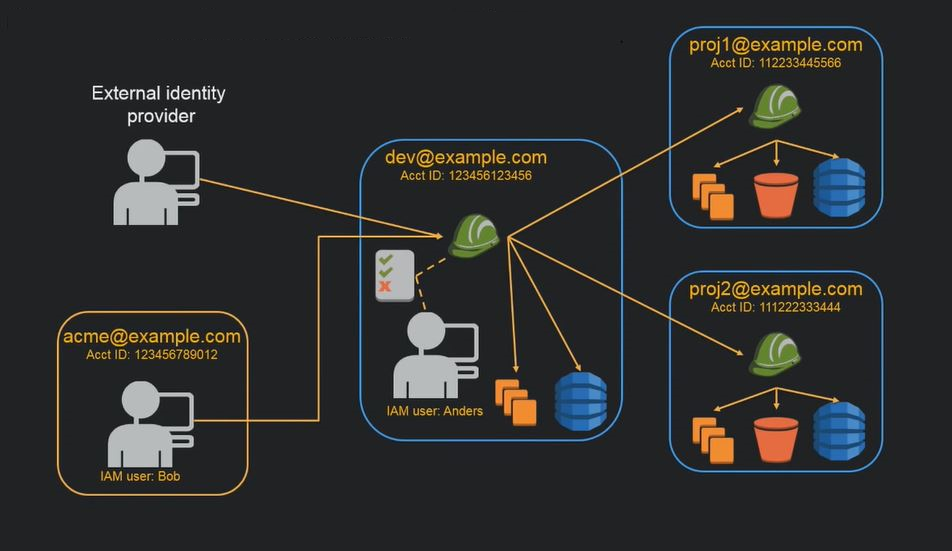
# **USE CASE-1:**



# **Description**

John is the founder of Example Corp. Upon starting the company, he creates his own AWS account and uses AWS products by himself. Then he hires employees to work as developers, admins, testers, managers, and system administrators.

John uses the AWS Management Console with the AWS account root user credentials to create a user for himself called John, and a group called Admins. He gives the Admins group permissions to perform all actions on all the AWS account's resources using the AWS managed policy **AdministratorAccess**. Then he adds the John user to the Admins group. For a step-by-step guide to creating an Administrators group and an IAM user for yourself, then adding your user to the Administrators group, see **Creating Your First IAM Admin User and Group**.

At this point, John can stop using the root user's credentials to interact with AWS, and instead he begins using only his user credentials.

John also creates a group called ***AllUsers*** so that he can easily apply any account-wide permissions to all users in the AWS account. He adds himself to the group. He then creates a group called ***Developers***, a group called ***Testers***, a group called ***Managers***, and a group called ***SysAdmins***. He creates users for each of his employees, and puts the users in their respective groups. He also adds them all to the ***AllUsers*** group. For information about creating groups, see Creating IAM Groups. For information about creating users, see Creating an IAM User in Your AWS Account. For information about adding users to groups, see Managing IAM Groups.

# **Acceptance criteria**

Create an AWS admin privileged IAM user via IAM with EC2

Create an AWS admin privileged IAM user via IAM with S3

# **Instructions**

## Use Case for IAM with Amazon EC2

A company like Example Corp typically uses IAM to interact with services like Amazon EC2. To understand this part of the use case, you need a basic understanding of Amazon EC2. For more information about Amazon EC2, go to the [Amazon EC2 User Guide for Linux Instances](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/).

### Amazon EC2 Permissions for the Groups

To provide "perimeter" control, John attaches a policy to the AllUsers group. This policy denies any AWS request from a user if the originating IP address is outside Example Corp's corporate network.

At Example Corp, different groups require different permissions:

* **System administrators** – Need permission to create and manage AMIs, instances, snapshots, volumes, security groups, and so on. John attaches a policy to the SysAdmins group that gives members of the group permission to use all the Amazon EC2 actions.
* **Developers** – Need the ability to work with instances only. John therefore attaches a policy to the Developers group that allows developers to call DescribeInstances, RunInstances, StopInstances, StartInstances, and TerminateInstances.

**Note**

Amazon EC2 uses SSH keys, Windows passwords, and security groups to control who has access to the operating system of specific Amazon EC2 instances. There's no method in the IAM system to allow or deny access to the operating system of a specific instance.

