[Instructions] Creating IAM Users

Let us get into the details related to IAM Users.

* Login as a root or user with Administrator policy
* Go to IAM Service
* Create the user and review the options

[Instructions] Logging into AWS Management Console using IAM User

Here are the instructions to login into Management Console using IAM User.

* A link will be generated for each AWS account with an AWS account id.
* We can use that link to launch the login page with the Account id pre-populated.
* Enter the IAM username and password shared with you. It will ask to reset the password if **Users must create a new password at next sign-in** is selected while creating the account.
* Make sure to reset the password adhering to the password policy.

[Instructions and Commands] Validate Programmatic Access to IAM User

Let us go ahead and validate programmatic access by configuring AWS CLI using the access key and secret key generated for our new account.

* You need to make sure that AWS CLI is set up on your system.
* Launch the terminal or shell and run the **aws configure** command to configure with new keys.
* You can create a sample bucket and validate to confirm that your credentials are configured to the user in the right account.
* Here are the steps I have followed.
  + Configure credentials using profile by name **itvadmin**.

1. aws configure --profile itvadmin

* Create an s3 bucket with some unique name.
* Run the following command to list the newly created bucket.

1. aws s3 ls --profile itvadmin

[Instructions and Commands] IAM Identity-based Policies

Let us understand [IAM Identity-based Policies](https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html#policies_id-based). We will be focusing on Predefined policies for now.

* Permissions are assigned to policies using JSON Syntax.
* Policies are typically attached to either group or role. It is also possible to attach policies directly to users.
* Users, groups, and roles are also known as **IAM identities**.
* We can attach more than one policy to an IAM identity.
* We have predefined policies which we can leverage and also we can define custom policies.
* Let us review the following policies to understand how permissions on services are typically defined.
  + [AmazonS3FullAccess](https://console.aws.amazon.com/iam/home?region=us-west-1#/policies/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAmazonS3FullAccess)
  + [AmazonS3ReadOnlyAccess](https://console.aws.amazon.com/iam/home?region=us-west-1#/policies/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAmazonS3ReadOnlyAccess)
* Here are the key terms which you should be familiar with related to policies.
  + Effect - This is where you typically define Allow or Deny
  + Action - This is where you typically define the type of actions that can be performed
  + Resource - We can control the permissions over the resources that are related to the Effect.
* For example:
  + Effect - s3 (service)
  + Action - Get, List, Put, Delete, etc.
  + Resource - Buckets or objects within the bucket we have created.
* Let us perform a few tasks related to Identity-based Policies.
  + Create a new user **itvsupport1** with only programmatic access.
  + Configure AWS CLI with profile **itvsupport1**.
  + Attach [AmazonS3ReadOnlyAccess](https://console.aws.amazon.com/iam/home?region=us-west-1#/policies/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAmazonS3ReadOnlyAccess) to **itvsupport1**.
  + Make sure a bucket is created using AWS Web Console by logging in as admin or root user. I will be creating a bucket by the name **dg-retail**. If you already have such a bucket, you can directly copy files into S3.
  + Try running this command to copy files into the bucket.

1. aws s3 cp ~/Research/data/retail\_db s3://dg-retail1/retail\_db/ \
2. --recursive \
3. --exclude "\*.sql" \
4. --exclude "README.md" \
5. --profile itvsupport1

[Instructions and Commands] Managing IAM Groups

Let us go through the details related to IAM Groups.

* We typically have users in IAM Groups.
* Permissions are typically defined using Policies and policies are assigned to Groups.
* All the users in the group will inherit all the permissions associated with the group.

Let us perform these tasks to make sure that we are comfortable in dealing with groups.

* Let us create two groups.
  + itvadmin
  + itvsupport
* Add user itvadmin to group itvadmin
* Add user itvsupport1 to group itvsupport
* Detach **AdministratorAccess** policy from user itvadmin and attach it to group itvadmin
* Detach **AmazonS3ReadOnlyAccess** policy from user itvsupport1 and attach it to group itvsupport
* Run the below command as an itvadmin user to confirm that the user account has all the admin permissions that were assigned earlier directly.

1. aws s3 rm s3://dg-retail1/retail\_db/ --recursive --profile itvadmin
3. aws s3 cp ~/Research/data/retail\_db s3://dg-retail1/retail\_db/ \
4. --recursive \
5. --exclude "\*.sql" \
6. --exclude "README.md" \
7. --profile itvadmin

* Run the below commands as itvsupport1 to confirm that the user account has only read-only access on s3.

