

# 5. IAM, ACCOUNTS AND AWS ORGANISATIONS

⌚ Created	@August 22, 2022 6:10 PM
▼ Class	
▼ Type	
📎 Materials	
✓ Reviewed	<input type="checkbox"/>
☰ Property	
📅 Date	

## IAM Policies

default deny policy will work

it will always take deny first and then go for allow

for example at the below code deny policies will take effect at first and then other



lets for example user is in 2 policy and also he is member of group  
 collect all of the statements and evaluate them and applied all together but there is explicit deny then its deny or explicit allow then allow

deny                        allow                        deny

if none of the apply then its default deny

## 1. Inline policy type

where you can separate assign policy using JSON but not the best practice

incase of exception we use inline

for ex blocking of access to that particular user

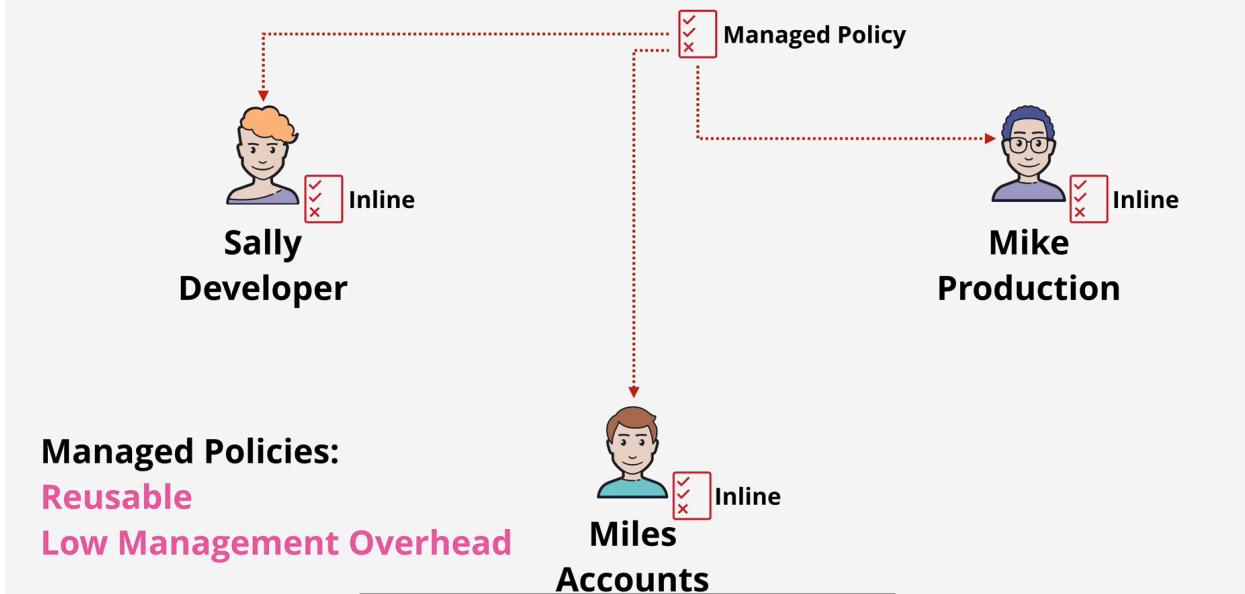
## 2. Managed policy



## IAM Policies

<https://learn.cantrill.io>

adriancantrill



1. [AWS managed policy](#)
2. Customer managed policy

Demo on IAM Users where we created IAM user using cloudFormation template and then explicitly allowed and denied some of the policies using upload template for sally we tested this by login into sally and username(inside IAMadmin) then tested everything then we tested inline policy by explicitly denying some s3 access and then we deleted the cloudformation stack

**to delete stack make sure to delete the objects from s3 bucket first**

IAM Groups:

