AWS Cloud for beginner

Instructor: Linh Nguyen -
Engineering Consultant, AWS Cloud Solution Architect
Level: Beginner

Identity and Access Management (IAM)

"Có 2 thứ ngăn cản chúng ta trên con đường trở thành chuyên gia Cloud:

1. *Security*
2. *Networking"*

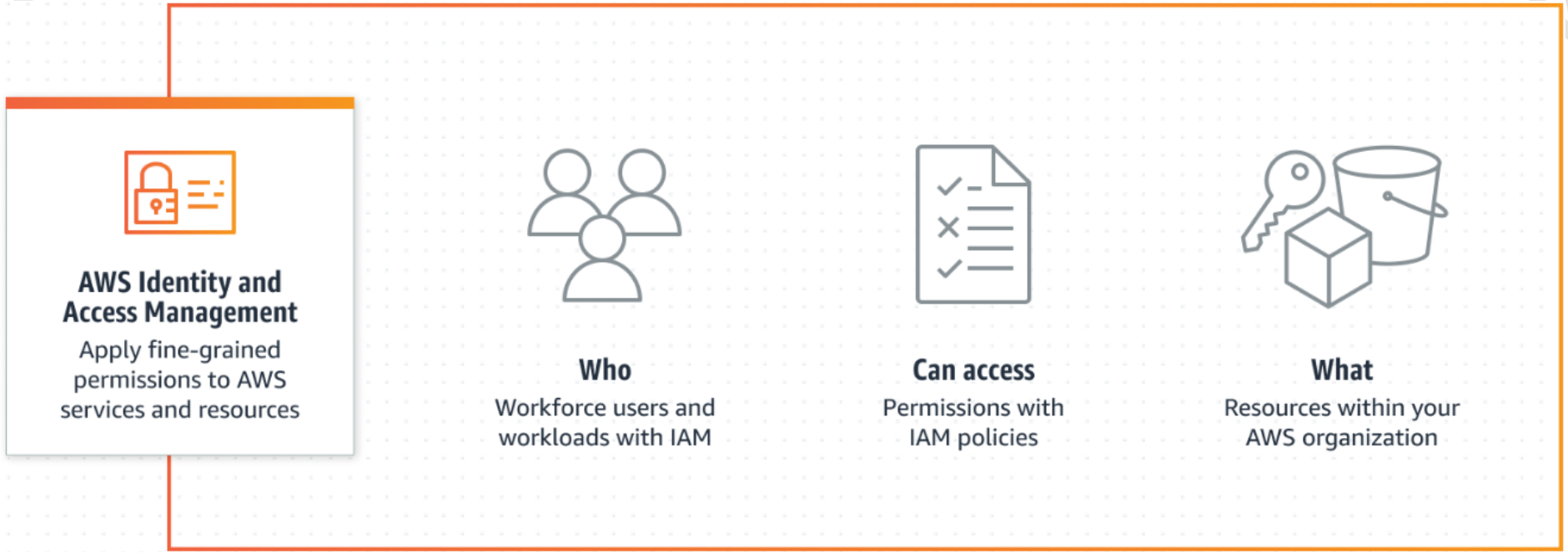
Target

- Hiểu được IAM là gì và vai trò của IAM trong quá trình thiết kế & sd hệ thống
- Nắm rõ các concept của IAM
- Thành thạo việc thiết kế và cấu hình User/Group/Role/Policy theo yêu cầu.
- Làm quen với AWS Command line interface (CLI) thông qua các ví dụ.
- Nắm được các security Best Practice liên quan IAM.

IAM

Viết tắt của **Identity and Access Management**

Nhiệm vụ định danh và phân quyền, quản lý việc ai(who) và cái gì (what) có thể access như thế nào tới các resources trên AWS, quản lý một cách tập trung các quyền chi tiết, phân tích truy cập để tinh chỉnh quyền.



