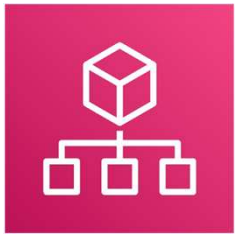
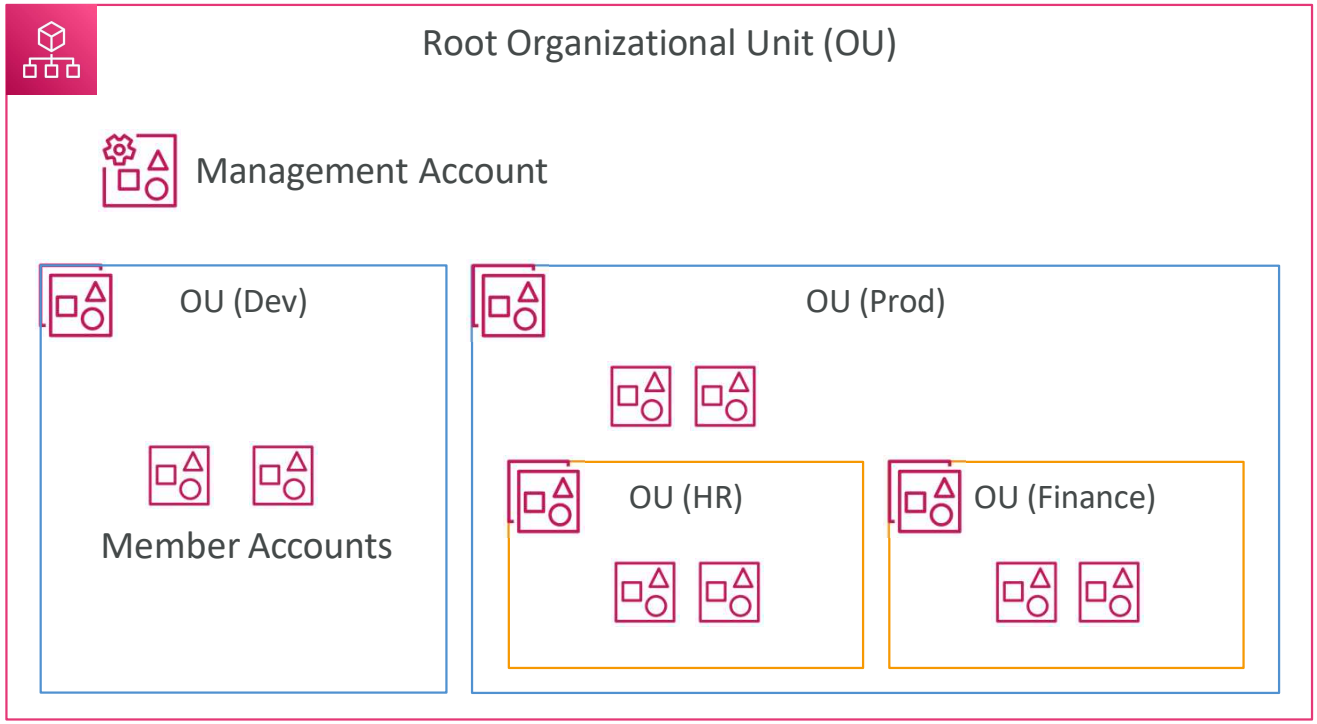
Advanced Identity in AWS

AWS Organizations



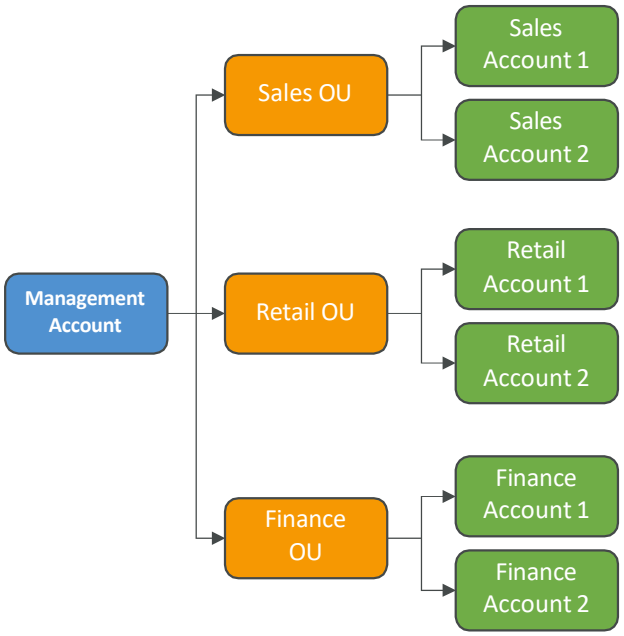
- Global service
- Allows to manage multiple AWS accounts
- The main account is the management account
- Other accounts are member accounts
- Member accounts can only be part of one organization
- Consolidated Billing across all accounts - single payment method
- Pricing benefits from aggregated usage (volume discount for EC2, S3...)
- Shared reserved instances and Savings Plans discounts across accounts
- API is available to automate AWS account creation

