Class 24

4, Policies

Policies are created to restrict or provide permissions to users/apps

If we go to IAM -> Policies -> “Filter policies”, we can see types of policies, mainly

* Customer managed

These are the policies which we create

* AWS managed

By default, these policies are given by “AWS”

**Lab:**

Customer managed: to create policy, we need to write “**JSON**” program

**JSON** - java script object notation (is used to exchange data)

What we use to create **JSON** program

1, **{}**

2, “ ”

3, **:**

4, **,**

5, **[]**

1, IAM -> Policies -> Create policy -> JSON

{

"Version": "2012-10-17",

"Statement": []

}

* Ex: Json program opens with

{

“key”: “value”,

“key”: “value”,

“key”: [“list”, “list”, “list”]

“key”: [

{

“value as key”: [

“list”,

“list”,

“list”,

]

},

“List”,

“List”,

“List”,

]

}

Eg: Json program of list of clouds

{

    "clouds": [

        {

            "name": "AWS",

            "estd": "2006",

            "parent": "Amazon",

            "CEO": "Andy Jassy",

            "services": [

                {

                    "Ec2": [

                        "VPC",

                        "VPG",

                        "TGW",

                        "VM",

                        "ELB"

                    ]

                },

                "RDS",

                "IAM",

                "Nosql"

            ]

        },

        {

            "name": "Azure",

            "estd": "2010",

            "parent": "MS",

            "CEO": "Sathya nadella",

            "services": [

                {

                    "Ec2": [

                        "VPC",

                        "VPG",

                        "TGW",

                        "VM",

                        "ELB"

                    ]

                },

                "RDS",

                "IAM",

                "Nosql"

            ]

        }

    ]

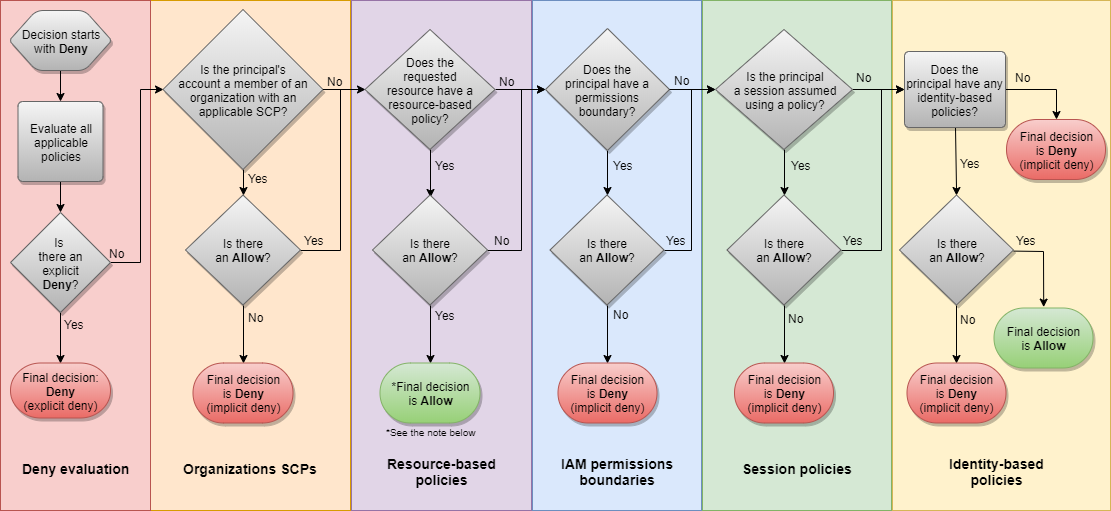
}

------------------------

🡺🡺what is policy hierarchy? Where to apply IAM policies?

By default, policy is denied. Organizations (on sub accounts), Resources (Ec2), I-DB (users, group, role)

Google: AWS IAM policy evaluation logic



**Lab**

Create 2 users and 2 Servers

i, ProdUser -> ProdServer (permissions: Start, Stop, reboot)

ii, DevUser -> DevServer (permissions: Start, Stop, reboot)

**Note**: For every resource (ec2, s3) in AWS we have ID called "ARN" (Amazon Resource Name)

Eg: IAM -> Users -> **ProdUser** -> User ARN arn:aws:iam::683850180077:user/ProdUser

S3 -> arn:aws:s3::xxxxxxxx

Ec2 doesn’t have "ARN" it has "instant ID"

🡺Create Prod and Dev users

i, IAM -> Users -> Add user -> User name: **ProdUser** -> "AWS Management Console access" -> password: "06c61A0542”

ii, IAM -> Users -> Add user -> User name: **DevUser** -> "AWS Management Console access" -> password: "06c61A0542”

iii, Ec2 -> deploy 2 instances “ProdServer & DevServer” and give “Tags -> Key: Env, Env & Value: Prod, Dev”

#!/bin/bash

yum update -y

yum install nginx -y

service nginx start

-----> Now write policy to restrict ec2 servers for IAM users

iv, IAM -> Policies -> Create policy -> JSON -> “write following program” -> save as “Produser\_Policy”