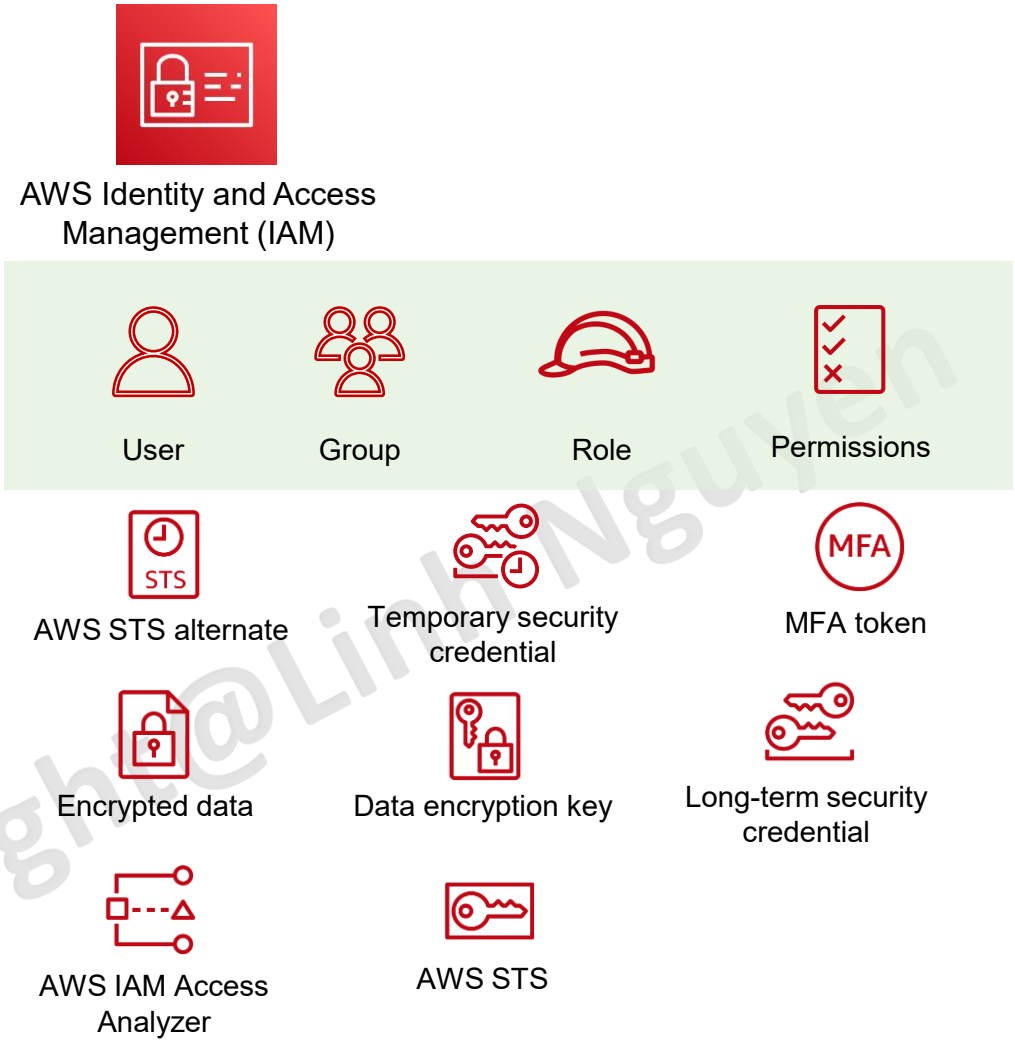
IAM usecase

- Áp dụng quyền chi tiết và mở rộng quy mô với khả năng kiểm soát truy cập dựa trên thuộc tính. Vd: Phòng ban, job role, tên nhóm.
- Quản lý truy cập theo từng tài khoản hoặc mở rộng quy mô truy cập trên các tài khoản và ứng dụng AWS.
- Thiết lập quy tắc bảo vệ & phòng ngừa cho toàn tổ chức
- Thiết lập, xác minh và điều chỉnh quy mô quyền đối với đặc quyền tối thiểu thông qua việc thiết lập, xác minh, tuỳ chỉnh.