AWS Organizations

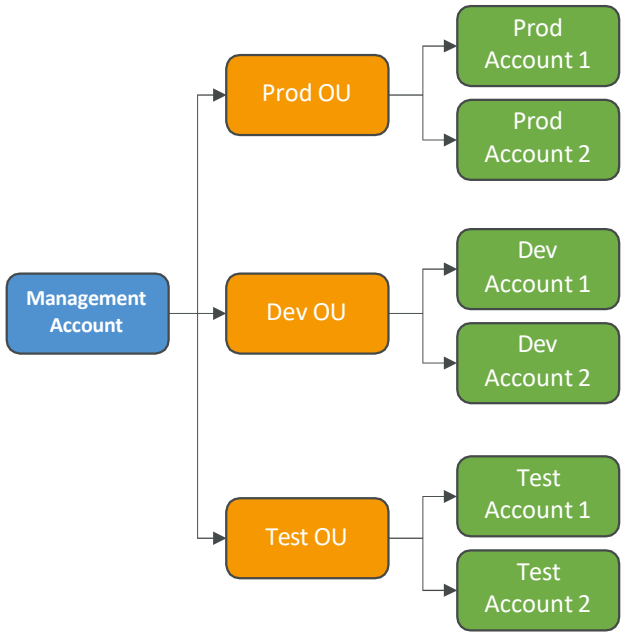


Organizational Units (OU) - Examples

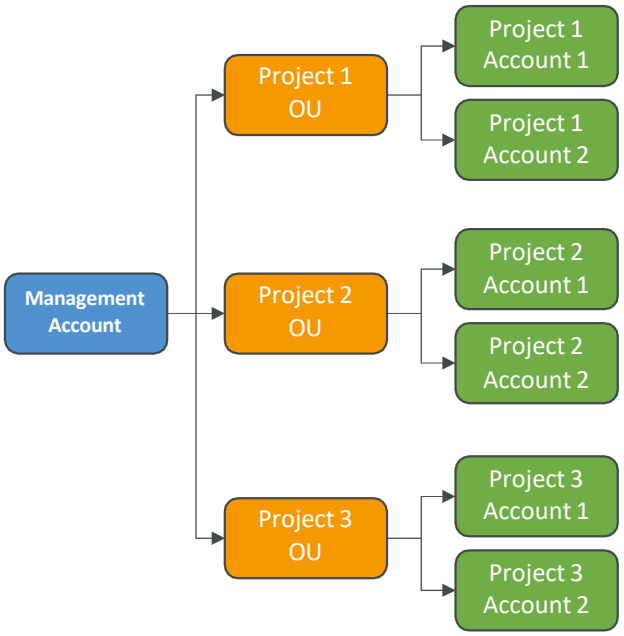
Business Unit



Environmental Lifecycle



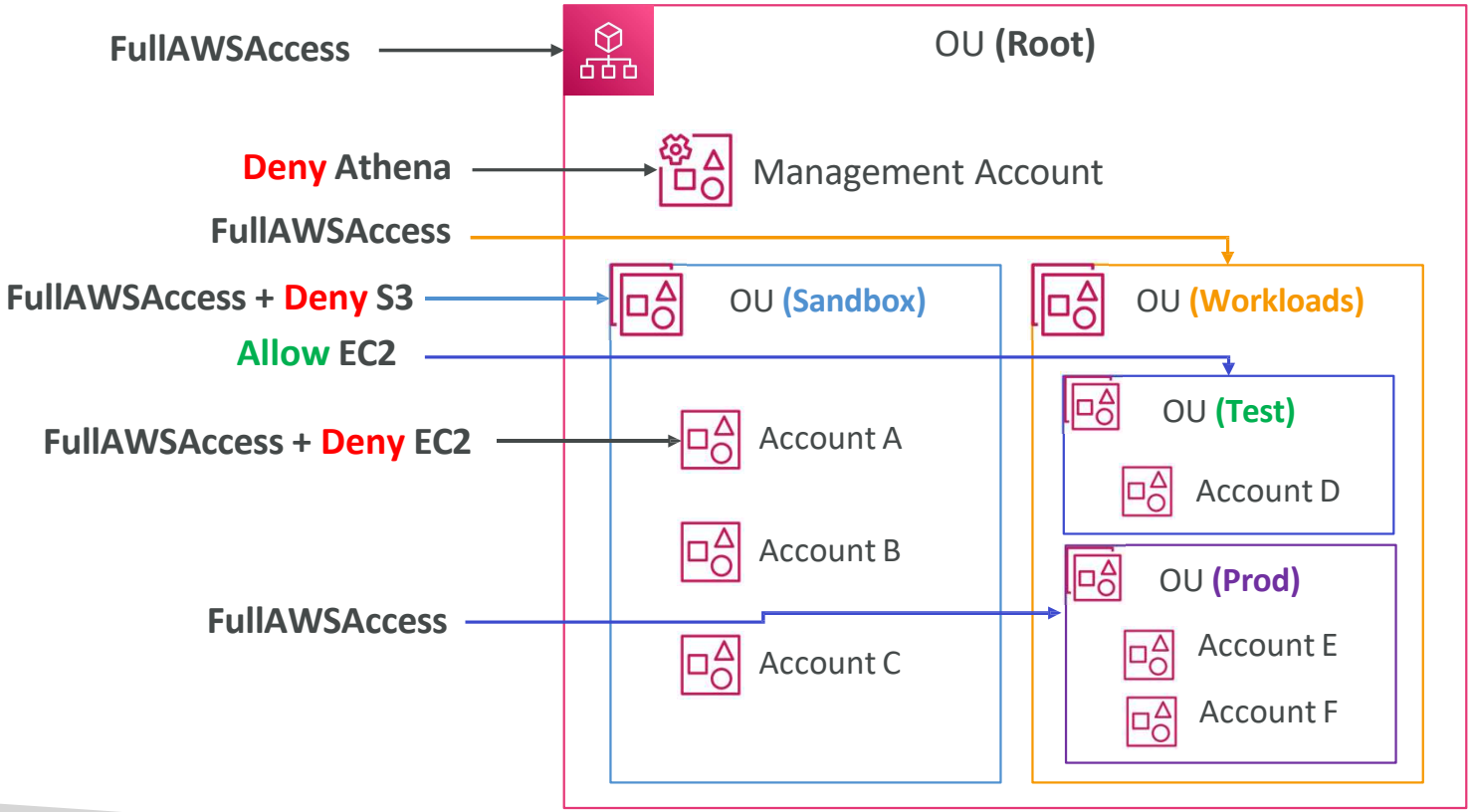
Project-Based



AWS Organizations

- Advantages
  - Multi Account vs One Account Multi VPC
  - Use tagging standards for billing purposes
  - Enable CloudTrail on all accounts, send logs to central S3 account
  - Send CloudWatch Logs to central logging account
  - Establish Cross Account Roles for Admin purposes
- Security: Service Control Policies (SCP)
  - IAM policies applied to OU or Accounts to restrict Users and Roles
  - They do not apply to the management account (full admin power)
  - Must have an explicit allow from the root through each OU in the direct path to the target account (does not allow anything by default – like IAM)

SCP Hierarchy



- Management Account
  - Can do anything (no SCP apply)
- Account A
  - Can do anything
  - EXCEPT S3 (explicit Deny from Sandbox OU)
  - EXCEPT EC2 (explicit Deny)
- Account B & C
  - Can do anything
  - EXCEPT S3 (explicit Deny from Sandbox OU)
- Account D
  - Can access EC2
- Prod OU & Account E & F
  - Can do anything

SCP Examples
Blocklist and Allowlist strategies

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAllActions",
            "Effect": "Allow",
            "Action": "*",
            "Resource": "*"
        },
        {
            "Sid": "DenyDynamoDB",
            "Effect": "Deny",
            "Action": "dynamodb:*",
            "Resource": "*"
        }
    ]
}
```

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:*",
                "cloudwatch:*"
            ],
            "Resource": "*"
        }
    ]
}
```