**google:** restrict “ec2 instance” to “iam user” tags

{

    "Version": "2012-10-17",

    "Statement": [

        {

            "Effect": "Allow",

            "Action": "ec2:Describe\*", -->this permission is for ec2 read only

            "Resource": "\*"

        },

        {

            "Effect": "Allow",

            "Action": [

                "ec2:StartInstances",

                "ec2:StopInstances",

                "ec2:RebootInstances"

            ],

            "Resource": [

                "arn:aws:ec2:us-east-1:683850180077:instance/\*"-->act no.,\*allec2

            ],

            "Condition": {

                "StringEquals": {

                    "ec2:ResourceTag/Env": "Prod"

                }

            }

        }

    ]

}

Attach this Produser\_Policy to **ProdUser**

i, IAM -> Users -> **ProdUser** -> Permissions -> Add permissions -> “Produser\_Policy”

-->Now login aws console with **ProdUser**, for that go to (IAM -> Dashboard -> <https://683850180077.signin.aws.amazon.com/console>) copy and open in browser

[here you can start,stop,Reboot ProdServer. Because policy is written for Env&Prod key&Value]

----------------------------------------------------

-> likewise write policy for “Devuser\_Policy” by keeping

"ec2:ResourceTag/Env": "Dev"

Attach this Devuser\_Policy to **DevUser**

i, IAM -> Users -> **DevUser** -> Permissions -> Add permissions -> “Devuser\_Policy”

-->Now login aws console with **DevUser**, for that go to (IAM -> Dashboard -> <https://683850180077.signin.aws.amazon.com/console>) copy and open in browser

[here you can start,stop,Reboot DevServer. Because policy is written for Env&Dev key&Value]

**Add inline policy:**

-->Apart from this policies we have [**Add inline policy**](https://console.aws.amazon.com/iam/home?region=us-east-1#/users/DevUser$createPolicy) in line policy, this is added on only particular user

Eg: for explicit deny or allow that policy on only that user

i, IAM -> Users -> **DevUser** -> Permissions -> Add inline policy -> save (we can’t see this policy in policies)

eg: if we give “deny” in policy, we can see that service in user

{

    "Version": "2012-10-17",

    "Statement": [

        {

            "Effect": "Allow",

            "Action": "ec2:Describe\*",

            "Resource": "\*"

        },

        {

            "Effect": "Allow",

            "Action": [

                "ec2:StartInstances",

                "ec2:StopInstances",

                "ec2:RebootInstances"

            ],

            "Resource": [

                "arn:aws:ec2:us-east-1:683850180077:instance/\*"

            ],

            "Condition": {

                "StringEquals": {

                    "ec2:ResourceTag/Env": "Dev"

                }

            }

        },

        {

            "Effect": "Deny",

            "Action": "\*", --> by this policy, we cant see that servise

            "Resource":"\*"

        }

    ]

}

-->when we stop or start without applying policy, we get error code, we can decode by

go to -> CMD: aws sts decode-authorization-message --encoded-message (past error)

we get error decoded in json format

**IAM permissions boundaries**: in policy hierarchy

IAM permissions boundaries will not give permissions, it limits permissions

Eg:

i, IAM -> Users -> **ProdUser** -> Permissions -> Add permissions -> “**AdministratorAccess**” (AWS managed policy)

* here ProdUser get all permissions, even IAM access also, so that ProdUser can make changes in IAM users, policies, passwords, because of that we should limit IAM permissions boundaries.

-->For limiting IAM user permissions to only **EC2,** go to

i, IAM -> Users -> **ProdUser** -> Permissions -> Permissions boundary -> click on (Set boundary) -> "**AmazonEC2ReadOnlyAccess**" (AWS managed)

Now, though we gave **AdministratorAccess** to **ProdUser** we are limiting to **AmazonEC2ReadOnlyAccess**

-->For limiting IAM user permissions to **EC2, S3, Route53** go to

adding two more “AWS managed policy” for S3 & Route53 is not possible, for that we need to customize policies

i, IAM -> Users -> **ProdUser** -> Permissions -> Permissions boundary -> click on (Set boundary) -> Create policy (for time being Read only policy)

{

    "Version": "2012-10-17",

    "Statement": [

        {

            "Effect": "Allow",

            "Action": "ec2:\*",

            "Resource": "\*"

        },

        {

            "Effect": "Allow",

            "Action": "S3:\*",

            "Resource": "\*"

        },

        {

            "Effect": "Allow",

            "Action": "Route53:\*",

            "Resource": "\*"

        }

    ]

}

**Policy hierarchy**:

When i login with user, these are policy hierarchy

1, policy start with default deny, that means we didn’t write any policy

2, policy goes to Organization and login with Users (or subaccounts)

3, policy checks, applied on any Resource (ec2)

4, IAM permissions boundaries, policy is to limit services to User, like ec2, S3,Route53 though User have Administrative Access