IAM Concept

Để có thể thiết kế & xây dựng hệ thống trên AWS đảm bảo tiêu chí về Security cũng như không gặp trouble, chúng ta cần nắm vững các concept cơ bản của IAM bao gồm:

- User
- Group
- Role
- Permission (Policy)



IAM – Policy

Permissions & Policy

Quy định việc ai/cái gì có thể hoặc không thể làm gì.

Một policy thường bao gồm nhiều Statement quy định Allow/Deny hành động trên resource dựa trên condition.

Mỗi statement cần định nghĩa các thông tin:

- **Effect**: có 2 loại là Allow & Deny. *Deny được ưu tiên hơn.
- **Action**: tập hợp các action cho phép thực thi.
- **Resource**: tập hợp các resource cho phép tương tác.
- [**Condition**]: Điều kiện kèm theo để apply statement này.

Policy có thể gắn vào Role/Group/User.

IAM – Policy

Permissions & Policy

Policy có 2 loại là: Inline Policy và Managed Policy

- Inline policy: được đính trực tiếp lên Role/User/Group và không thể tái sử dụng ở Role/User/Group khác.
- Managed Policy: Được tạo riêng và có thể gắn vào nhiều User/Group/Role.

Managed Policy lại được chia thành 2 loại là AWS Managed và User Managed.

Việc lựa chọn giữa Inline vs Managed phải được tính toán dựa trên các yếu tố như: tính tái sử dụng, quản lý thay đổi tập trung, versioning & rollback.

