

SESSION-4:

IAM



AWS Root User

- It has full access to all AWS resources.
- It shouldn't be used for day-to-day interactions with AWS.

Sign in

Root user

Account owner that performs tasks requiring unrestricted access. [Learn more](#)

IAM user

User within an account that performs daily tasks. [Learn more](#)

Root user email address

username@example.com

Next

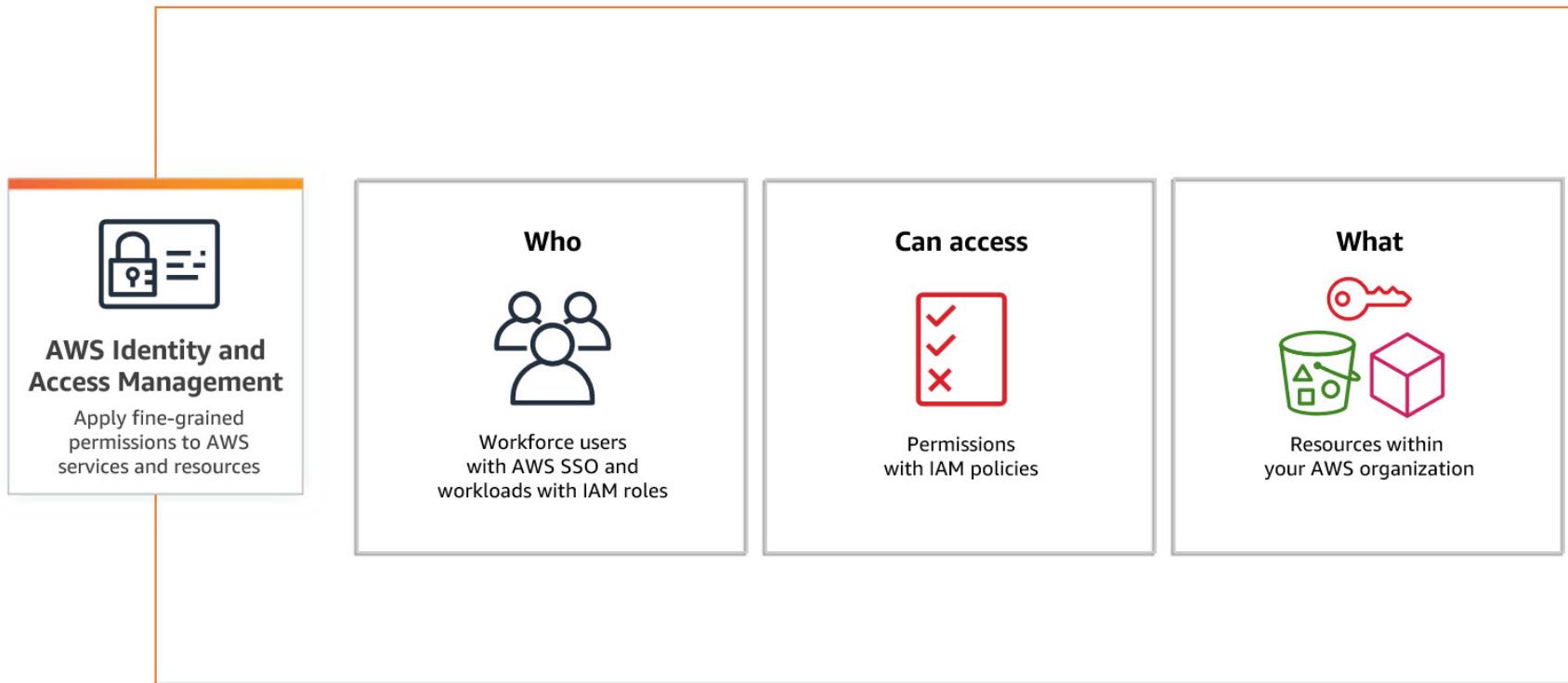
By continuing, you agree to the [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.

New to AWS?

Create a new AWS account

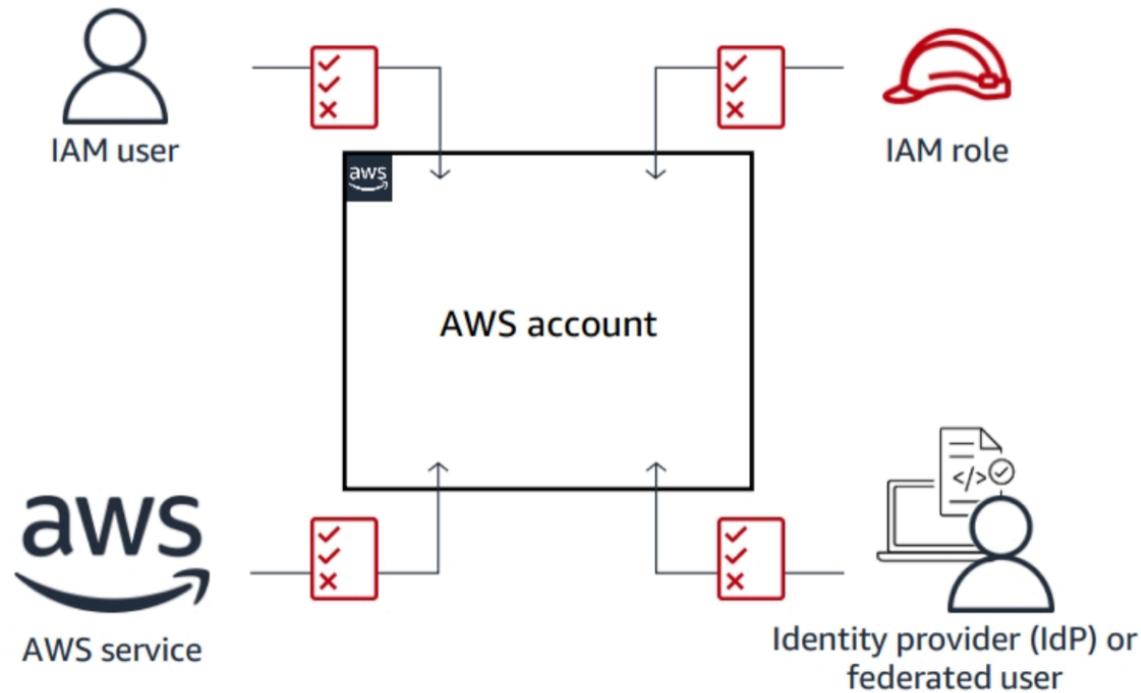
AWS Identity and Access Management (IAM)