**It is not user its container so you cannot login to groups**

its used to manage iam users



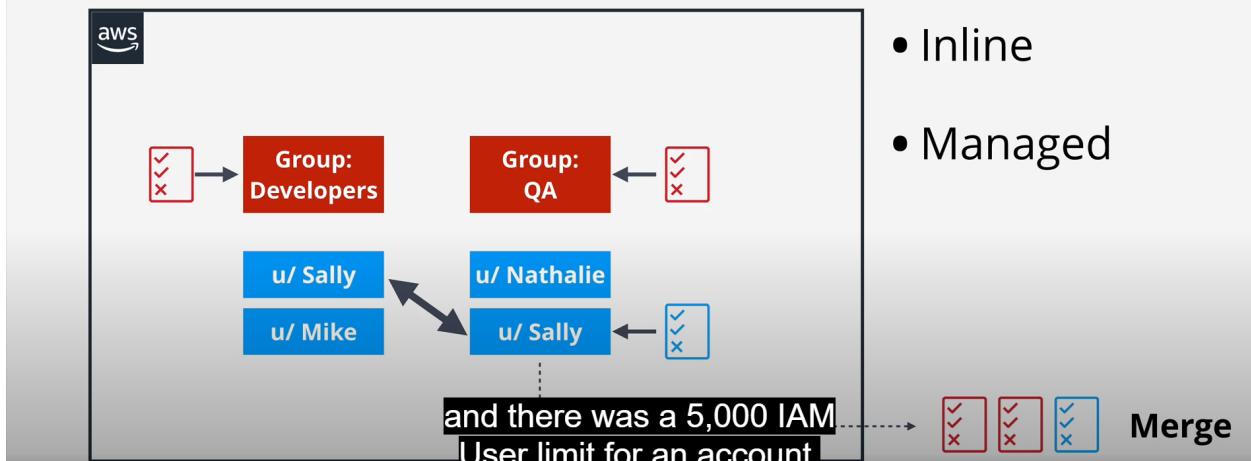
## IAM Groups

<https://learn.cantrill.io>

@adriancantrill



# ... IAM Groups are containers for Users



## ▼ IAM user restriction:

Can be member of 10 groups

and 5000 IAM user limit for a single account



**Groups are **not** a true identity.  
They can't be referenced as a  
principal in a policy**

**but a resource policy cannot**

**grant access to an IAM Group,**

IAM Groups are a feature of IAM which you need to know about for the exam.

They are an admin or container feature.

You can add IAM users to groups, and add permissions to Groups

Groups are NOT real identities ... can't be used from resource policies and have no credentials to login with.

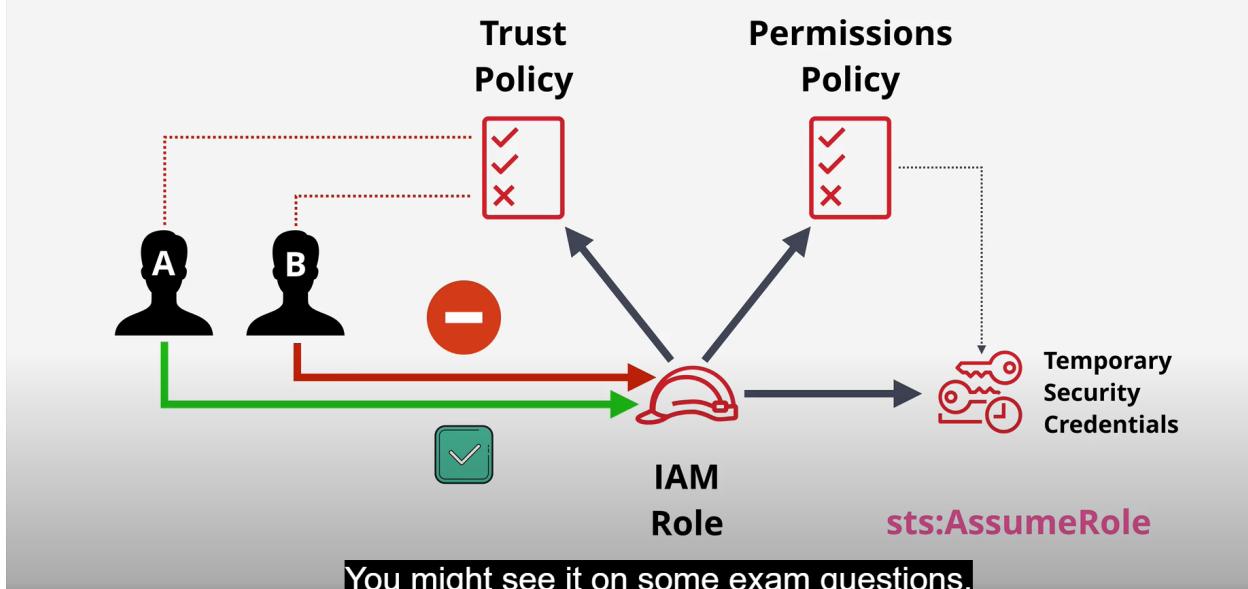
## IAM Roles

**Principle: Is the physical person, identity, process or application which wants to authenticate with AWS**



## IAM ROLES

<https://learn.cantrill.io> adriancan



When to use IAM Roles:

To prevent from hardcoding roles and authentication information into any function such as for example in lambda function we use IAM Roles which gives temporary access to the function using `sts:AssumeRole` and once the execution is done access can be revoked.