IAM – Policy

Sample of an IAM Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StartInstance",
                "ec2:StopInstance"
            ],
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Environment": "Dev"
                }
            }
        }
    ]
}
```

Permissions & Policy

*Policy này quy định đối tượng được gán policy này được phép thực hiện 2 hành động là StartInstance và StopInstance trên toàn bộ các EC2 instance với điều kiện instance đó có 1 thẻ tag tên Environment và giá trị = Dev.

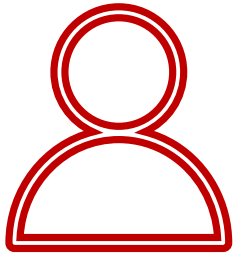
IAM – User

Đại diện cho 1 profile của 1 người dùng trên AWS account.

User có thể login vào AWS Console sử dụng username/password.

User mặc định khi tạo ra sẽ không có quyền gì. Cần cấp quyền cho user thông qua Policy hoặc Group (slide sau).

User có thể phát hành access-key/secret-key để sử dụng cho CLI hoặc test SDK trong quá trình test code. Cặp access/secret key này cũng sẽ đại diện cho user (thay vì dùng username/password).



User

IAM – Role

Role

Đại diện cho 1 quyền trên AWS. Không giống như khái niệm Role của 1 user như trong phân quyền hệ thống, cần lưu ý để tránh nhầm lẫn.

Sử dụng khi muốn cấp quyền cho 1 thực thể có thể tương tác với các resources khác trên AWS. Thường dùng để gắn vào EC2, Lambda, Container,...

Có thể sử dụng để cấp quyền cho 1 user nằm ở AWS account khác mà không muốn tạo ra user profile cho họ trên account AWS của mình. Bằng cách cho phép 1 user assume-role tới 1 role trên account, user có thể có các quyền tạm thời để thực hiện thao tác.

*Lưu ý: một resource trên AWS không thể tương tác với resource khác nếu không được gán Role với các quyền thích hợp. Đây cũng chính là lý do khiến cho việc Role & Permission khiến cho mọi người tốn thời gian trouble shooting nếu không nắm rõ dịch vụ mà mình đang sử dụng.

IAM – Group

Đại diện cho 1 nhóm user trên hệ thống.

Sử dụng khi muốn phân chia quyền dựa theo vai trò trong dự án, phòng ban,...

Nên thiết kế các nhóm user và phân quyền hợp lý, sau đó khi có người mới chúng ta chỉ cần add user đó vào các nhóm cần thiết giúp tiết kiệm thời gian và tránh sai sót (cấp dư hoặc thiếu quyền).

Lưu ý tránh bị chồng chéo quyền (vd 1 group allow action A nhưng group khác lại deny action A).

Một group không thể chứa group khác (lồng nhau).

Một user có thể không thuộc group nào hoặc thuộc nhiều groups.