- Create and manage users, groups and roles.
- Manage access to AWS resources and services.



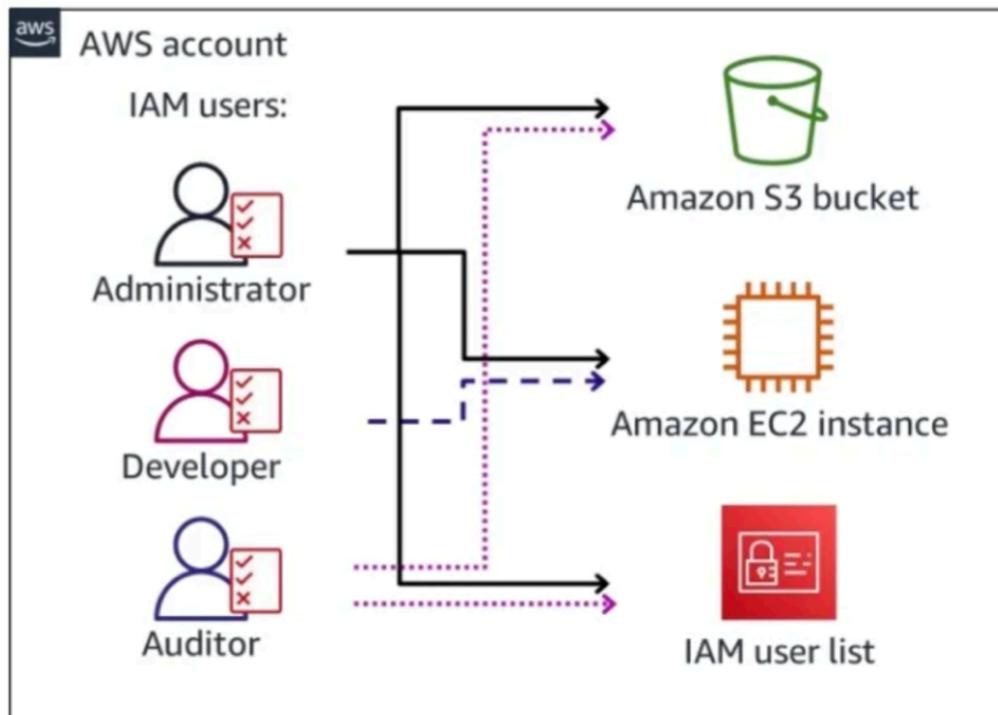
Principal

- It can make a request for an action or operation on an AWS resource.
- It can be a person, application, assumed role or federated user.



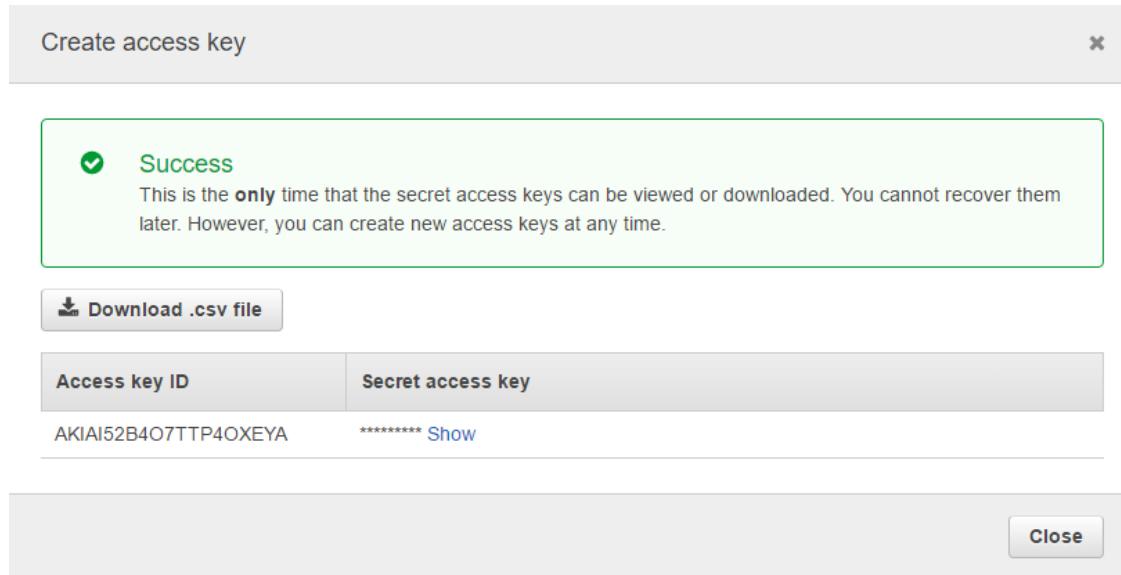
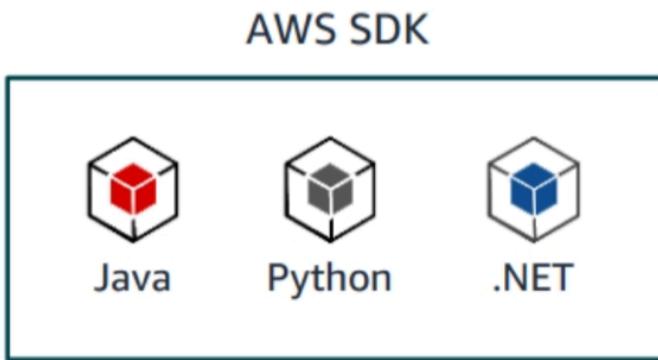
IAM Users

