**Hands-On Project on Migrating users from On-premises to AWS Cloud**

**Worked as a Cloud Specialist**

**Technologies Used:**

* Based on the data on the excel sheet, I created three User groups (teams) on AWS – Identity and Access Management (IAM) and added appropriate policies depending on the functionalities of each team.
* Automated the process of adding 14 On-premises users to their appropriate User group in AWS Cloud. On the csv file:
  + - Rename headings – change email to user and team to group.
    - Add another column to the right titled “password”, and populated the column with “GeneralPassword123@”
    - Edit the file to get first and last names together so that the group-name will match names under User group in IAM - AWS console. Do this by using the find and replace function of Excel.
    - Save the edited file named “users.csv”.
* On AWS Cloud Shell, used the AWS Command Line Interface (CLI) to create users –

On Cloudshell:

* Install a utility in CloudShell, that is ‘dos2unix’ using sudo yum install dos2unix -y #this command installs the utility “dos2unix” on a linux system, ensuring the installation process completes without prompting the user for confirmation.
* Next, download the shell script file to create the users in AWS, use wget command to download the iam-create- script file stored in s3 bucket that will automate the whole process. For now, run wget <https://tcb-bootcamps.s3.amazonaws.com/bootcamp-aws/en/aws-iam-create-user.sh>

“Wget” is used to download the script

* OR just upload the script “aws-iam-create-user.sh” directly if you have it.

“cat” to open the script

* Use cat aws-iam-create-user.sh to open the script.
* Use chmod +x aws-iam-create-user.sh to change the permission of the script in a Unix-like operating system so we can execute it.
* Upload the “users.cvs” file on CloudShell. Click on “Actions” at the right top.
* Type the command ls -ltr to check all the files in the root directory.
* Check the content of “users.csv” using cat users.csv
* **The automation script will take as an input, users2.csv, it will read the username, read which group is associated to that username and also set the password.**
* Run the command ./aws-iam-create-user.sh user2.csv to run the script.
* Check Users under IAM in AWS Console to see the result.
* **GOOOOOOOOOOD JOB!!!** 😊😊😊
* Associate a policy on AWS Console to allow the IAM users to change their own password since the current password for all the users is “GeneralPassword123@”.
  + IAM 🡪 Users groups 🡪 Click on a group 🡪 Permissions 🡪Attach permissions (Attach policies) 🡪 Search for IAMUserChangePassword
* Do this for all Users group.
* For each user to change their password:
  + copy the “sign-in URL for IAM users” on IAM Dashboard.
  + Paste on a new browser.
  + Put in the old password which is “GeneralPassword123@” and add your new password with at least an uppercase, a special character e.g !@ and the new password should be at least 8 characters long.

To grant access to an IAM user to create a custom policy like creating an email address:

* Enabled multi-factor authentication (MFA) for the **root user** using google authentication to add extra security to my AWS account. To do this:
  + Download “Google Authenticator” on your phone.
  + On AWS console, under IAM dashboard, click “Add MFA”.
  + Click “Activate MFA”.
  + Click “Virtual MFA device”. Note some org like physical token like “YubiKey”.
  + Click “Show QR code”.
  + Scan the QR code with your phone camera and type in the first two consecutive token codes.

**Note that after adding MFA, to access your AWS account, you’ll need a combination of your password and the token on google authenticator on your mobile device (phone).**

A screenshot of a computer

Description automatically generated

The attached picture above shows that all the users successfully migrated to AWS Cloud.

A screenshot of a computer

Description automatically generated

Confirming if “KennyBrown” is a Database Administrator from the result in AWS Console. From the attached picture above, she is.

A screenshot of a computer

Description automatically generated

Confirming if “KennyBrown” is a Database Administrator from the result in AWS CloudShell. From the attached picture above, she is.

A screenshot of a computer

Description automatically generated

Confirming if “KennyBrown” is a Database Administrator from the given data (Excel Sheet). From the attached picture above, she is.

**Kenny Brown, a Database Administrator, only has FullAccess to the AWS Database i.e RDS but no access to the EC2 instances.**

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Kenny Brown, a Database Administrator, only has full access to the AWS Database i.e RDS but no access to the EC2 instances.**

From the below screenshot, Nick Cosy is a Software Engineer and has only ReadOnlyAccess to the EC2 instance and RDS, that is the reason why the attached picture is showing Access Denied.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

These are the policies all Software Engineers have. Mostly ReadOnlyAccess

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated From the above image, Nick Cosy do not have access to RDS as expected.

A screenshot of a computer

Description automatically generated