Một group có thể không có user nào hoặc có nhiều users.

Lab1: IAM User, Group, Policy

Login to AWS console, thực hiện nội dung sau:

1. Tạo 1 Group “developer-group” có quyền **AdministratorAccess** (managed policy)
2. Create 1 user “developer-01” Add 1 user vào “developer-group”
3. Login vào console, thử thực hiện 1 vài thao tác vd Launch Instance, create S3 bucket, upload, download.
4. Tạo thêm group “**tester-group**” có quyền “**Readonly Access**” (managed policy)
5. Thêm custom policy “**deny-delete-object-s3**” vào group “**tester-group**”
6. Add user “developer-01” vào “tester-group”
7. Thử dùng “developer-01” xóa 1 object trên s3 => expect deny.

IAM Policy vs Resource Policy

Một số resouce như S3, SQS, KMS hỗ trợ định nghĩa policy ở cấp độ resource.

Về cơ bản cấu trúc resource policy tương tự IAM policy nhưng được gán cho một resource cụ thể.

Quyền của một user (group/role) đối với resource sẽ là kết hợp của IAM Policy & Resource Policy sau khi đã loại trừ Deny.

Một số resource cần security cao sẽ thường được ưu tiên setting resource policy.

Lưu ý: IAM Policy nói chung không có tác dụng đối với account root.

=>Nếu lỡ tay setting deny all không thao tác được trên một resource, có thể login = account root để chỉnh lại.

Lab2: Resource policy for S3

Login to AWS console, thực hiện nội dung sau:
*Sử dụng lại user đã tạo ở lab1

1. Gỡ user developer-01 ra khỏi group tester (mục đích nhằm gỡ rule deny).
2. Thử thao tác lại trên s3 ->OK
3. Tạo S3 bucket policy Deny toàn bộ thao tác đối với bucket S3, principle là user developer-01.
4. Thử upload, download file lên bucket -> Deny

Lab3: AWS CLI, MFA with CLI

Login to AWS console, thực hiện nội dung sau:

*Yêu cầu máy cài sẵn AWS CLI latest.

1. Phát hành access key/secret key cho user đã tạo ở Lab1
2. Thiết lập access key/ secret key cho AWS CLI.
3. Tương tác với 1 vài service bằng CLI.
4. Tạo policy enforce-mfa-policy và gắn vào “developer-group”
5. Thử gõ lại CLI xem còn xài dc ko?
6. Cài MFA cho user “developer-01”
7. Thiết lập sesssion_token sử dụng MFA code.
8. Gõ lệnh CLI sử dụng profile đã có MFA.

Lab4: IAM Role for EC2

Login to AWS console, thực hiện nội dung sau:

1. Tạo 1 Role cho phép access full tới một S3 bucket chỉ định.
2. Tạo 1 ec2, gán role vừa tạo.
3. Login vào EC2.
4. Thực hiện các action uload, download, delete file dùng CLI (AMZ Linux 2 có cài sẵn AWS CLI)

IAM – Assume Role