- They are users in the AWS account, each have different credential.
- They are able to perform specific AWS actions based on permissions.





Programmatic Access



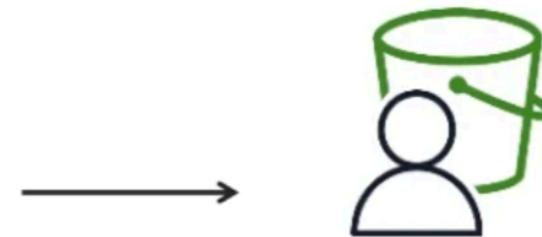
```
C:\ Command Prompt - aws configure
C:\Users\ Rushi >aws configure
AWS Access Key ID [*****QEPTE]: AKIA2C3YBHP4I5TVF
AWS Secret Access Key [*****rPkE]: TL1+V8fZ/j39g9AA8AQTG9EG1Xq/c4CKk+why
Default region name [us-east-2]: eu-north-1
Default output format [None]:
```

Setting Permissions with IAM Policies



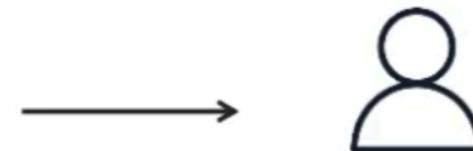
IAM policy

Select	Policy name
	AdministratorAccess
	AmazonEC2ReadOnlyAccess
✓	AmazonS3FullAccess
	AmazonS3ReadOnlyAccess



Amazon S3 administrator

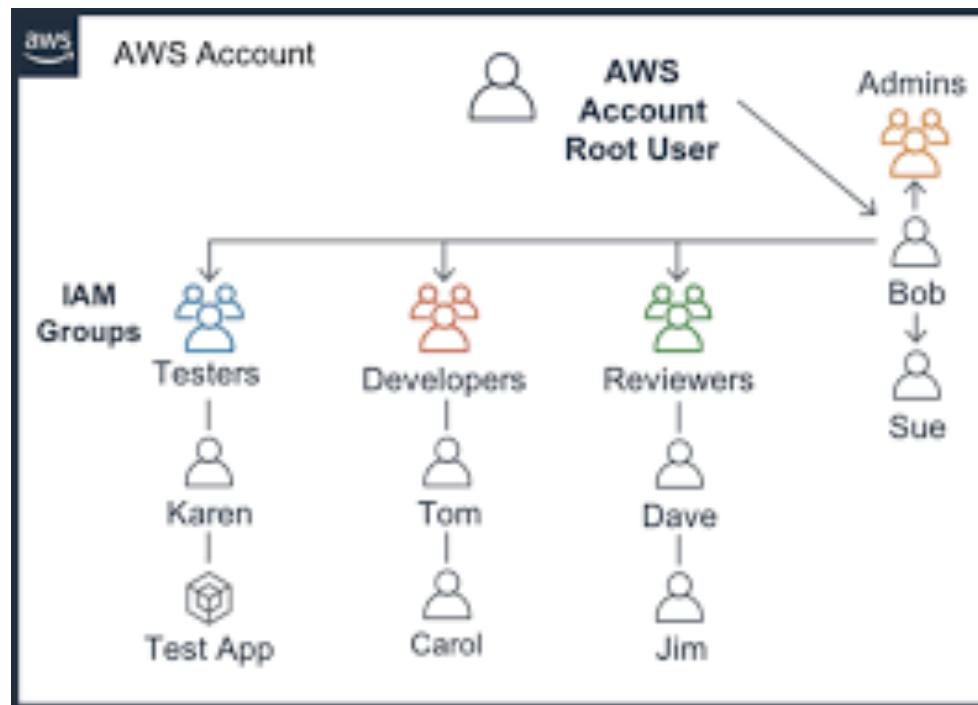
Select	Policy name
	AdministratorAccess
✓	AmazonEC2ReadOnlyAccess
	AmazonS3FullAccess
✓	AmazonS3ReadOnlyAccess



