HI-TECH INSTITUTION

CORPORATE CAREER ENHANCEMENT TRAININGS

OUR ROOT LEVEL TRAINING WILL GIVE YOU BETTER GROWTH

Hi-Tech Institution

ABOUT US

Our Vision:

To provide better training by full filing the requirements of our trainee.

Our Mission:

We always ensure to give practical based training. And we make the candidates to get good hands-on experience on any platform.

Philosophy:

Our Root Level Training Will give you Better Growth.

We successfully survived around 5 years in the IT field. Started this is as small Training room. But now we are having 5 branches across India.

Certified Trainers taking the session on various domain with any level of doubts clarification.

For More Details: **www.hitechins.in**

Write feedback to **operations@hitechins.in**

Identity and Access Management

What Is IAM?

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

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 is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

IAM Features

IAM gives you the following 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)

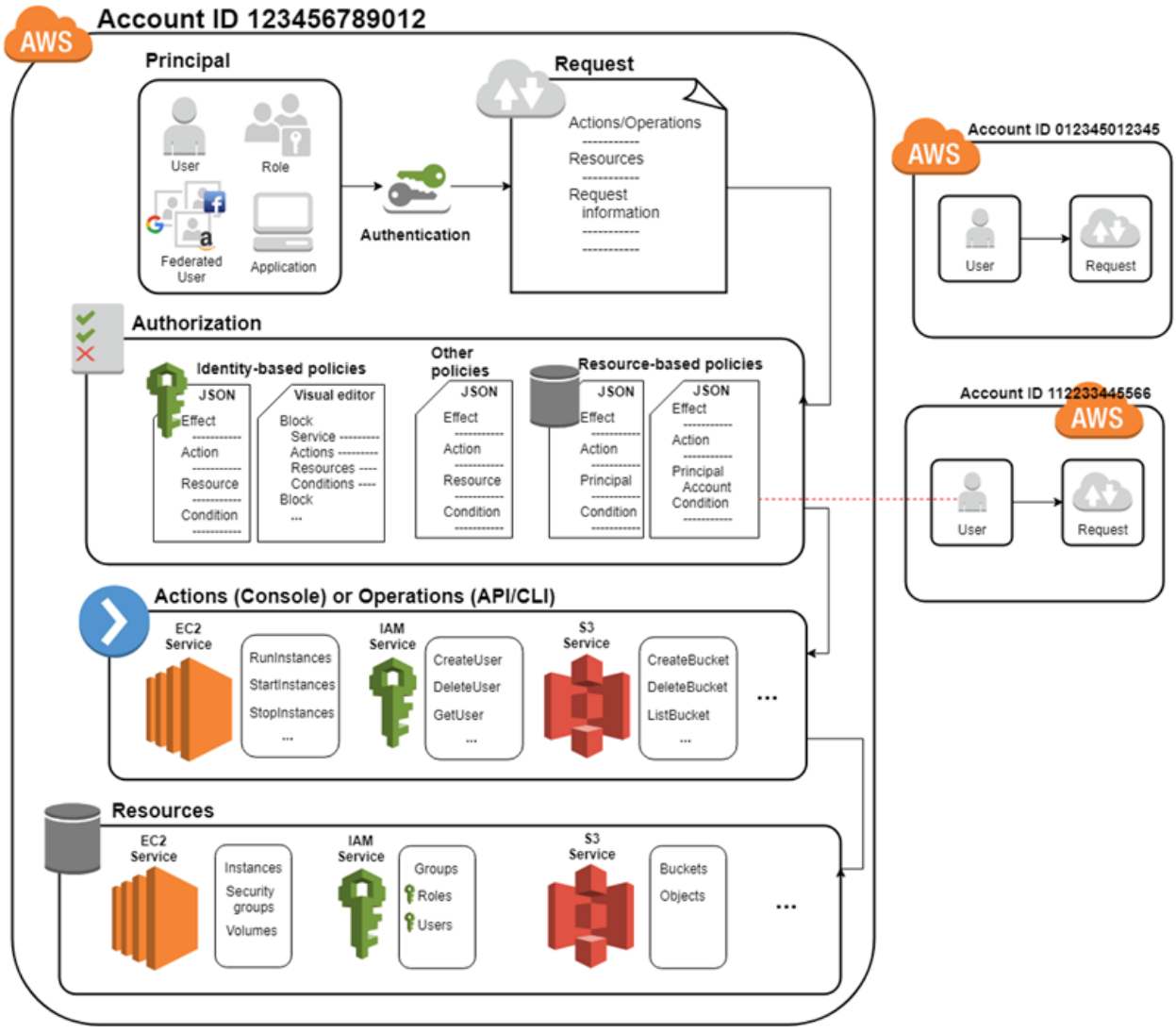
> You can add two-factor authentication to your account and to individual users for extra security. With MFA you or your users must provide not only a password or access key to work with your account, but also a code from a specially configured device.