More examples: https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies_example-scps.html

IAM Conditions

aws:SourceIp

restrict the client IP **from** which the API calls are being made

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "NotIpAddress": {
                    "aws:SourceIp": ["192.0.2.0/24", "203.0.113.0/24
                }
            }
        }
    ]
}
```

aws:RequestedRegion

restrict the region the API calls are made **to**

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": ["ec2:*", "rds:*", "dynamodb:*"],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestedRegion": ["eu-central-1", "eu-west-1"]
                }
            }
        }
    ]
}
```

IAM Conditions

ec2:ResourceTag
restrict based on **tags**

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["ec2:startInstances", "ec2:StopInstances"],
            "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/Project": "DataAnalytics",
                    "aws:PrincipalTag/Department": "Data"
                }
            }
        }
    ]
}
```

aws:MultiFactorAuthPresent
to force **MFA**

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:*",
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": ["ec2:StopInstances", "ec2:TerminateInstances"],
            "Resource": "*",
            "Condition": {
                "BoolIfExists": {
                    "aws:MultiFactorAuthPresent": false
                }
            }
        }
    ]
}
```

IAM for S3

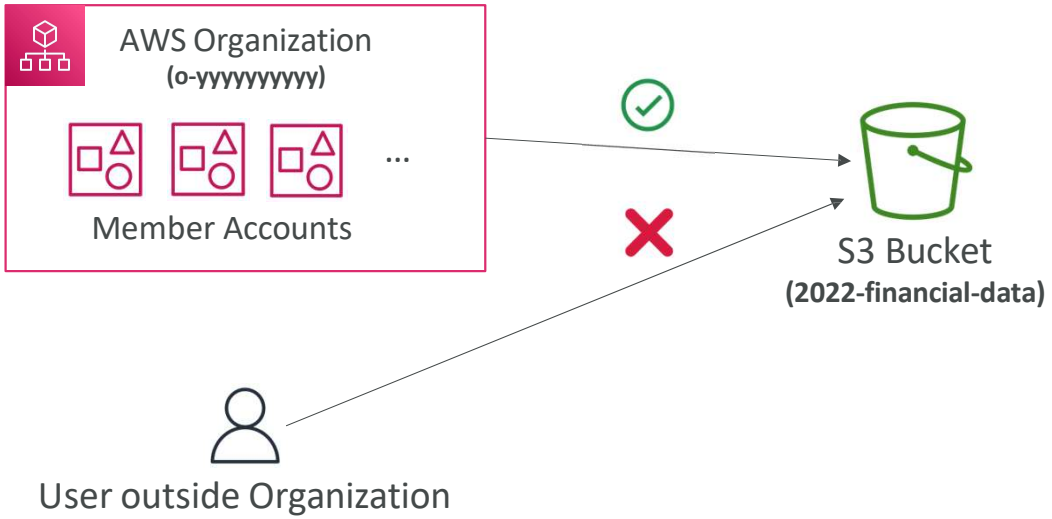
- s3:ListBucket permission applies to
  arn:aws:s3:::test
- => bucket level permission

- s3:GetObject, s3:PutObject, s3:DeleteObject applies to
  arn:awn:s3:::test/*
- => object level permission

