

IAM Control

User Guide

Document Information

Distribution:	External
Date:	14 th September 2024
Author:	Keval Sheth
Document code:	001
Version:	2.0

Approval

Date	Name	Title
25/09/2024	Keval Shet	Solutions Architect
14/09/2024	Chris Fleming	Solutions Architect
	Gordon Scobie	Chief Technical Officer

Table of Contents

Introduction	2
Login Process.....	2
Architecture.....	2
IAM User Setup	3
Multi-Factor Authentication (MFA).....	3
Switch Roles using AWS Console	4
Switch Roles Browser Extension	5
AWS with your preferred IDE	5
Creating Access Keys	5
Switch Roles using AWS CLI.....	6
Credentials File	7
Config File	7
Testing.....	7

Introduction

AWS [Identity and Access Management](#) (IAM) is an Amazon web service that helps you securely control access to AWS resources. IAM is used to control who is authenticated (*signed in*) and authorized (*has permissions*) to use resources.

This document details the access control settings used across your Amazon Web Services accounts, to establish a secure single point of access using role-based configuration to prevent individual access to the “root account” with unlimited powers across all accounts and no permission restrictions at all. Access to the accounts require the assignment and assumption of a “*Role*” to complete tasks within specific accounts.

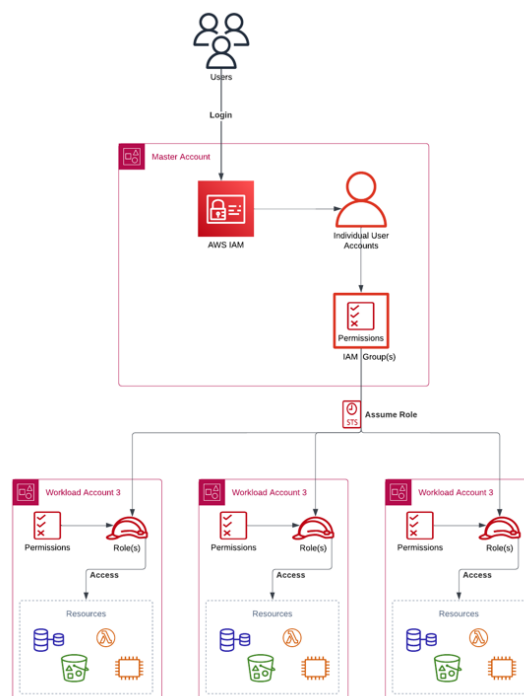
Login Process

A Master AWS account is in place which is the only account users are able to directly login to (with a username and password) unless in exceptional circumstances where necessary to login into other accounts with the root user (typical scenarios include creating support tickets & changing the email address associated with the account).

From the master account, there are roles in every other account that will need to be ‘assumed’. These roles will provide certain functionality based on job role and access to the roles is based on the groups an individual IAM User is attached to.

Architecture

The diagram below shows the login process between the master account and subsequent workload accounts.



IAM User Setup

All IAM Users should be logged into through the Master AWS Account, this login account will not change. When you are initially provided with your username and password you will be prompted to change to a password of your choosing, this must follow the set account password policy, which is typically:

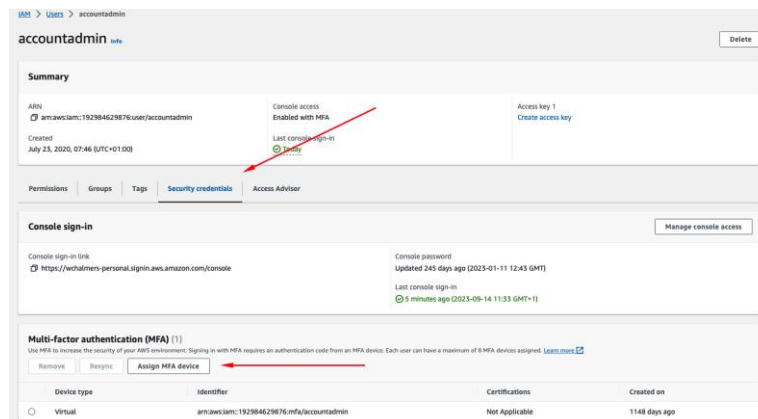
- Minimum Length: 10 characters
- At least 1 symbol
- At least 1 number
- At least 1 upper and lowercase character

Note: You may be required to periodically change your password, in this case you should follow the same password policy as above.