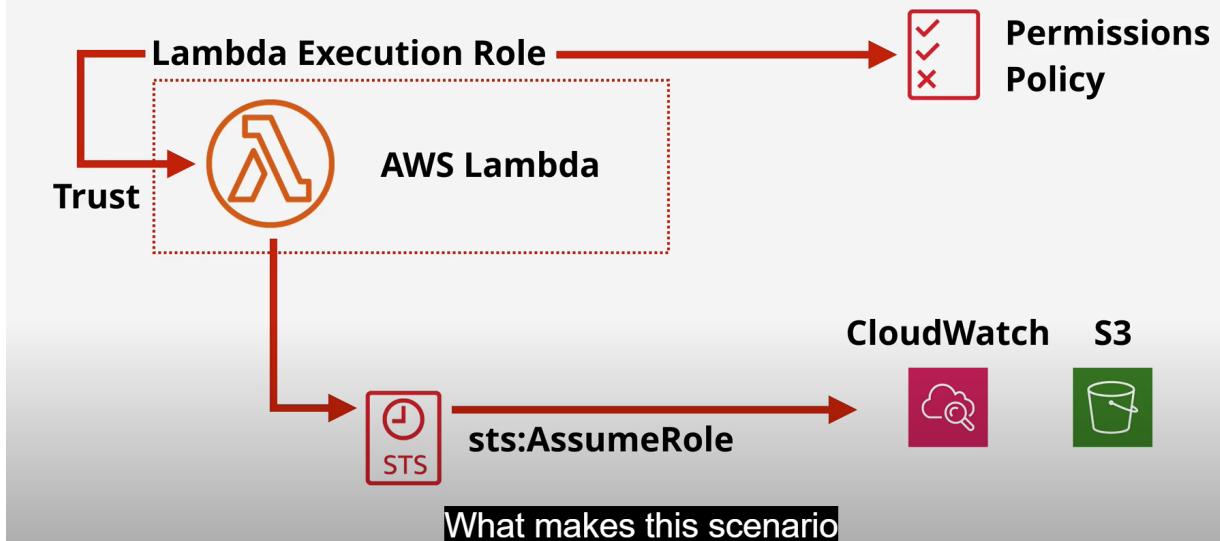
also we dont know how many functions will be running on runtime so its better to have IAM roles assigned



## When to use IAM Roles

<https://learn.cantrill.io>

adriancantrill



Another use of IAM Roles is when on-premises for example Active directory allows them to login

external accounts or entities cannot be used to access aws account



## When to use IAM Roles

<https://learn.cantrill.io>

adriancantrill



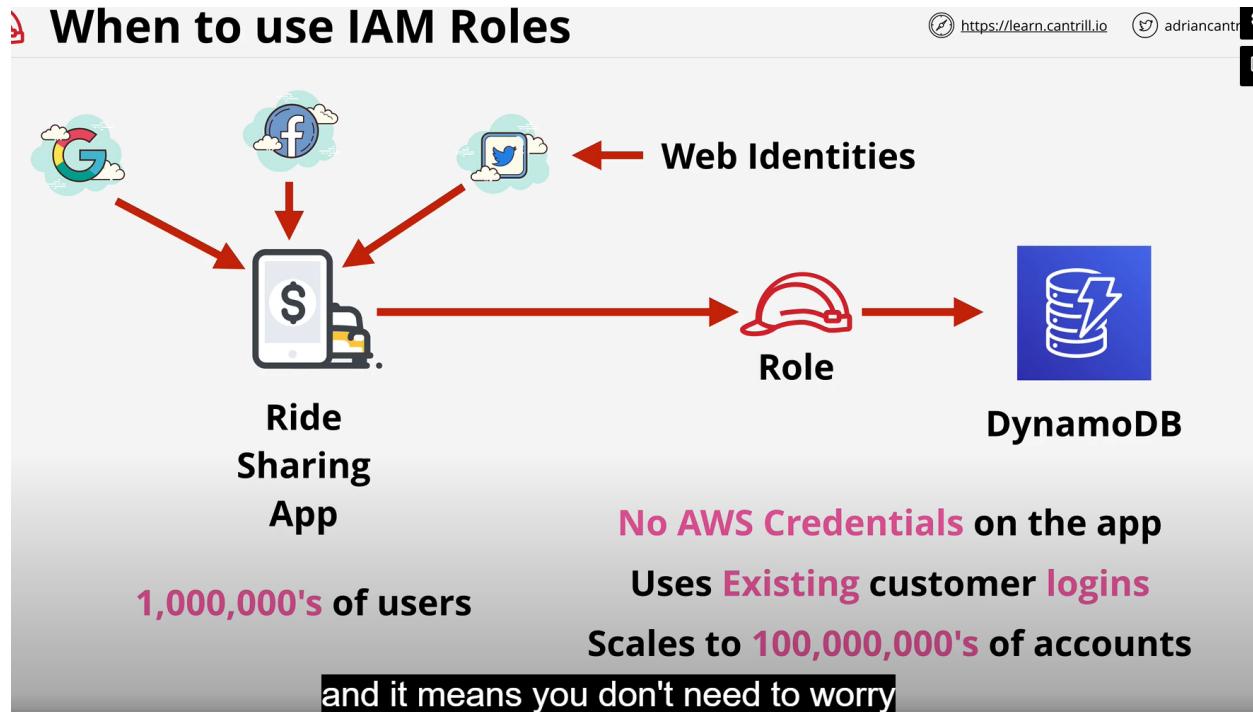
Single Sign-on or  
> 5000 Identities

you can't directly use Facebook, Twitter.