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["s3:ListBucket"],
            "Resource": "arn:aws:s3:::test"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObject"
            ],
            "Resource": "arn:aws:s3:::test/*"
        }
    ]
}
```

Resource Policies & aws:PrincipalOrgID

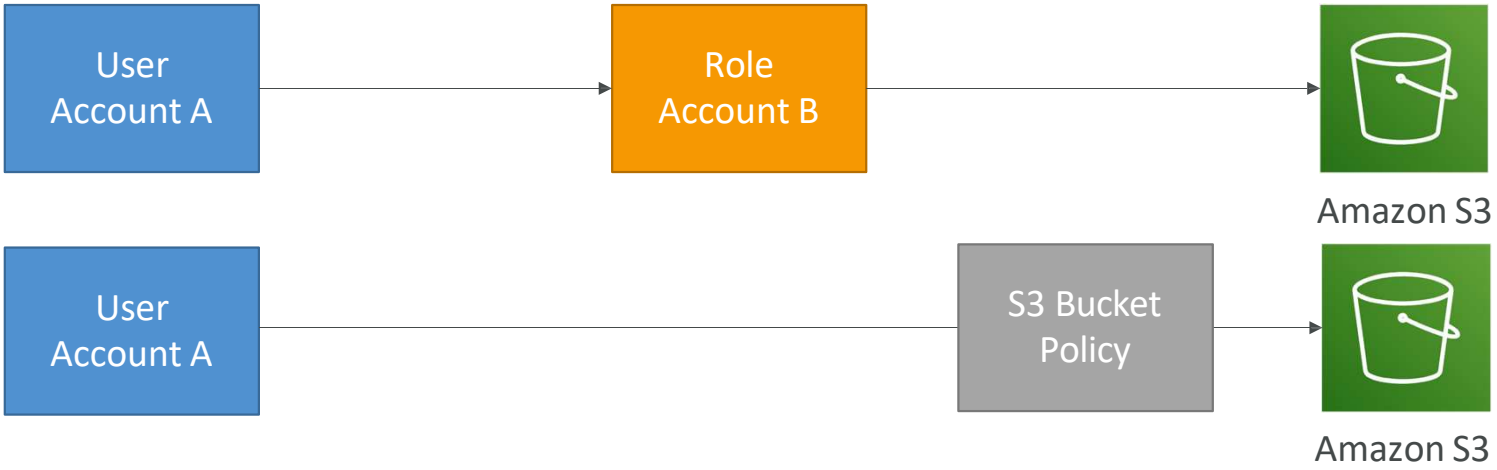
- aws:PrincipalOrgID can be used in any resource policies to restrict access to accounts that are member of an AWS Organization




```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["s3:PutObject", "s3:GetObject"],
            "Resource": "arn:aws:s3:::2022-financial-data/*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": ["o-yyyyyyyyyy"]
                }
            }
        }
    ]
}
```

IAM Roles vs Resource Based Policies

- Cross account:
  - attaching a resource-based policy to a resource (example: S3 bucket policy)
  - OR using a role as a proxy



IAM Roles vs Resource-Based Policies

- When you assume a role (user, application or service), you give up your original permissions and take the permissions assigned to the role

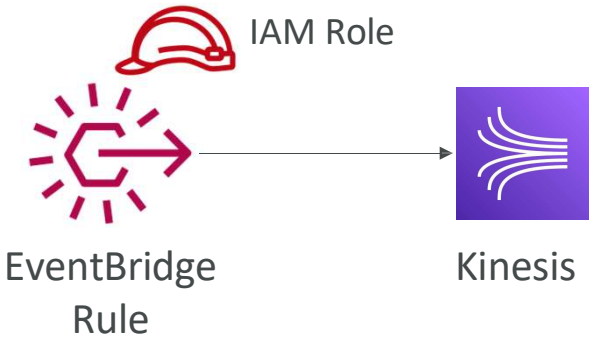
- When using a resource-based policy, the principal doesn't have to give up his permissions

- Example: User in account A needs to scan a DynamoDB table in Account A and dump it in an S3 bucket in Account B.

- Supported by: Amazon S3 buckets, SNS topics, SQS queues, etc...

Amazon EventBridge – Security

- When a rule runs, it needs permissions on the target
- Resource-based policy: Lambda, SNS, SQS, S3 buckets, API Gateway...
- IAM role: Kinesis stream, Systems Manager Run Command, ECS task...



IAM Permission Boundaries

- IAM Permission Boundaries are supported for users and roles (not groups)
- Advanced feature to use a managed policy to set the maximum permissions an IAM entity can get.

Example:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:*",
                "cloudwatch:*",
                "ec2:*"
            ],
            "Resource": "*"
        }
    ]
}
```