Auditor

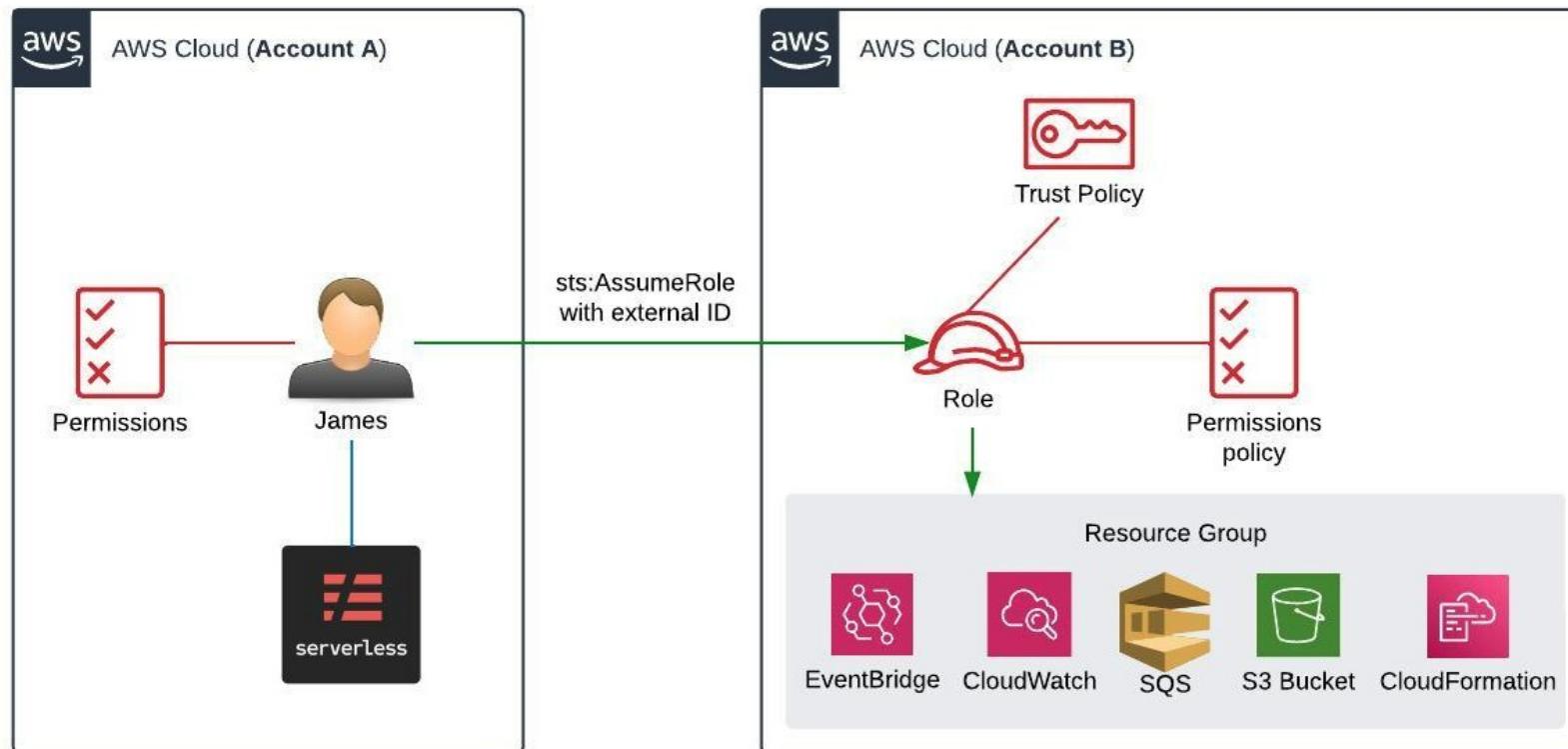
IAM User Groups

- IAM Users are assigned to IAM User Groups.
- Policies can be attached to IAM User Groups so that it can be applied to all IAM Users within the group.

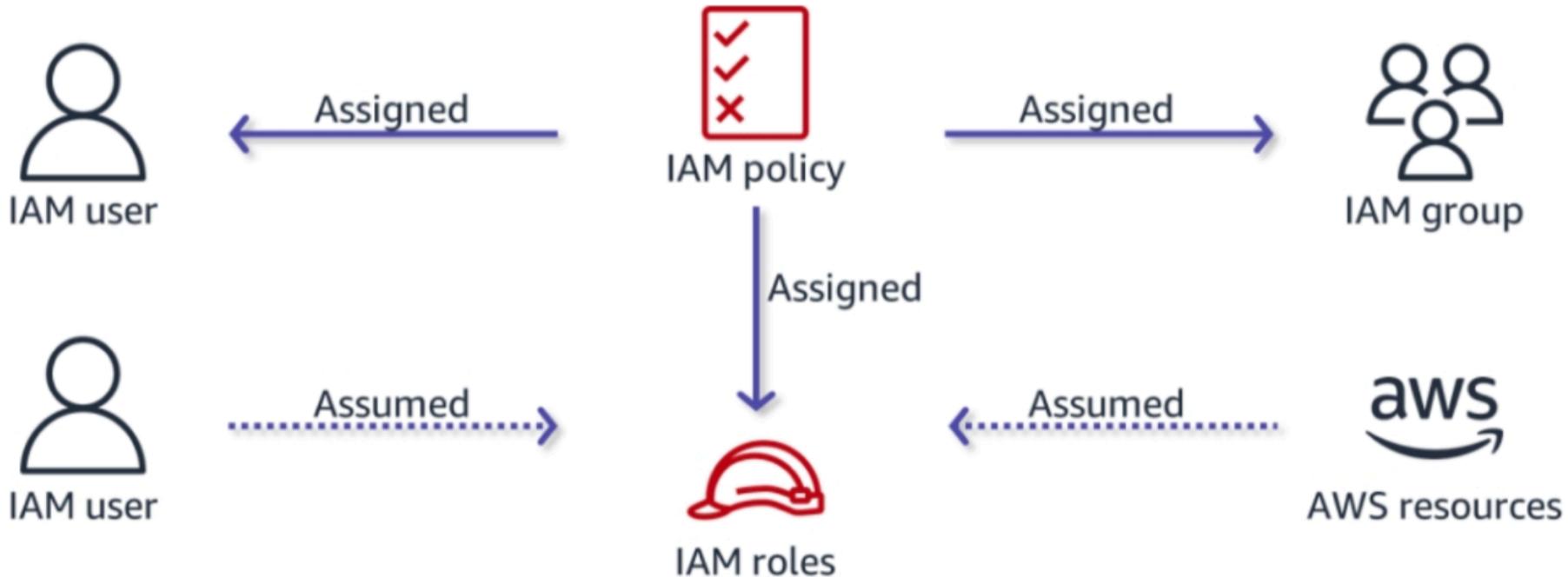


IAM Roles

- Set of permissions can be given to specific users or services.
- The role is assumed without credentials so that the permissions can be valid.