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.

Understanding How IAM Works

Before you create users, you should understand how IAM works. IAM provides the infrastructure necessary to control authentication and authorization for your account. The IAM infrastructure includes the following elements:



Terms

Resources

The user, role, group, and policy objects that are stored in IAM. As with other AWS services, you can add, edit, and remove resources from IAM.

Identities

The IAM resource objects that are used to identify and group. These include users, groups, and roles.

Entities

The IAM resource objects that AWS uses for authentication. These include users and roles. Roles can be assumed by IAM users in your or another account as well as users federated through a web identity or SAML.

Principals

A person or application that uses an entity to sign in and make requests to AWS.

Principal

A *principal* is a person or application that can make a request for an action or operation on an AWS resource. As a principal, you first sign in as the AWS account root user. As a best practice, do not use your root user for your daily work. Instead, create IAM entities (users and roles). You can also support federated users or programmatic access to allow an application to access your AWS account.

Request

When a principal tries to use the AWS Management Console, the AWS API, or the AWS CLI, that principal sends a *request* to AWS. The request includes the following information:

- **Actions or operations** – The actions or operations that the principal wants to perform. This can be an action in the AWS Management Console, or an operation in the AWS CLI or AWS API.
- **Resources** – The AWS resource object upon which the actions or operations are performed.
- **Principal** – The person or application that used an entity (user or role) to send the request. Information about the principal includes the policies that are associated with the entity that the principal used to sign in.
- **Environment data** – Information about the IP address, user agent, SSL enabled status, or the time of day.

- **Resource data** – Data related to the resource that is being requested. This can include information such as a DynamoDB table name or a tag on an Amazon EC2 instance.

AWS gathers the request information into a *request context*, which is used to evaluate and authorize the request.

Authentication

As a principal, you must be authenticated (signed in to AWS) using an IAM entity to send a request to AWS. Although some services, such as Amazon S3 and AWS STS, allow a few requests from anonymous users, they are the exception to the rule.

To authenticate from the console as a user, you must sign in with your user name and password. To authenticate from the API or AWS CLI, you must provide your access key and secret key. You might also be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account.

Authorization

You must also be authorized (allowed) to complete your request. During authorization, AWS uses values from the request context to check for policies that apply to the request. It then uses the policies to determine whether to allow or deny the request. Most policies are stored in AWS as JSON documents and specify the permissions for principal entities. There are several types of policies that can affect whether a request is authorized. To provide your users with permissions to access the AWS resources in their own account, you need only identity-based policies. Resource-based policies are popular for granting cross-account access. The other policy types are advanced features and should be used carefully.

AWS checks each policy that applies to the context of your request. If a single permissions policy includes a denied action, AWS denies the entire request and stops evaluating. This is called an *explicit deny*. Because requests are *denied by default*, AWS authorizes your request only if every part of your request is allowed by the applicable permissions policies. The evaluation logic for a request within a single account follows these general rules:

- By default, all requests are denied. (In general, requests made using the AWS account root user credentials for resources in the account are always allowed.)
- An explicit allow in any permissions policy (identity-based or resource-based) overrides this default.
- The existence of an Organizations SCP, IAM permissions boundary, or a session policy overrides the allow. If one or more of these policy types exists, they must all allow the request. Otherwise, it is implicitly denied.
- An explicit deny in any policy overrides any allows.

If you need to make a request in a different account, a policy in the other account must allow you to access the resource *and* the IAM entity that you use to make the request must have an identity-based policy that allows the request.

Actions or Operations

After your request has been authenticated and authorized, AWS approves the actions or operations in your request. Operations are defined by a service, and include things that you can do to a resource, such as viewing, creating, editing, and deleting that resource. For example, IAM supports approximately 40 actions for a user resource, including the following actions:

- `CreateUser`
- `DeleteUser`
- `GetUser`
- `UpdateUser`

To allow a principal to perform an operation, you must include the necessary actions in a policy that applies to the principal or the affected resource.