IAM Permission Boundary

+

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "iam:CreateUser",
    "Resource": "*"
  }
}
```

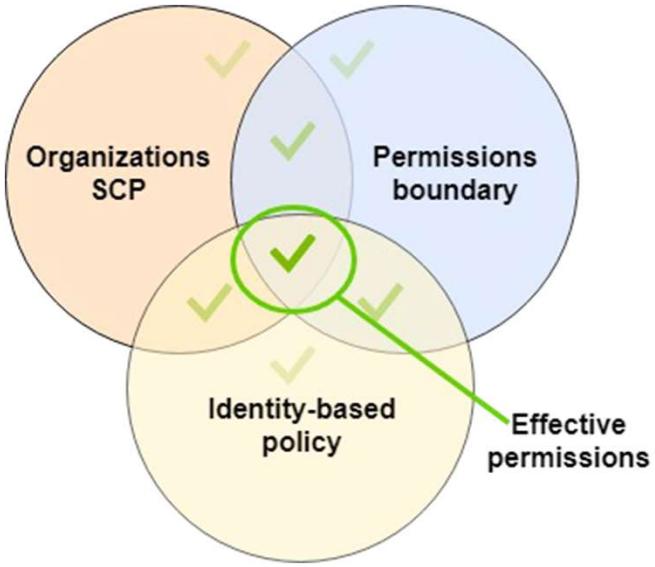
IAM Permissions Through IAM Policy

=

No Permissions

IAM Permission Boundaries

- Can be used in combinations of AWS Organizations SCP

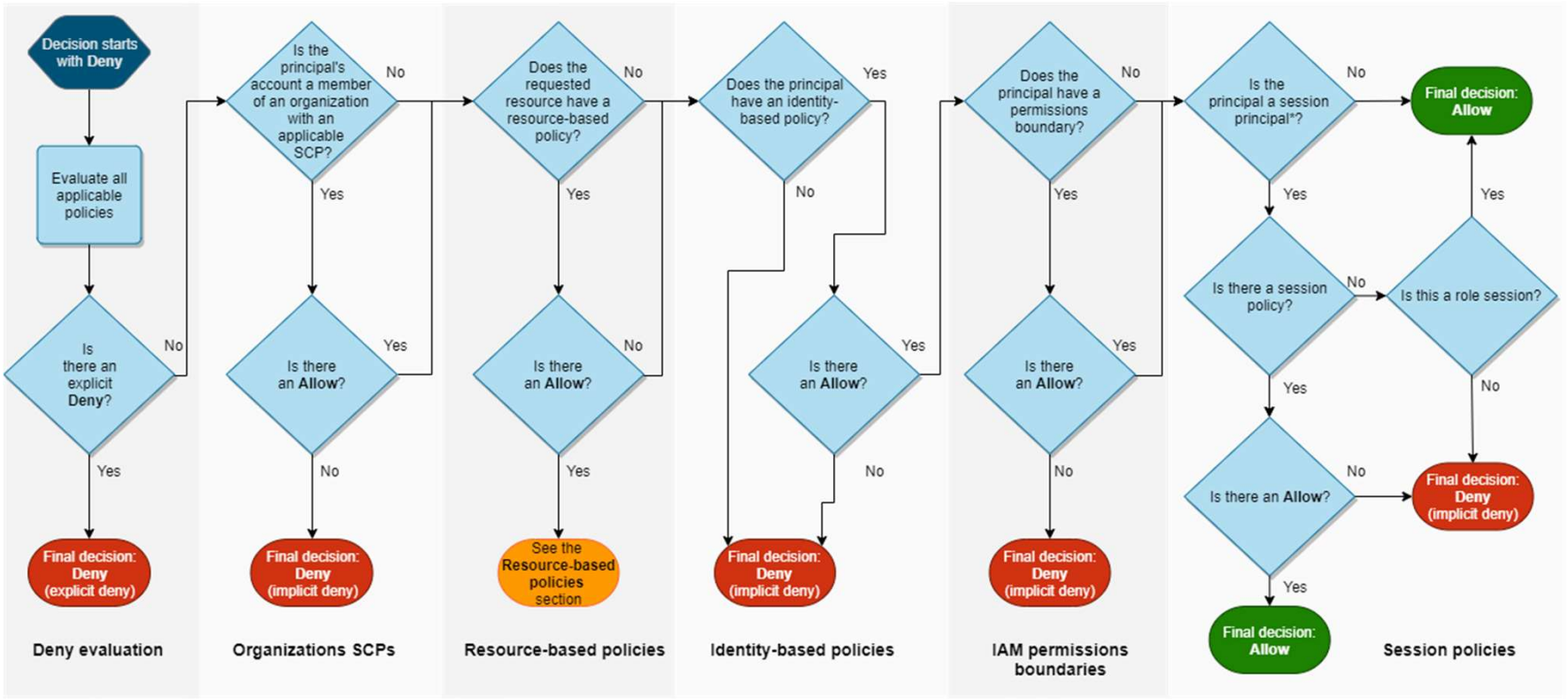


https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_boundaries.html

Use cases

- Delegate responsibilities to non administrators within their permission boundaries, for example create new IAM users
- Allow developers to self-assign policies and manage their own permissions, while making sure they can't "escalate" their privileges (= make themselves admin)