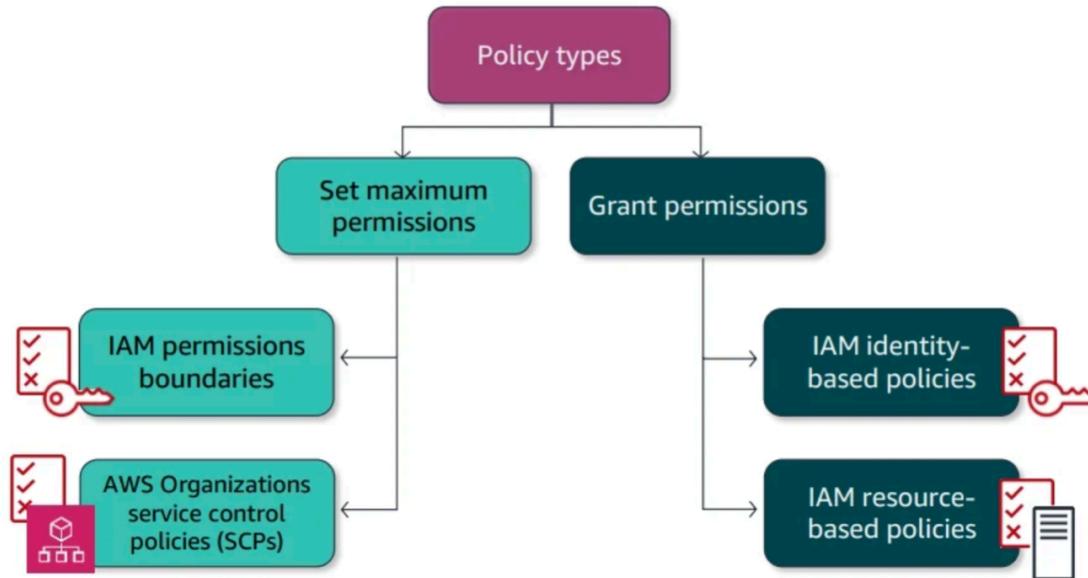


IAM Policy Assignment



Security Policy Categories

- Identity-based policies are assigned to users, groups and roles.
- Resource-based policies are assigned to resources and they are checked when someone tries to access the resource.
- IAM Permission Boundaries limit the user's permissions.



Identity-Based Policy Types

Service access

- AmazonEC2FullAccess
- AmazonEC2ReadOnlyAccess

Job function

- AdministratorAccess
- Billing
- DataScientist

Custom policy

- Level9Admins
- EasternTeam

AWS managed

Customer managed

Identity-Based Policy Example

```
{  
  A "Version": "2012-10-17",  
  "Statement": [  
    {  
      B "Effect": "Allow",  
      "Action": [  
        C "ec2:StartInstances",  
        "ec2:StopInstances"  
      ],  
      "Resource": "arn:aws:ec2:*:*:instance/*",  
      "Condition": {  
        E "StringEquals": {  
          "ec2:ResourceTag/Owner": "${aws:username}"  
        }  
      }  
    }  
  ]  
}
```

- A Use this version date to use all of the available policy features.
- B Indicate whether the policy allows or denies an action.
- C Include a list of actions that the policy allows or denies.
- D Choose a list of resources to which the effect applies.
- E Optional: Specify the conditions under which the policy applies.

