## Assignment (2.6)

### Brief

Objective:

The objective of this assignment is to gain a deeper understanding of AWS permissions and policies and how they can be used to secure access to AWS resources.

Instructions:

* Research and investigate the concepts of AWS IAM (Identity and Access Management) and the different types of permissions and policies available.
* Create a sample AWS environment, including at least two different types of resources (e.g. an S3 bucket, an EC2 instance).
* Use IAM to create and apply permissions and policies to the resources, including creating and managing users, groups, and roles.
* Write a report summarizing your experience with IAM, including an overview of the permissions and policies used, any challenges encountered, and any benefits or limitations observed.
* Provide examples of how these permissions and policies can be used in a real-world scenario.

Deliverables:

* A sample AWS environment that demonstrates the use of AWS IAM permissions and policies
* A report summarizing your experience with IAM.

### Submission

* Submit the URL of the GitHub Repository that contains your work to NTU black board.
* Should you reference the work of your classmate(s) or online resources, give them credit by adding either the name of your classmate or URL.

**A sample AWS environment that demonstrates the use of AWS IAM permissions and policies**

Overview of IAM

Diagram

Description automatically generated

By creating policies and tying them to IAM identities (user, group of users or roles) or AWS resources, we can manage access in AWS.

AWS Screenshot for the user



Say, if we want to have more specific policies by means of Customer managed, instead of AWS managed, which we think are too general and not quite satisfied with our requirements for S3. As such, here we create a new S3 bucketcontrolpolicy to list , create and delete bucket with a right to access to all resources by using visual editor. As a result, we can attach this newly created policy to our user/group of users (in this case we assign the user to the group, the user get the permission right inherited from the group as we have attached the policy to it directly instead of the user on this occasion) to let them have the permission to perform all those tasks (list, create and delete buckets). For the case of role, it works a little different as this time we select the service (e.g.EC2) that will use this role e.g it allows EC2 instance to call AWS service on our behalf or allows or allow Lambda functions to call AWS service on our behalf.

**A report summarizing your experience with IAM**.

Implementation of IAM helps an organization in the management of authorization and privileges throughout the system for increased security. Most important of all, IAM solutions help in improving productivity through automation of manual security tasks. So, IAM definitely seems like a necessary security measure if you want to move to the AWS cloud, First of all, it is essential to know the difference between Roles, Users, and Groups for the effective deployment of access security in a particular cloud environment.

IAM Users are account objects that help individual users in accessing an AWS environment with a particular set of credentials. While IAM Groups and objects with specifically assigned permissions through Policies that can allow access to specific resources for Group members. IAM Roles are objects in the IAM with certain Policy permissions.

However, IAM Roles do not associate with Users like in the case of Groups. On the contrary, IAM roles associated with instances at the time of launch, thereby enabling the instance to follow the permissions in the role. As a result, there is no need for local storage of Access Keys on the concerned instance.

There are two common scenarios of using a role. The first method involves an interactive approach for using AWS IAM Roles in the IAM console. The second option involves a programmatic approach for using IAM roles with the AWS CLI, API, or Tools for Windows PowerShell. Other different scenarios in which AWS IAM roles find a suitable applications include:

* Providing access for an IAM user in one AWS account under your ownership for accessing resources in another account under your ownership.
* Provision of access to IAM users in AWS accounts under the ownership of third parties.
* Providing access for externally authenticated users, also known as identity federation.
* Provision of access for AWS services offered to AWS resources.

Now, let us take the final step in this discussion on the basics of AWS IAM roles and policies. Till now, the discussion’s emphasis was on IAM roles. Let us find out more about AWS IAM policies. Users can manage access in AWS through the creation of policies and then associating them with IAM identities or AWS resources. The policy is an AWS object that defines permissions of identity or resource, with which it associates. AWS undertakes an evaluation of these policies upon the request by a principal entity such as user or role. Permissions in the policies help in determining the allowing or denying of requests.

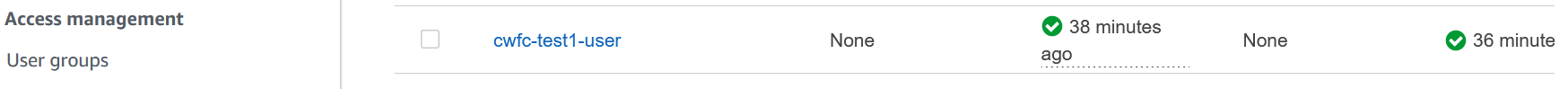
Conclusion:

Identity and access management (IAM) is definitely a staple requirement for shifting your business to the cloud. AWS IAM roles serve a wide range of functionalities to ensure the security of AWS resources. However, the effectiveness of IAM roles and policies on AWS depends considerably on the use of best practices.

19Now you’ve had some experience with creating your own permissions, try out the below questions:

* 1. You need access to create EC2 instances. How would you edit the permissions for this?
  2. You need access to view RDS without creating them. How would you edit the permissions for this?
  3. You need access to create and manage VPCs. How would you edit the permissions for this?

**Screenshot of AWS**

****

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, application, table

Description automatically generated**