External accounts can't be  
used in AWS directly

max 5k IAM roles



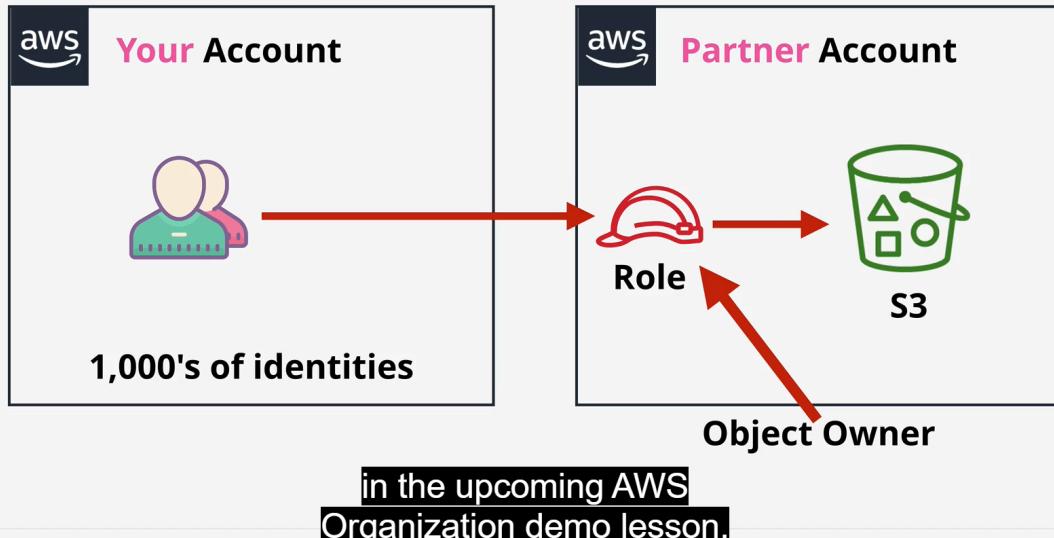
lets say my 1k user wants to access and use s3 from another account we can apply IAM roles



## When to use IAM Roles

<https://learn.cantrill.io>

@adriancantrill



Server Linked roles:



## Service-linked roles

<https://learn.cantrill.io>

@adriancantrill

- IAM role linked to a **specific AWS service**
- **Predefined** by a **service** ...
- ... providing permissions that a **service** needs to interact with **other AWS services** on your behalf
- **Service** might create/delete the role ...
- .. or allow **you** to during the **setup** or within **IAM**
- You can't delete the role until it's no longer required

And that means that it's no longer used.

Example of policy which allows service linked role:

The screenshot shows a web-based JSON editor for an IAM policy. The policy document is as follows:

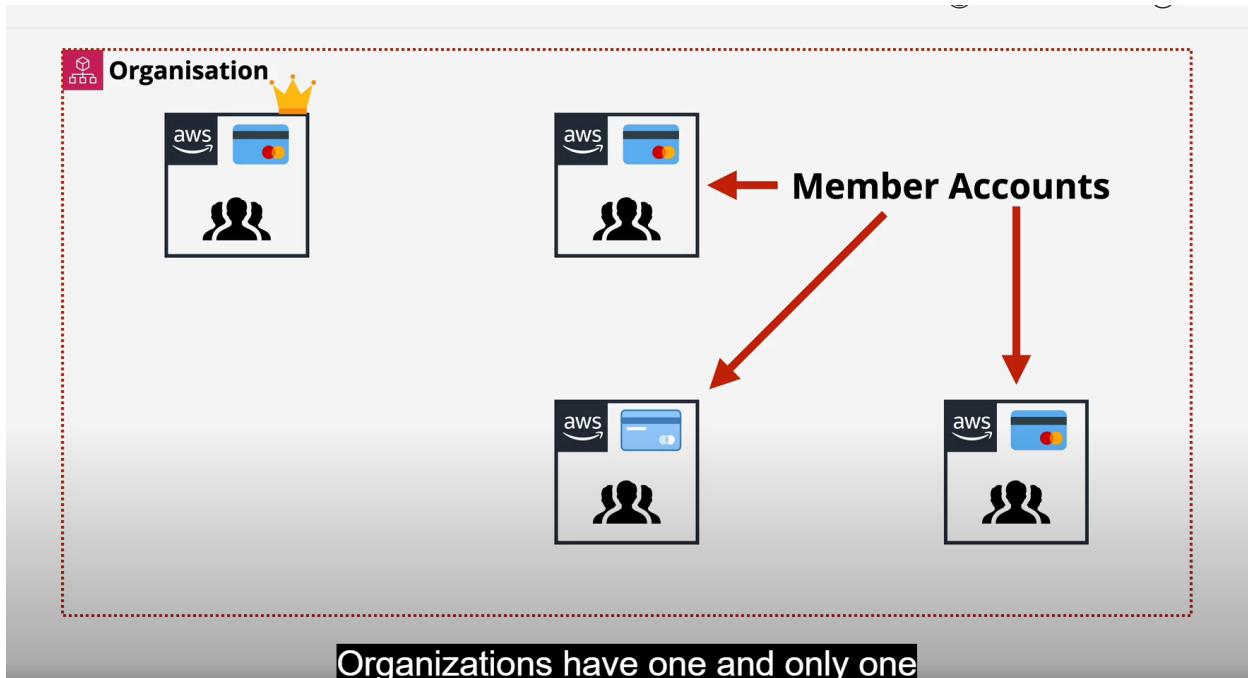
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/SERVICE-NAME.amazonaws.com/SERVICE-LINKED-ROLE-NAME-PREFIX*",
            "Condition": {"StringLike": {"iam:AWSPropertyName": "SERVICE-NAME.amazonaws.com"}}
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/SERVICE-NAME.amazonaws.com/SERVICE-LINKED-ROLE-NAME-PREFIX*"
        }
    ]
}
```

A callout box highlights the second statement's Action field, specifically the "iam:PutRolePolicy" action, with the text "which allows you to create".

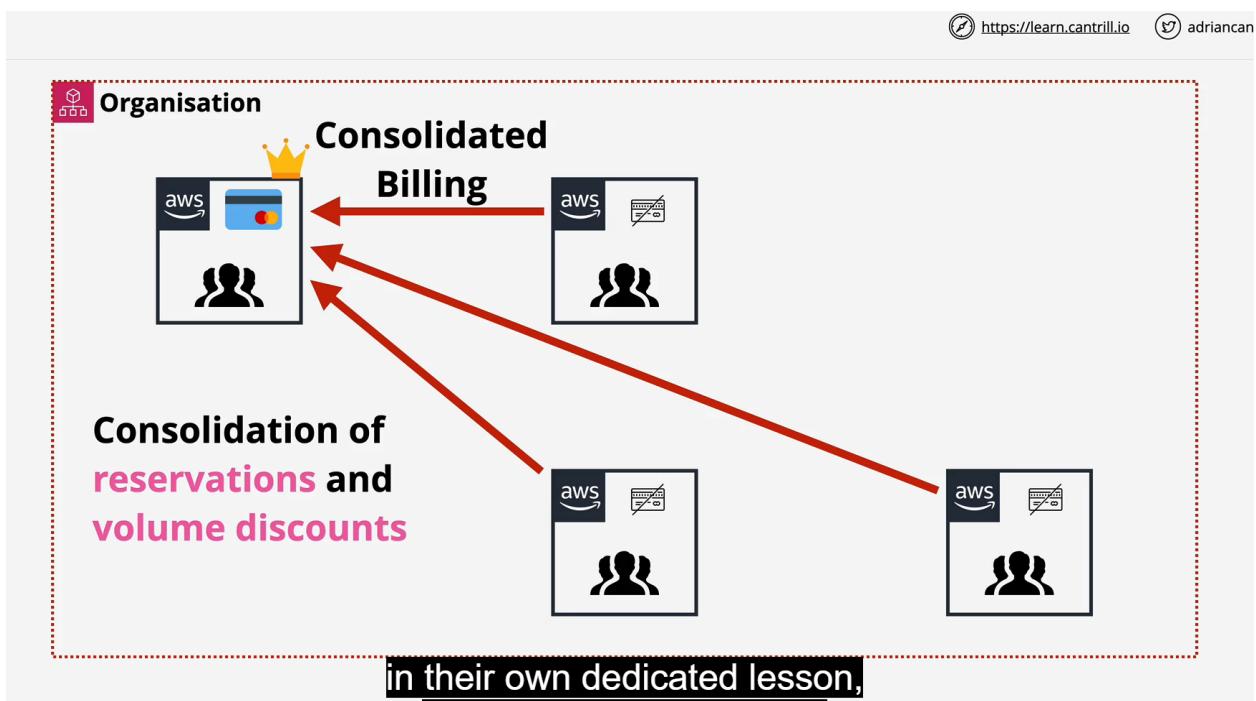
Passrole is role in aws which allows to give role separation in aws

## AWS organization:

Without aws organization we have to manage hundreds of accounts  
standard accounts are account which are not part of aws organization  
but when they accept the invite of organization account they are no longer standard account but a member accounts.



with AWS organization we get consolidated billing and get volume discount along the same



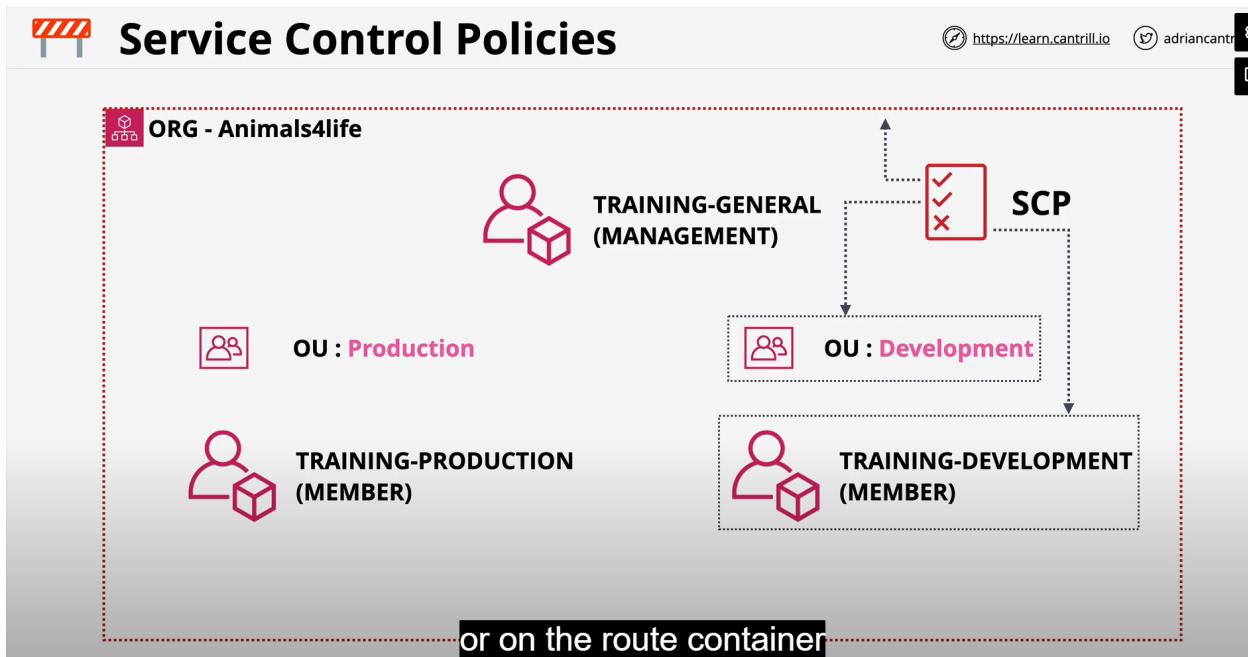
# Service Control policy

default is deny

its basically firewall with default block as aws being aws for security

Service Control policy: its basically limitor to account on what it can do

You cannot restrict management aka aws account