Resource-Based Policy Example

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "StartStopIfTags",  
            "Effect": "Allow",  
            "Action": [  
                "ec2:StartInstances",  
                "ec2:StopInstances",  
                "ec2:DescribeTags"  
            ],  
            "Resource": "arn:aws:ec2:<region>:<account-id>:instance/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:ResourceTag/Project": "DataAnalytics",  
                    "aws:PrincipalTag/Department": "Data"  
                }  
            }  
        }  
    ]  
}
```

Allow starting or stopping instances and describing the tags

All instances in the specified region and account id only

Only those instances with tag Project : DataAnalytics

Only by the Principals with tag Department : Data

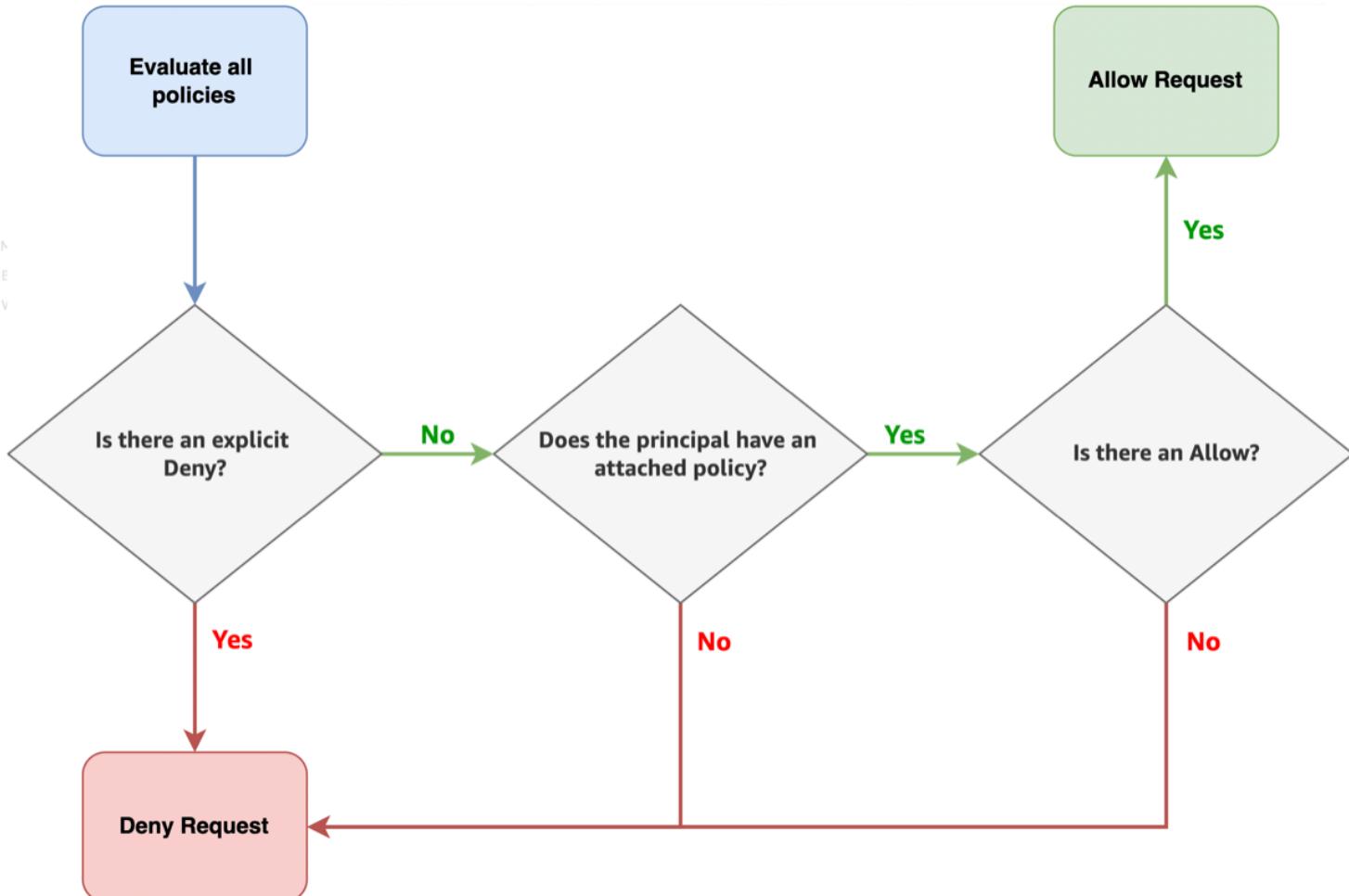
Explicit Allow and Explicit Deny