Creating an Administrator IAM User and Group (Console)

This procedure describes how to use the AWS Management Console to create an IAM user for yourself and add that user to a group that has administrative permissions from an attached managed policy.

To create an administrator user for yourself and add the user to an administrators group (console)

1. Use your AWS account email address and password to sign in as the *AWS account root user* to the IAM console at https://console.aws.amazon.com/iam/.

   Note

   We strongly recommend that you adhere to the best practice of using the **`Administrator`** IAM user below and securely lock away the root user credentials. Sign in as the root user only to perform a few account and service management tasks.

2. In the navigation pane, choose **Users** and then choose **Add user**.
3. For **User name**, type **`Administrator`**.
4. Select the check box next to **AWS Management Console access**, select **Custom password**, and then type your new password in the text box. By default, AWS forces the new user to create a new password when first signing in. You can optionally clear the check box next to **User must create a new password at next sign-in** to allow the new user to reset their password after they sign in.
5. Choose **Next: Permissions**.
6. On the **Set permissions** page, choose **Add user to group**.

7. Choose **Create group**.
8. In the **Create group** dialog box, for **Group name** type `Administrators`.
9. For **Filter policies**, select the check box for **AWS managed - job function**.
10. In the policy list, select the check box for **AdministratorAccess**. Then choose **Create group**.

    Note

    You must activate IAM user and role access to Billing before you can use the `AdministratorAccess` permissions to access the AWS Billing and Cost Management console. To do this, follow the instructions in Step 1 of the tutorial about delegating access to the billing console.

11. Back in the list of groups, select the check box for your new group. Choose **Refresh** if necessary to see the group in the list.
12. Choose **Next: Tagging**.
13. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM Identities.
14. Choose **Next: Review** to see the list of group memberships to be added to the new user. When you are ready to proceed, choose **Create user**.

You can use this same process to create more groups and users and to give your users access to your AWS account resources

