAM roles cannot make direct requests to AWS services; they are meant to be assumed by authorized entities, such as IAM users, applications, or AWS services like EC2.

Q: When should I use an IAM user, IAM group or IAM role?

An IAM user has permanent long-term credentials and is used to directly interact with AWS services. An IAM group is primarily a management convenience to manage the same set of permissions for a set of IAM users. An IAM role is an AWS Identity and Access Management (IAM) entity with permissions to make AWS service requests. IAM roles cannot make direct requests to AWS services, they are meant to be "assumed" by authorized entities, such as IAM users, applications or AWS services like EC2. IAM roles are used to delegate access within or between AWS accounts.

Q: Can an IAM role be added to an IAM group?

Not at this time.

Q: How many policies can be attached to an IAM role?

You can add as many policies as needed to an IAM role, as long as the total size of all the policies doesn't exceed 10 KB.

Q: How many IAM roles can I create?

You are limited to 250 IAM roles under your AWS account. If you need more roles, submit the IAM limit increase request form with your use case and your IAM role increase will be considered.

Q: What are the features of IAM roles for EC2 instances?

IAM roles for EC2 instances provides the following features:

- AWS temporary security credentials to use when making requests from running EC2 instances to AWS services.
- Automatic rotation of the AWS temporary security credentials.
- Granular AWS service permissions for applications running on EC2 instances.

Q: Can I use the same IAM role on multiple EC2 instances?

Yes.

Q: Can I change the IAM role on a running EC2 instance?

No, at this time you cannot change the IAM role on a running EC2 instance. You can change the permissions on the IAM role associated with a running instance, and the updated permissions will take effect almost immediately.

Q: Can I associate an IAM role with an already running EC2 instance?

No. You can associate only one IAM role with an EC2 instance.

Q: Can I use an IAM role with other services that launch EC2 instances?

Yes. Auto Scaling and AWS CloudFormation also support IAM roles. Other services will add support over time.

Q: Can I associate an IAM role with an Auto Scaling group?

Yes. You can add an IAM role as an additional parameter in an Auto Scaling launch configuration and create an Auto Scaling group with that launch configuration. All EC2 instances launched in an Auto Scaling group that is associated with an IAM role will be launched with the role as an input parameter.

Q: Can I associate more than one IAM role with an EC2 instance?

No. You can only associate one IAM role with an EC2 instance at this time.

Q: What happens if I delete an IAM role that is associated with a running EC2 instance?

Any application running on that instance that's using the role will be denied access immediately.

Q: Can I control which IAM roles an IAM user can associate with an EC2 instance?

Yes.

Q: Which permissions are required to launch EC2 instances with an IAM role?

An IAM user must be granted two distinct permissions to successfully launch EC2 instances with roles:

- Permission to launch EC2 instances.
- Permission to associate an IAM role with EC2 instances.

Q: Who can access the access keys on the EC2 instance?

Any local user on the instance can access the access keys associated with the IAM role.

Q: How do I rotate the temporary security credentials on the EC2 instance?

The AWS temporary security credentials associated with an IAM role are automatically rotated multiple times a day. New temporary security credentials are made available no later than five minutes before the existing temporary security credentials expire.

Q: Can I use IAM roles for EC2 instances with any instance type or AMI?

Yes. IAM roles for EC2 instances also work in Amazon Virtual Private Cloud (VPC), with spot and reserved instances.

Temporary Security Credentials

Q: What are temporary security credentials?

Temporary security credentials consist of the AWS access key ID, secret access key, and security token. Temporary security credentials are valid for a specified duration and for a specific set of permissions. Temporary security credentials are sometimes simply referred to as *tokens*. Tokens can be requested for IAM users or for federated users you manage in your own corporate directory.

Identity Federation

Q: What are federated users?

Federated users are users that are managed outside of AWS in your corporate directory, but are granted access to your AWS account using temporary security credentials. They differ from IAM users, which are created and maintained in your AWS account.

Q: Do you support SAML?

Yes, AWS supports the Security Assertion Markup Language (SAML) 2.0.