```
{  
  "Effect": "Allow",  
  "Action": [  
    "s3>ListBucket",  
    "s3GetObject"  
  ],  
  "Resource": [  
    "arn:aws:s3:::DOC-EXAMPLE-BUCKET",  
    "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"  
  ]  
}
```

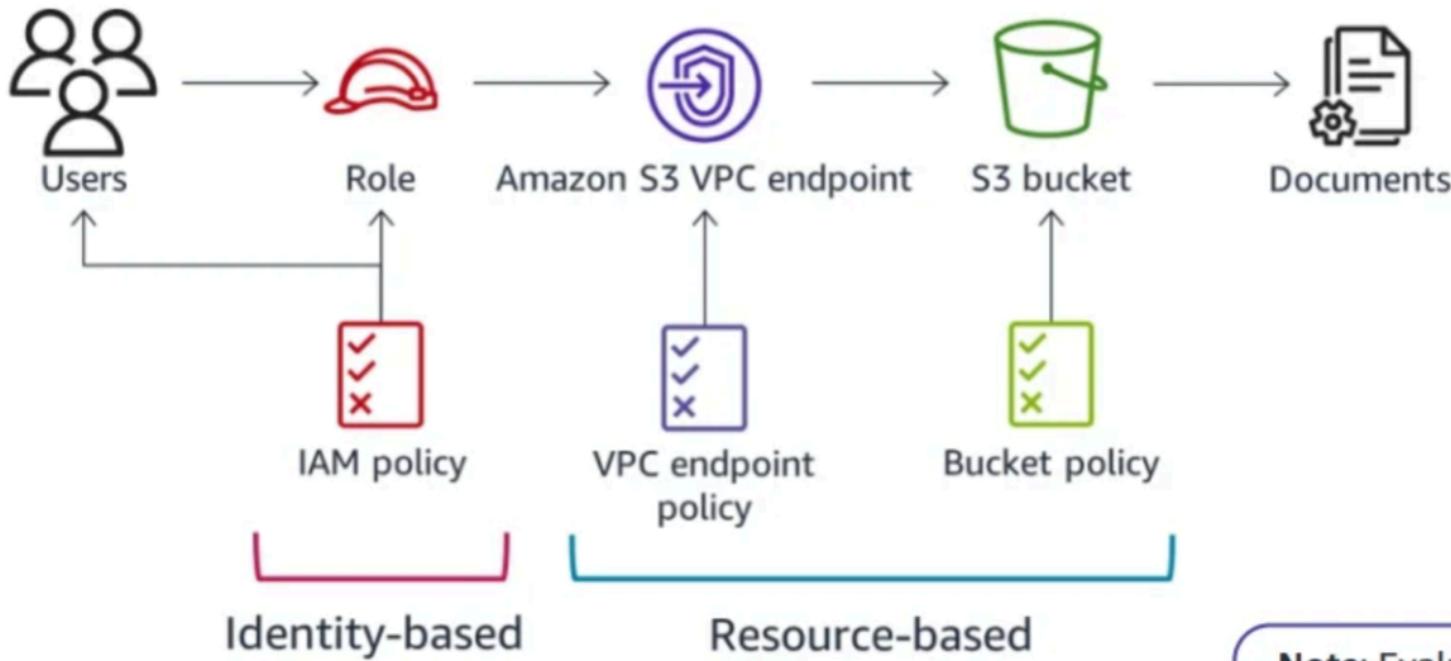
```
{  
  "Effect": "Deny",  
  "Action": [  
    "ec2*",  
    "s3*"  

```

Policy Evaluation



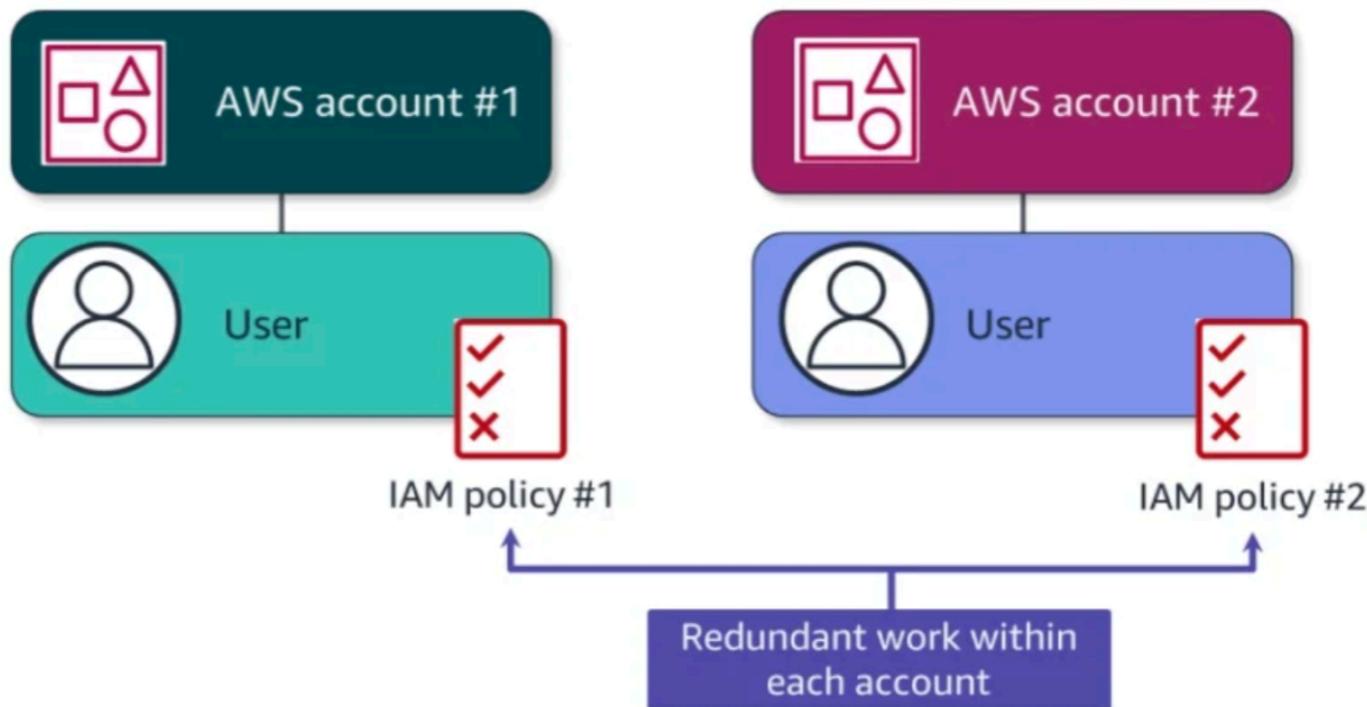
Architecture Tip



Note: Evaluate identity-based policies and resource-based policies together.

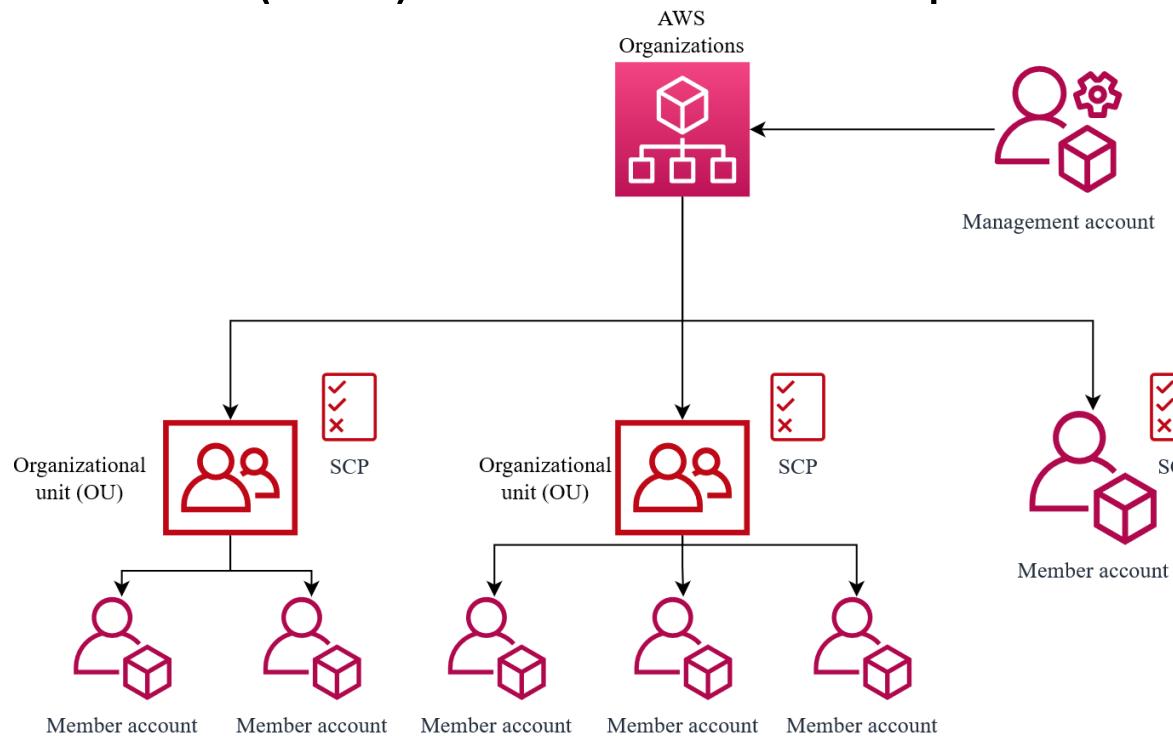
Without AWS Organizations