Yêu cầu AWS STS cung cấp 1 set **temporary security credential** để có thể access resource mà thông thường ta không có quyền access.

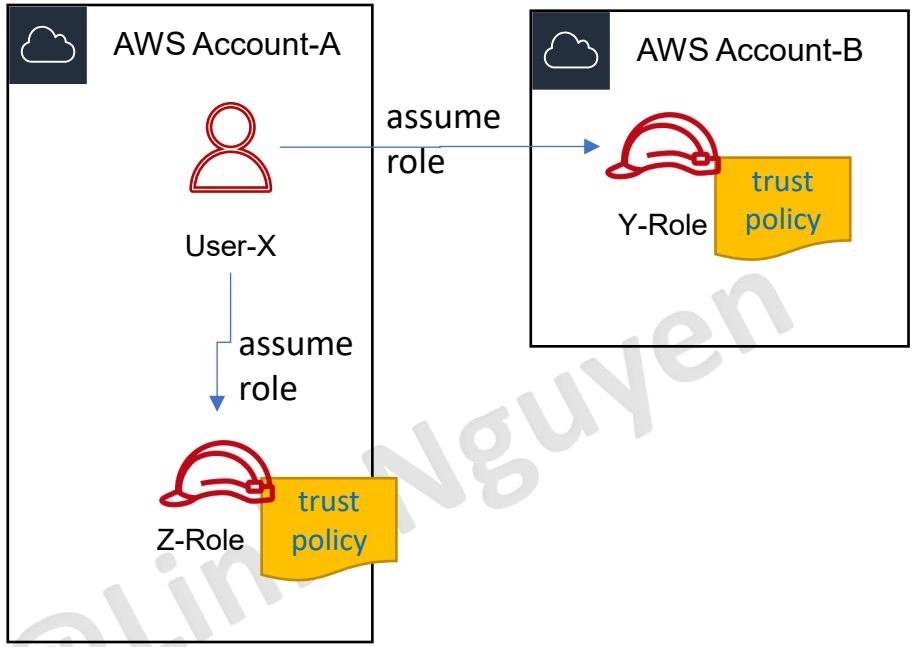
Có thể assume tới 1 role trên cùng account hoặc khác account.

Sử dụng trong các usecase cấp quyền tạm thời mà không muốn tạo nhiều user, quản lý tập trung user thông qua Single-Sign-On thay vì tạo user trên từng account.

Điều kiện:

- **Trust policy** ở role cho phép entity thực hiện sts:AssumeRole.
- **Permission policy** ở entity cho phép gọi sts:AssumeRole

*Bản chất AssumeRole là một API Action, có thể thực hiện bởi một User, Role hoặc một Service để nhận credentials tạm thời được cấp phát bởi AWS STS.

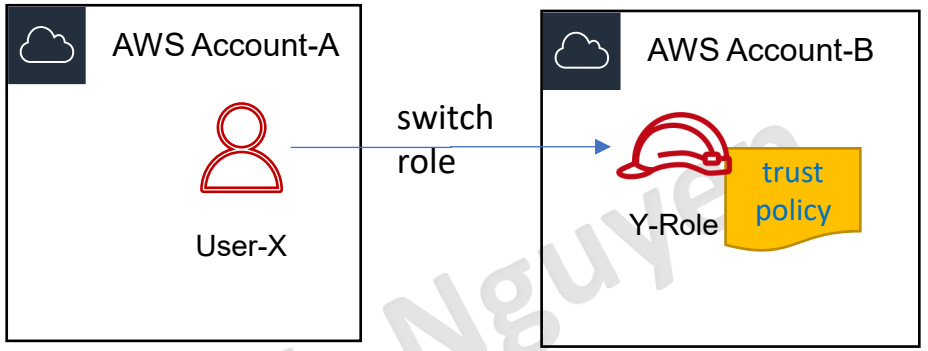


IAM – Switch Role

Switch role là một tính năng giao diện người dùng trên AWS Console (thực hiện Assume Role trên GUI).

Cho phép người dùng IAM chuyển sang role khác bằng cách nhập thông tin role (Account ID, Role name).

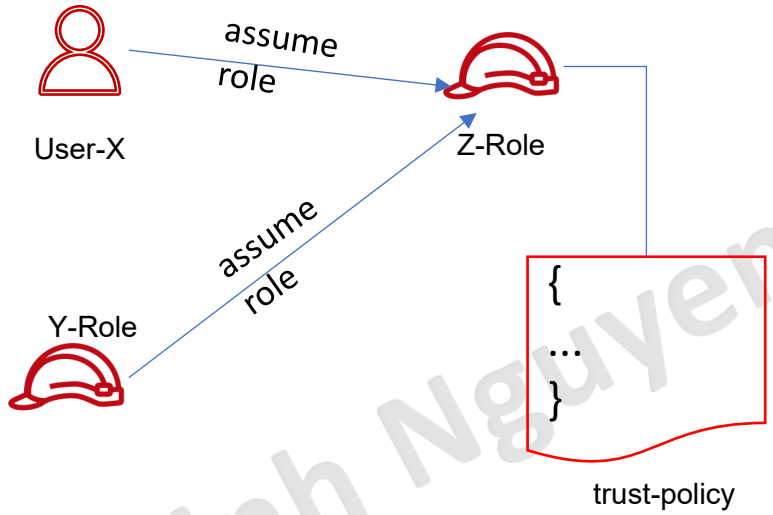
Sau khi switch role, user sẽ rời bỏ quyền ban đầu của mình và sẽ tương tác với các resource khác với tư cách là role Y.



IAM – Trust policy

Quy định đối tượng nào có thể assume sang role được gắn trust policy này.

Đối tượng được trust có thể là User/Role/AWS service.



**Trong rất nhiều trường hợp các dịch vụ của AWS cần được seting trust policy một cách tường minh để có thể assume sang một role cần thiết. Chi tiết khi đến các bài thực hành liên quan tới dịch vụ mình sẽ trình bày.*

IAM – Trust policy example

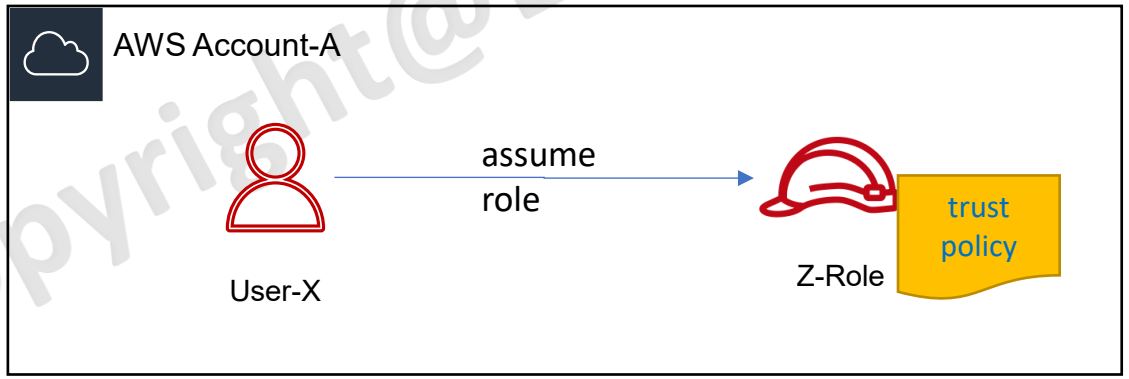
```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": [
               "arn:aws:iam::111122223333:user/PauloSantos"
            ]
         },
         "Action": "sts:AssumeRole",
         "Condition": {
            "BoolIfExists": {
               "aws:MultiFactorAuthPresent": "true"
            },
            "IpAddress": {
               "aws:SourceIp": "203.0.113.0/24"
            },
            "DateGreaterThan": { "aws:CurrentTime": "2023-04-23T12:00:00Z" },
            "DateLessThan": { "aws:CurrentTime": "2023-04-23T23:59:00Z"  }
         }
      }
   ]
}
```

Trust policy này cho phép user PauloSantos có thể assume role được gắn policy này trong điều kiện MFA bật, kết nối từ dải IP 203.0.113.0/24 và quy định thời gian từ 2023/04/23 12:00 – 2023/04/23 23:59

Lab 5: Assume Role sd AWS CLI