Multi-Factor Authentication (MFA)

By default on the account, all permissions are denied for your IAM User unless Multi Factor Authentication is enabled. To enable this, please follow the steps below:

- Navigate to the IAM Users section of the AWS Console
 - AWS Console > IAM > Users
 - Link: <https://us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/users>
- Click on your IAM User (it will be highlighted in blue)
- Go to the Security Credentials tab
- Select 'Assign MFA Device' & follow the setup instructions provided on the console

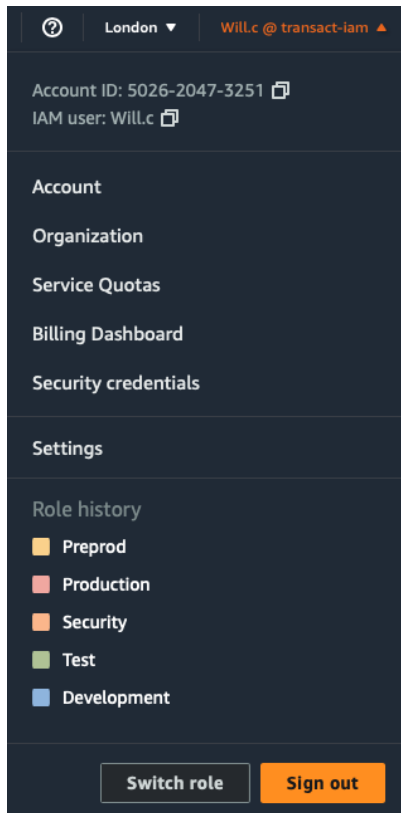


- Once complete, you will need to log out, and then log back into AWS for the permissions to then update.

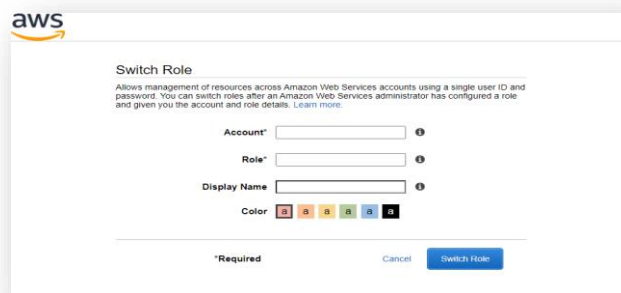
Switch Roles using AWS Console

To switch roles between AWS Accounts, please follow the steps below.

- On the very top right of the AWS Console you will see text: [username] @ [account name]
- From the dropdown, select 'Switch Role'



- From there, you will be brought to the page below, now:
- Enter the Account ID
- Enter the Role name
- Add a memorable name for the role (optional)
- Select a colour you want to associate to the role (optional)
- Click Switch Role



Tip: You can find both the Account ID and the Role's name from it's ARN

arn:aws:iam::123456123456:role/DevOps-Engineer-Prod

Account ID Role Name

IAM Role ARN

Note: Once you have switched roles, the previous 5 will be remembered in the history by default.

There is also a supplementary video walkthrough of this process which can be found [here](#)

Switch Roles Browser Extension

To make switching roles easier, there is a 3rd party browser extension called AWS Extend Switch Roles that you can install which you can input all your roles which will all be remembered, instead of only the last 5 that AWS remembers.

[Chrome Download Link](#) | [Safari Download Link](#) | [Firefox Download Link](#)

To Use:

Simply paste in the role details, like above, and optionally add a colour hex code to help differentiate them (note the American spelling of colour).

To switch roles with this extension, make sure you are logged into the aws console, click the extension button, and then click the role you would like to assume.

```
[profile Production]
role_arn = arn:aws:iam:: 123456123456:role/Developer-Prod
region = eu-west-1
color = ff2600

[profile Development]
role_arn = arn:aws:iam::123456123456:role/Developer-Dev
region = eu-west-1
color = 00ccff
```

AWS with your preferred IDE

The official AWS Toolkit extension is available with most popular IDEs, this will allow you to create, debug, and deploy applications through AWS more easily.

For this to work it firstly requires the AWS CLI to be installed: [Install Guide](#)