- IAM Policies only apply to individual principals in a single account.
- Policies must be managed in each account, resulting in generation of multiple bills.



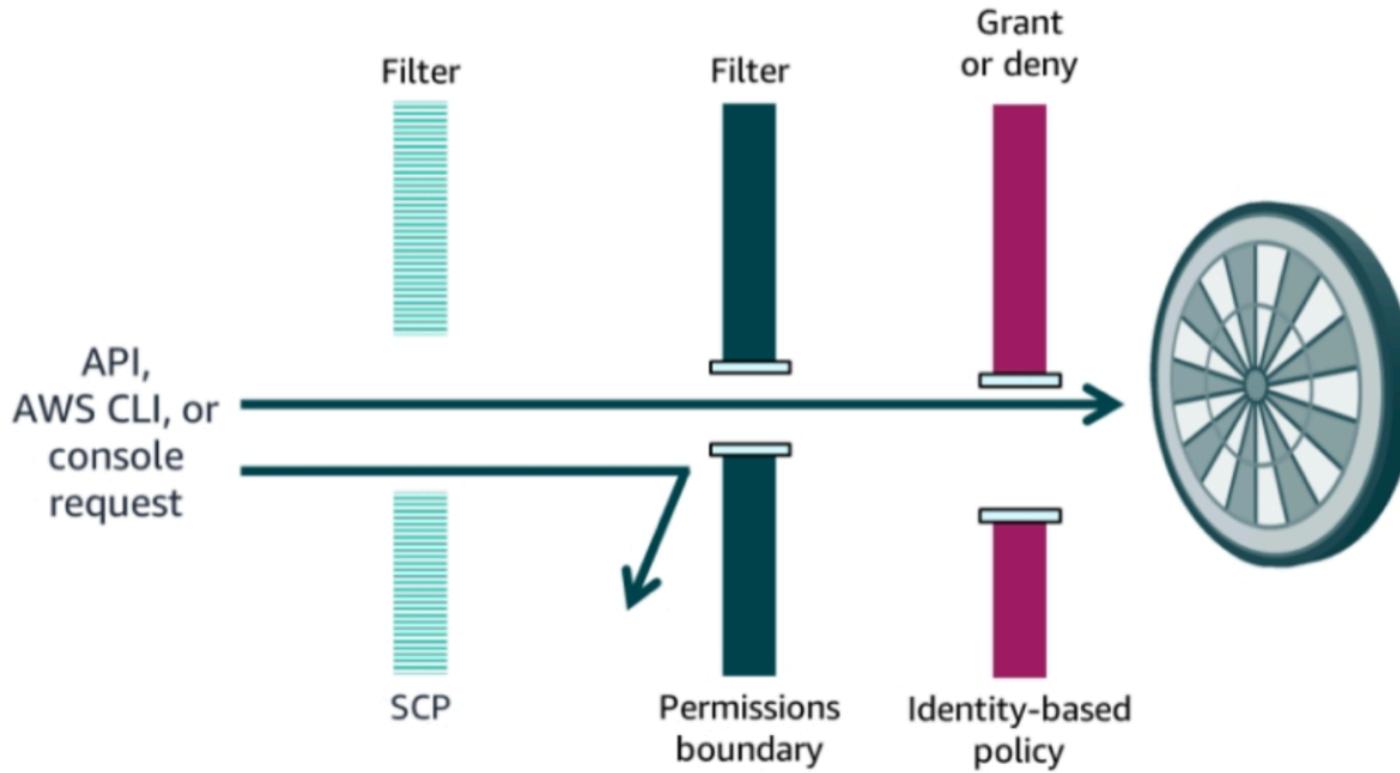
With AWS Organizations

- Enable to develop multi-account strategy for business needs.
- Create a hierarchy by grouping accounts into Organizational Units (OUs).
- Service Control Policies (SCPs) control maximum permissions in every account under OUs.



Layered Defense with Policies

- SCPs don't grant permissions, they only act as a filter!



THANKS FOR LISTENING