## Service Control Policies

 <https://learn.cantrill.io>  adriancantr

- SCPs are **account permissions boundaries**
- They limit what the account (**including account root user**) can do
- They **don't grant** any permissions

You still need to give identities

CloudWatch Logs:



## CloudWatch Logs

 <https://learn.cantrill.io>  adriancantr 

- **Public Service** - usable from AWS or on-premises
- **Store, Monitor** and **access** logging data
- **AWS Integrations** - EC2, VPC Flow Logs, Lambda, CloudTrail, R53 and more...

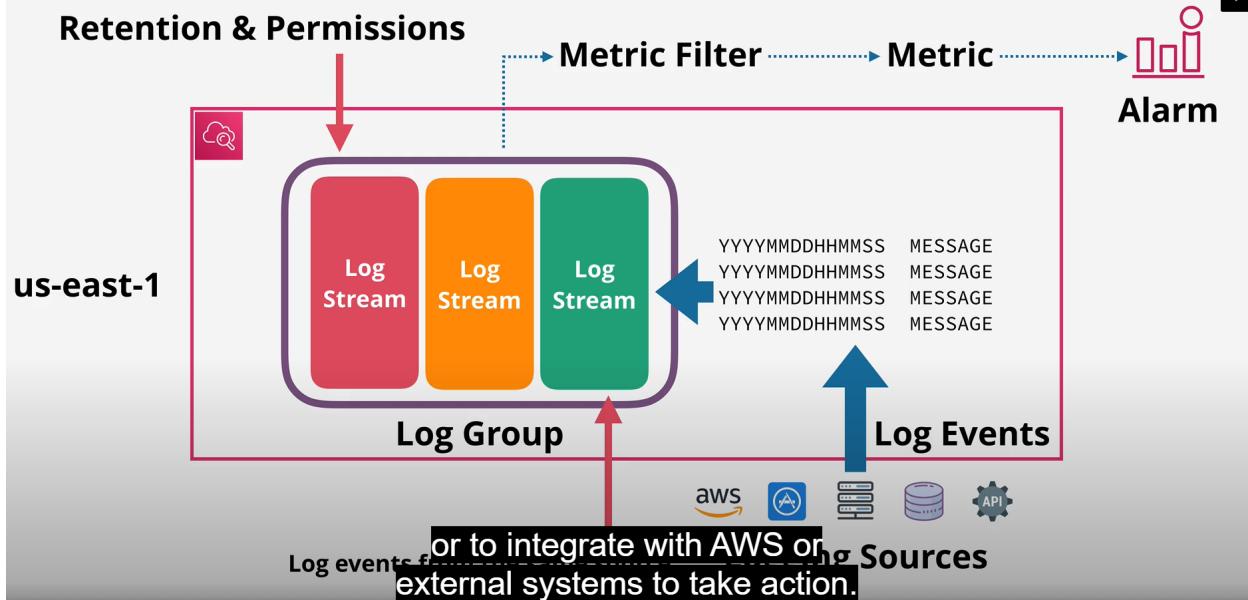
in which you can use the development kits for AWS



## CloudWatch Logs

<https://learn.cantrill.io>

adriancantrill



CloudTrail: Uses cloudwatch logs

## CloudTrail:

Uses Cloudwatch for taking logs

90 days free event logs

by default it only logs management events to reduce the log list



## CloudTrail Essentials

<https://learn.cantrill.io>

@adriancantr



- Logs API calls/activities as a **CloudTrail Event**
- 90 days stored by default in **Event History**
- Enabled **by default** - no cost for 90 day history
- To customise the service .. create 1 or more **Trails**
- **Management Events** and **Data Events**  
or when a Lambda function

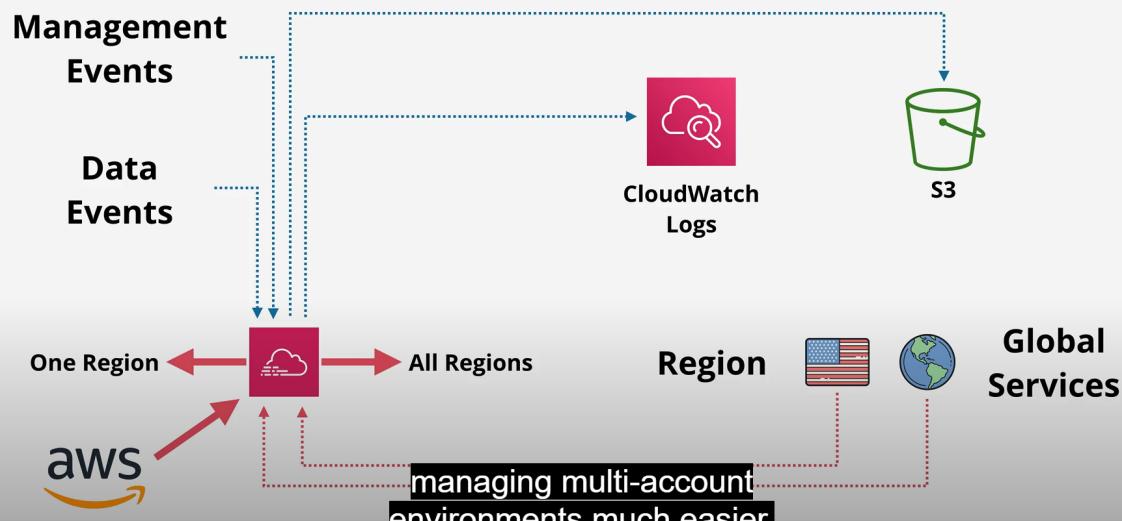