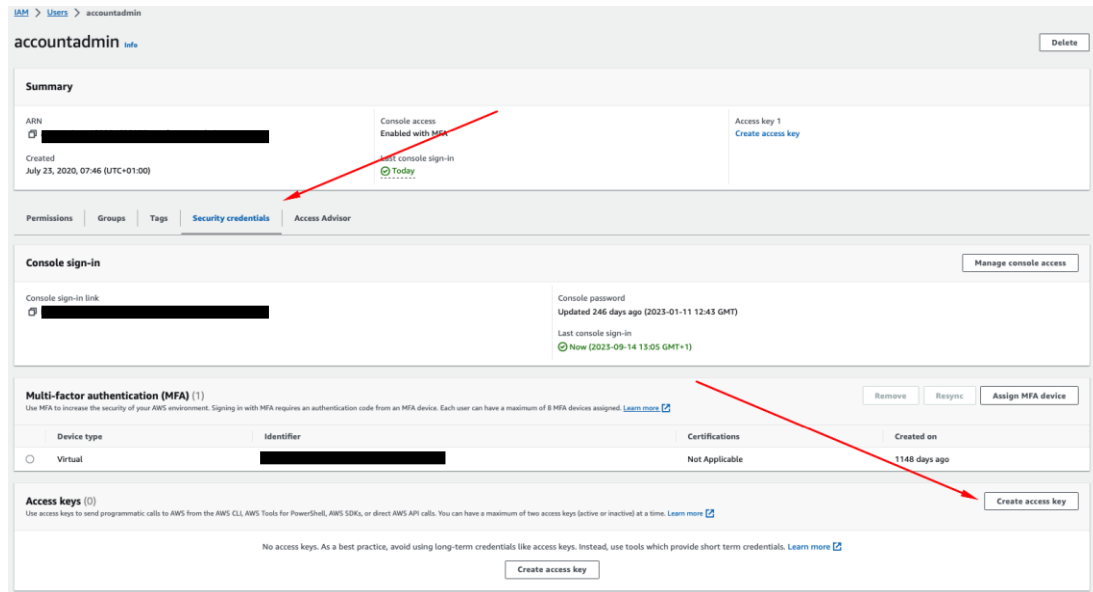
Browser	Extension Download Link
Visual Studio	Download Link
Visual Studio Code	Download Link
IntelliJ IDEA	Download Link
JetBrains	Download Link
PyCharm	Download Link
Rider	Download Link

Creating Access Keys

Access Keys are a set of credentials that allow you to interact with AWS through the command line. Two sets of CLI credentials may be active at any one time, however it is strongly recommended that only 1 set is ever in use, and the second set be saved for credential rotation periods only. These are very similar to GitHub Access Tokens.

To generate access keys for your IAM User:

- Navigate to the [IAM Users section](#) of the AWS Console
 - AWS Console > IAM > Users
- Click on your IAM User (it will be highlighted in blue)
- Go to the Security Credentials tab
- Select 'Assign MFA Device' & follow the setup instructions provided on the console
 - After Clicking Create Access Key, select 'Command Line Interface (CLI)' as your use case



The screenshot shows the AWS IAM console for the 'accountadmin' user. The 'Security credentials' tab is selected. It displays the user's ARN, console access status (Enabled with MFA), and a 'Create access key' button. Below this, the 'Console sign-in' section shows the console sign-in link and password. The 'Multi-factor authentication (MFA)' section shows the user is not enrolled. The 'Access keys' section shows no access keys are present, with a 'Create access key' button. Red arrows in the original image point to the 'Console access' status and the 'Create access key' button.

Note: Access Keys can't be viewed again after creation, so be careful to store them someplace safe. It's also strongly recommended to rotate them every 90 days.

Switch Roles using AWS CLI

The AWS CLI allows you to interact with AWS through the command line rather than the web console. This requires some additional setup to configure the AWS Roles.

Before starting this you will need to generate a set of Access keys, which is documented further up in this guide & have the AWS CLI installed, a guide can be found here: [Install Guide](#)

To enable the AWS CLI to use your credentials, there are two key files that you will need to create, a config file and a credentials file.

File	Linux Location	Windows Location
Credentials	~/.aws/credentials	C:\Users\username\.aws\credentials
Config	~/.aws/config	C:\Users\username\.aws\config

Credentials File

```
1 [profile name]
2 aws_access_key_id=AKIAIOSFODNN7EXAMPLE
3 aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
4 region=eu-west-1
```

Note: replace 'profile name' on line 1 with a friendly name but keep it inside the square brackets.

You may also choose to call this profile 'default' and if so, it will be set as the default user for all commands run. If you have multiple IAM Users, you will need to add the `--profile profile-name` flag at the end of your CLI commands to specify the correct user to perform the action as.

Config File

```
1 [profile profile name]
2 role_arn = arn:aws:iam::123456123456:role/DevOps-Engineer-Prod
3 source_profile = profile name set in above credentials
4 region=eu-west-1
```

Note: Replace profile with a friendly name (keeping the first 'profile'), for example [profile prod-account].

This is the file where all your roles go, so when you want to assume a role, simply add the `--profile profile-name` flag at the end of your commands, like above.

Testing

To test that your command worked you can run the following command to confirm that you are able to successfully connect (replacing prod with whatever your profile name is): `aws sts get-caller-identity --profile prod`

The above command works for both IAM users (defined in the credentials file) and Roles (defined in the config file).