1. # Both these commands will fail.
2. aws s3 rm s3://dg-retail1/retail\_db/ --recursive --profile itvsupport1
4. aws s3 cp ~/Research/data/retail\_db s3://dg-retail1/retail\_db/ \
5. --recursive \
6. --exclude "\*.sql" \
7. --exclude "README.md" \
8. --profile itvsupport1
10. # This command should work as the user have s3 read-only access
11. aws s3 ls dg-retail1 --recursive --profile itvsupport1

[Instructions and Commands] Managing IAM Roles

Roles are used to assigning the permissions on one service to another.

* We typically create roles and attach policies with them.
* All the permissions associated with the roles via policies will be inherited by the service when we attach a role to it.

Let us perform these tasks to understand how the roles are defined and used.

* Create a role by name itvsupport and attach the **AmazonS3ReadOnlyAccess** to it. Make sure to choose the service as EC2 as we want to use this role to attach to the EC2 instance.
* Launch EC2 instance using this role. We will be using the Amazon Linux image as it will come with AWS CLI already setup.
* We don’t need to configure AWS CLI as the permissions are assigned via role to this EC2 instance.
* If you provision, EC2 instance with other operating systems than Amazon Linux, then you need to install AWS CLI first.
* Login to the EC2 instance and run these commands.

1. # This command will fail.
2. aws s3 rm s3://dg-retail1/retail\_db/ --recursive
4. # This command should work as the role has s3 read-only access
5. aws s3 ls dg-retail1 --recursive

[Instructions and Commands] Overview of Custom Policies

We can provide finer granular permissions on any AWS Service or component using Custom Policies. Let us do a lab to understand how we can use custom policies to grant specific permissions to a user on a specific bucket.

* Go to Policies and click on **Create Policy**. We will be using **ITVSupportS3RetailDBAll** as the name for the policy.
* We can enter the service and use the wizard to create the policy or we can import from managed policies and improvise on top of it.
* Let us import **AmazonS3ReadOnlyAccess** and then customize it. We will give the get, put and delete permissions on the retail\_db folder in the dg-retail1 bucket along with list all buckets permission.

1. {
2. "Version": "2012-10-17",
3. "Statement": [
4. {
5. "Effect": "Allow",
6. "Action": [
7. "s3:\*"
8. ],
9. "Resource": "arn:aws:s3:::dg-retail1/retail\_db\*"
10. },
11. {
12. "Effect": "Allow",
13. "Action": [
14. "s3:List\*"
15. ],
16. "Resource": "\*"
17. }
18. ]
19. }

* Once the policy is created, detach the existing policy from the itvsupport group and attach the new policy.
* You can validate by running the below commands. We should be able to delete and copy the files again using **the itvsupport1** profile as **itvsupport1** is part of the **itvsupport** group.

1. aws s3 rm s3://dg-retail1/retail\_db/ --recursive --profile itvsupport1
3. aws s3 cp ~/Research/data/retail\_db s3://dg-retail1/retail\_db/ \
4. --recursive \
5. --exclude "\*.sql" \
6. --exclude "README.md" \
7. --profile itvsupport1
9. aws s3 ls dg-retail1 --recursive --profile itvsupport1
11. # This command should fail
12. aws s3 mb s3://dg-retail2 --profile itvsupport1

[Instructions and Commands] Managing IAM using AWS CLI

As we have understood how to manage IAM using Web Console, let us get into the details about managing IAM using CLI.

* We should be able to manage IAM components using the **aws iam** command.

1. aws iam list-users --profile itvadmin
2. aws iam list-groups --profile itvadmin
3. aws iam list-roles --profile itvadmin
5. # Lists all AWS Managed Policies as well as custom policies
6. aws iam list-policies --profile itvadmin
8. # List only custom policies
9. aws iam list-policies --scope Local --profile itvadmin

* Let us create a user **itvsupport2** and assign it to the group **itvsupport**.

1. aws iam create-user --user-name itvsupport2 --profile itvadmin
2. aws iam list-users --profile itvadmin
4. aws iam add-user-to-group \
5. --group-name itvsupport \
6. --user-name itvsupport2 \
7. --profile itvadmin
9. aws iam list-groups-for-user \
10. --user-name itvsupport2 \
11. --profile itvadmin

* Let us remove the user from the group and then delete. We cannot delete the user when he is part of the group.

1. aws iam remove-user-from-group \
2. --group-name itvsupport \
3. --user-name itvsupport2 \
4. --profile itvadmin
5. aws iam delete-user \
6. --user-name itvsupport2 \
7. --profile itvadmin
9. aws iam list-users --profile itvadmin