cloudtrail is regional service to get logs of global region it has to be enabled first



## CloudTrail Essentials

<https://learn.cantrill.io>

@adriancantr





- Enabled by default ... but **90** days .. **no s3**
- **Trails** are how you configure S3 and CWLogs
- Management events **only** by default
- **IAM, STS, CloudFront** => Global Service Events
- **NOT REALTIME** - There is a delay

CloudTrail typically delivers log files

IAM STS n Cloudfront stores data on global services and to use them they have to be enabled first

## Not real-time **15-20 min**

also stores data on s3 bucket in json format

## AWS Control Tower:



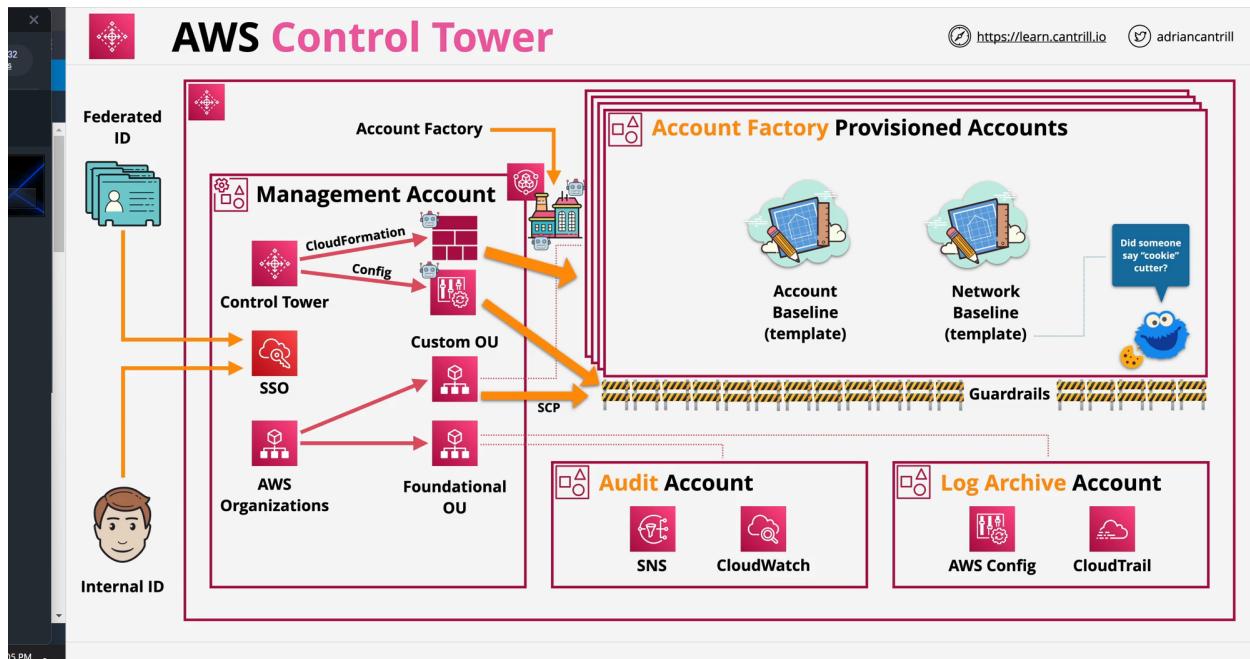
## AWS Control Tower

<https://learn.cantrill.io>

[adriancantrill](#)

- Quick and Easy setup of multi-account environment
- Orchestrates other AWS services to provide this functionality.
- Organizations, IAM Identity Center, CloudFormation, Config and more....
- Landing Zone - multi-account environment
  - ... SSO/ID Federation, Centralised Logging & Auditing
- Guard Rails - Detect/Mandate rules/standards across all accounts
- Account Factory - Automates and Standardises new account creation.
- Dashboard - single page oversight of the entire environment

better version of amazon organization