* **Managers** – Should not be able to perform any Amazon EC2 actions except listing the Amazon EC2 resources currently available. Therefore, John attaches a policy to the Managers group that only lets them call Amazon EC2 "Describe" API operations.

For examples of what these policies might look like, see [Example IAM Identity-Based Policies](https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_examples.html) and [Using AWS Identity and Access Management](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/index.html?UsingIAM.html) in the Amazon EC2 User Guide for Linux Instances.

### User's Role Change

At some point, one of the developers, Paulo, changes roles and becomes a manager. John moves Paulo from the Developers group to the Managers group. Now that he's in the Managers group, Paulo's ability to interact with Amazon EC2 instances is limited. He can't launch or start instances. He also can't stop or terminate existing instances, even if he was the user who launched or started the instance. He can list only the instances that Example Corp users have launched.

## Use Case for IAM with Amazon S3

Companies like Example Corp would also typically use IAM with Amazon S3. John has created an Amazon S3 bucket for the company called example\_bucket.

### Creation of Other Users and Groups

As employees, Zhang and Mary each need to be able to create their own data in the company's bucket. They also need to read and write shared data that all developers work on. To enable this, John logically arranges the data in example\_bucket using an Amazon S3 key prefix scheme as shown in the following figure.

/example\_bucket

/home

/zhang

/mary

/share

/developers

/managers

John divides the master /example\_bucket into a set of home directories for each employee, and a shared area for groups of developers and managers.

Now John creates a set of policies to assign permissions to the users and groups:

* **Home directory access for Zhang** – John attaches a policy to Zhang that lets him read, write, and list any objects with the Amazon S3 key prefix /example\_bucket/home/Zhang/
* **Home directory access for Mary** – John attaches a policy to Mary that lets her read, write, and list any objects with the Amazon S3 key prefix /example\_bucket/home/mary/
* **Shared directory access for the Developers group** – John attaches a policy to the group that lets developers read, write, and list any objects in /example\_bucket/share/developers/
* **Shared directory access for the Managers group** – John attaches a policy to the group that lets managers read, write, and list objects in /example\_bucket/share/managers/

**Note**

Amazon S3 doesn't automatically give a user who creates a bucket or object permission to perform other actions on that bucket or object. Therefore, in your IAM policies, you must explicitly give users permission to use the Amazon S3 resources they create.

# **USE CASE-2:**

# **Description**

Creation of Admin user to perform programmatic changes on your AWS infrastructure. IAM users in concept are separate to the root access user that you are originally log in to a new account using the AWS Management Console. IAM Users are something created after this point and which Amazon recommend you use to access infrastructure and other resources instead.

# **Acceptance criteria**

Create an AWS admin privileged IAM user

# **Instructions**

1. Create IAM user for yourself to access AWS (not the AWS root account credentials)
2. AWS console: <http://aws.amazon.com/>
3. select your login name dropdown
4. Security credentials (referred to as IAM hereon (AWS - Identity Access Management))
5. Users (from navigation pane on the left)
6. Create New User (ensure Generate an access key for each user checked if AWS CLI is to be used by this user)
7. Also create an AWS console login for this account  
   Setup a password for this account (it can be auto generated if you secure it in KeePass):  
   IAM > Users > Security Credentials tab > Manage Password can be used to change the set password
8. Create an Administrators Group, assign a policy to it, then assign this IAM user to the group:
9. note: membership to the group conveys its permissions  
   > <http://docs.aws.amazon.com/IAM/latest/UserGuide/getting-started_create-admin-group.html>
10. select Groups
11. Create New Group (name:  Administrators) > Next Step
12. select the check box next to the Administrator Access policy > Next Step
13. Create New Group
14. Users
15. check the user you just created
16. from User Actions select Add User to Groups
17. Add to the Administrators group just created
18. Apply an IAM password policy  
    IAM > Account Settings  
    note: IAM settings are available across regions.
19. Note this admin user-specific sign-in URL for the AWS Management console  
    Users can sign in to the AWS Management Console using a special URL:  
    https://<AWS-account-ID-or-alias>.[signin.aws.amazon.com/console](http://signin.aws.amazon.com/console)
20. Look for this URL at the top of IAM > Dashboard in the main window.
21. You can sign-in as the root account using the link beneath this IAM User login window: "Sign-in using root account credentials". This is needed in order to see Billing information which even Administrator group users cannot see (this provokes an IAM error if you try).