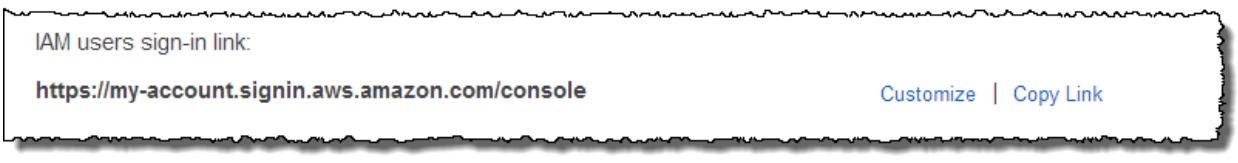
How Users Sign In to Your Account

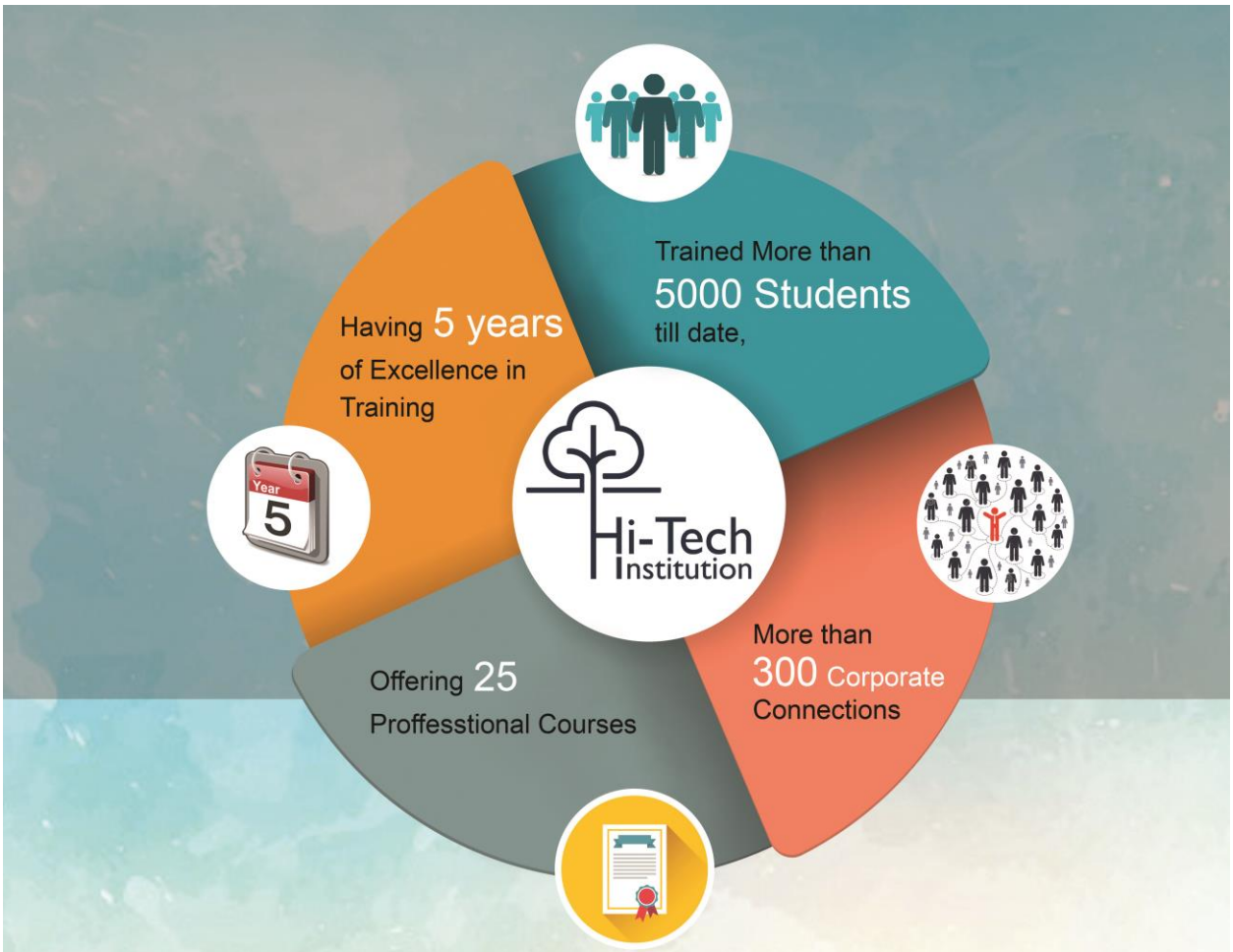
After you create IAM users (with passwords), those users can sign in to the AWS Management Console using your account ID or alias, or from a custom URL that includes your account ID.

Note

If your company has an existing identity system, you might want to create a single sign-on (SSO) option. SSO gives users access to the AWS Management Console without requiring them to have an IAM user identity. SSO also eliminates the need for users to sign in to your organization's site and to AWS separately.

Before you create a sign-in URL for your account, you create an account alias so that the URL includes your account name instead of an account ID. You can find the sign-in URL for an account on the IAM console dashboard.





TOP RECRUITERS

Capgemini · Mindtree · Tech Mahindra · MISYS · EMC² · subex · FIS
L&T Technology Services · virtusa · TRIGENT · HUAWEI · KENNAMETAL · HCL
Atos · ALLEGIS GROUP · izmo · AWPL · Softtek · Infosys · BOSCH
hp · infinite · CGI · OPENTEXT · IGATE
mellow infosystems · GLOBAL EDGE · ARTECH · Fonezela virtual business assistant · CORPORATE LADDER
Simbus · BENISON TECHNOLOGIES · BRISTLECONE Your Supply Chain. Optimized. · THOUGHTFOCUS · and more...

Hi-Tech Institution

50% offer for School or College students

30% offer for IT Employees

Above offer applicable only technical courses. Terms and conditions apply

Email Us
operations@hitechins.in

Search Us
www.hitechins.in

CONTACT US

7092 90 91 92 / 82 20 21 7640

PONDICHERRY
No.32, 100 feet road,
Ellaipillaichavady,
Pondicherry – 605 005,
Nearby Rajiv Gandhi Hospital

TAMBARAM
No.24, Chithi Vinayagar Kovil street,
KamarajNagar, Tambaram Sanatorium,
Chennai – 600 047,
Nearby Sanatorium Railway Station

VELACHERRY
No: 21, Officer Colony,
100 feet road, VijayaNagar,
Velacherry – 600 042,
Nearby Sathya Home Appliances

Locations
Chennai & Pondicherry