## AWS Control Tower - Landing Zone

<https://learn.cantrill.io>

@adriancantrill



- Well Architected multi-account environment - Home Region
- .. built with AWS Organizations, AWS Config, CloudFormation
- Security OU - Log Archive & Audit Accounts (CloudTrail & Config Logs)
- Sandbox OU - Test/less rigid security
- You can create other OU's and Accounts
- IAM Identity Center (AWS SSO) - SSO, multiple-accounts, ID Federation
- Monitoring and Notifications - CloudWatch and SNS



## AWS Control Tower - Guard Rails

<https://learn.cantrill.io>

@adriancantrill



- Guardrails are rules - multi-account governance
- Mandatory, Strongly Recommended or Elective
- Preventive - Stop you doing things (AWS ORG SCP)
  - ... enforced or not enabled
  - ... i.e allow or deny regions or disallow bucket policy changes
- Detective - compliance checks (AWS CONFIG Rules)
- .. clear, in violation or not enabled
  - .. detect CloudTrail enabled or EC2 Public IPv4



- **Automated Account Provisioning**
  - .. cloud **admins** or **end users** (with appropriate permissions)
- **Guardrails - automatically** added
- Account **admin** given to a **named user** (IAM Identity Center)
- Account & network **standard configuration**
- Accounts can be **closed** or **repurposed**
- Can be fully **integrated** with a businesses **SDLC**