Login to AWS console, thực hiện nội dung sau:

1. Gỡ toàn bộ quyền/policy của user đã được tạo ở lab1. Remove user ra khỏi group.
2. Thử thực hiện 1 action (vd **list s3**) sử dụng access/secret key của user => bị deny không thể access.
3. Tạo 1 role tên **MyPowerRole**, gán policy **PowerUserAccess**, thiết lập trust policy cho phép user ở trên có thể assume role.
4. Assume role sang Role vừa tạo ở step 3 (Sử dụng CLI).
5. Sử dụng access/secret key lấy được trong quá trình assume role để thao tác với resource (vd list s3, download, upload file) => có thể access.

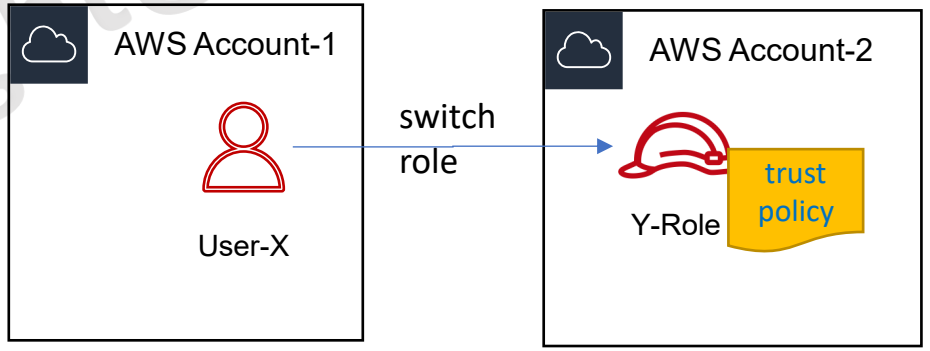


Lab 6: Switch role khác account trên AWS console

LƯU Ý: bài lab này chỉ dành cho các bạn có 2 account AWS. Các bạn không có có thể xem để tham khảo nếu tương lai có dự án nào cần dùng.

*Sử dụng lại user của bài lab trước (developer-01).

1. Sử dụng một account aws thứ hai, tạo một IAM Role có tên **MyReadOnlyRole** và assign permission ReadOnlyAccess
2. Setting trust policy cho phép user của account thứ nhất trong bài lab trên có thể switch role qua.
3. Quay trở lại account thứ nhất, tạo một custom policy: **my-custom-policy-for-switchrole,** gán quyền **sts:AssumeRole** cho user developer-01.
4. Thử switch role sang account thứ hai (điền thông tin account ID, Role name)
5. Thử truy cập một số ressource (EC2, S3) xem có thể access được không.
6. Switch back trở lại account thứ nhất.



Best practices for IAM

- Sử dụng group để gán quyền cho các user.
- Cấp quyền vừa đủ cho user, không cấp quá dư thừa.
- Chú ý các deny rule khi thiết kế policy để tránh bị conflict permission giữa các group (group này allow nhưng group kia deny).
- Luôn force MFA cho các IAM user (không bật MFA sẽ không xài dc console). *Sẽ hơi phiền nếu dùng CLI kết hợp MFA nhưng nếu làm quen sẽ không mất nhiều thời gian.
- Sử dụng Role khi muốn cấp quyền cho server/ứng dụng. Access/Secret key chỉ dùng để sử dụng khi gõ CLI trên local hoặc test các app đang trong quá trình phát triển.
- Tuyệt đối KHÔNG share User hoặc Access/Secret key giữa các member trong dự án.

Lab: Clear resources

Login to AWS console, thực hiện nội dung sau:

1. Xoá toàn bộ các user không sử dụng đến.
2. Có thể giữ lại 1 user duy nhất dùng để sử dụng thay cho account root.

Bonus section

Hướng dẫn các bạn cách enable tính năng cho phép IAM user cũng có thể xem được thông tin billing.

*Xem thao tác trên AWS console.