Q: What SAML profiles does AWS support?

The AWS single sign-on (SSO) endpoint supports the identity provider initiated HTTP-POST binding WebSSO SAML Profile. This enables a federated user to log into the AWS Management Console using a SAML assertion. A SAML assertion can also be used to request temporary security credentials using the AssumeRoleWithSAML API.

Q: Can federated users access AWS APIs?

Yes. You can programmatically request temporary security credentials for your federated users to provide them secure and direct access to AWS APIs.

Refer the [sample application](#) that demonstrates how you can enable identity federation, providing users maintained by Microsoft Active Directory access to AWS service APIs.

Additional Questions

Q: What happens if a user tries to access a service that has not yet been integrated with IAM?

The service will return an “access denied” error.

Q: Are AWS Identity and Access Management actions logged for auditing purposes?

Yes. You can log IAM actions, STS actions, and AWS Management Console sign-ins by activating AWS CloudTrail.

Q: Is there any distinction between people and software agents as AWS entities?

No, both of these entities are treated like users with security credentials and permissions. However, people are the only ones to use a password in the AWS Management Console.

Q: Do users work with AWS Support Center and Trusted Advisor?

Yes, IAM users have the ability to create and modify support cases as well as use Trusted Advisor.

Q: Are there any default quota limits associated with IAM?

Yes, by default your AWS account has initial quotas set for all IAM-related entities.

These quotas are subject to change. If you require an increase, you can use the [Service Limit Increase form on the Contact Us page](#) and select “AWS IAM groups and users”

Troubleshooting IAM

Refer: <http://docs.aws.amazon.com/IAM/latest/UserGuide/iam-troubleshooting.html>

LIMITATIONS:

The following are the default maximums for your entities:

- Groups per AWS account: 100
- Users per AWS account: 5000

If you need to add a large number of users, consider using temporary security credentials.

- Roles per AWS account: 250
- Instance profiles per AWS account: 100
- Roles per instance profiles: 1 (each instance profile can contain only 1 role)
- Number of groups per user: 10 (that is, the user can be in this many groups)
- Access keys per user: 2
- Signing certificates per user: 2
- MFA devices in use per user: 1
- MFA devices in use per AWS account (at the root account level): 1
- Virtual MFA devices (assigned or unassigned) per AWS account: equal to the user quota for the account
- Server certificates per AWS account: 20
- AWS account aliases per AWS account: 1
- Login profiles per user: 1
- SAML providers per account: 100
- Identity providers (IdPs) per SAML provider: 10
- Keys per SAML provider: 10
- Customer managed policies per AWS account: 1000
- Versions per managed policy: 5
- Managed policies attached per IAM user, group, or role: 2

You can request to increase some of these quotas for your AWS account on the [IAM Limit Increase Contact Us Form](#). Currently you can request to increase the limit on users per AWS account, groups per AWS account, roles per AWS account, instance profiles per AWS account, and server certificates per AWS account.

The following are the maximum lengths for entities:

- Path: 512 characters
- User name: 64 characters
- Group name: 128 characters
- Role name: 64 characters

- Instance profile name: 128 characters
- Unique ID (applicable to users, groups, roles, managed policies, and server certificates): 32 characters
- Policy name: 128 characters
- Certificate ID: 128 characters
- Login profile password: 1 to 128 characters
- AWS account ID alias: 3 to 63 characters.
- Role trust policy (the policy that determines who is allowed to assume the role): 2,048 characters
- For [inline policies](#): You can add as many inline policies as you want to a user, role, or group, but the total aggregate policy size (the sum size of all inline policies) per entity cannot exceed the following limits:
 - User policy size cannot exceed 2,048 characters
 - Role policy size cannot exceed 10,240 characters
 - Group policy size cannot exceed 5,120 characters

Note

IAM does not count whitespace when calculating the size of a policy against these limitations.

- For [managed policies](#): You can add up to two managed policies to a user, role, or group. The size of each managed policy cannot exceed 5,120 characters.

Note

IAM does not count whitespace when calculating the size of a policy against this limitation.