- Useful to restrict one specific user (instead of a whole account using Organizations & SCP)

IAM Policy Evaluation Logic



*A session principal is either a role session or an IAM federated user session.

https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_evaluation-logic.html

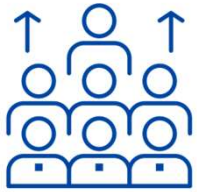
Example IAM Policy

- Can you perform sqs:CreateQueue?
- Can you perform sqs:DeleteQueue?
- Can you perform ec2:DescribeInstances?

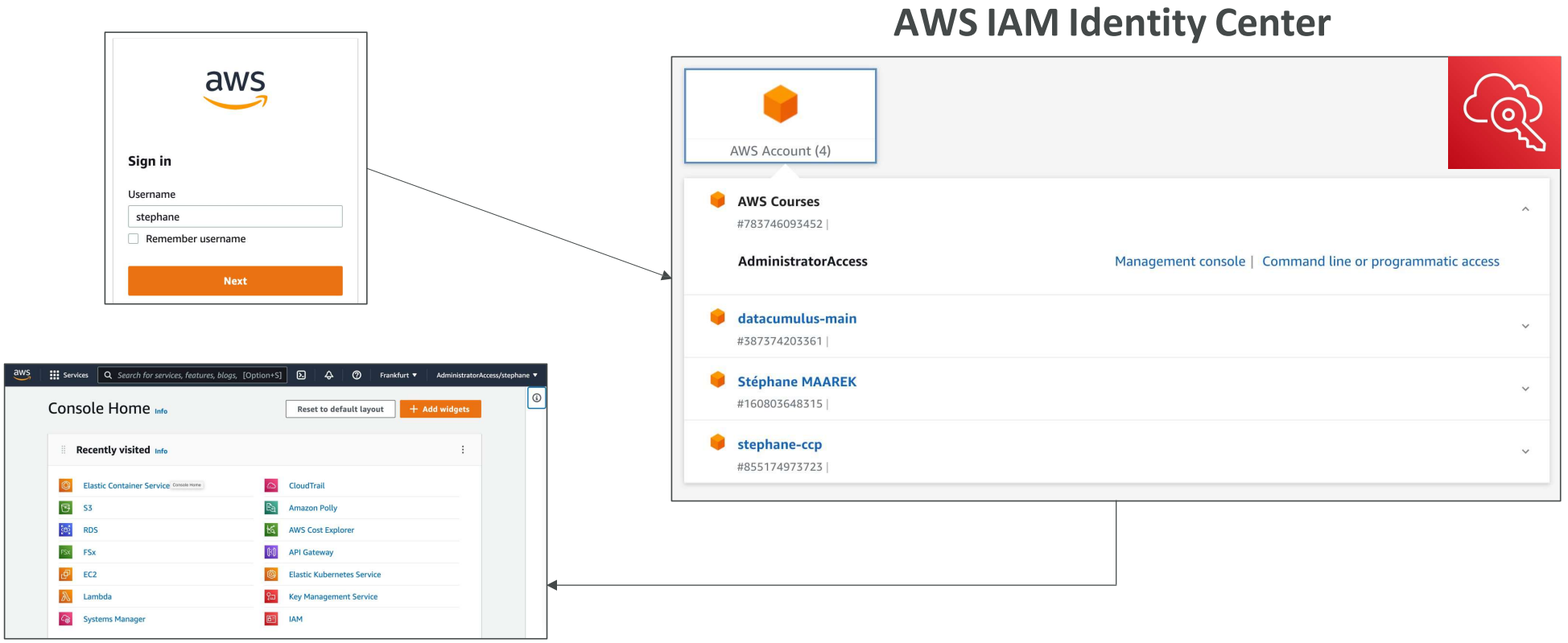
AWS IAM Identity Center (successor to AWS Single Sign-On)

- One login (single sign-on) for all your
  - AWS accounts in AWS Organizations
  - Business cloud applications (e.g., Salesforce, Box, Microsoft 365, ...)
  - SAML 2.0-enabled applications
  - EC2 Windows Instances

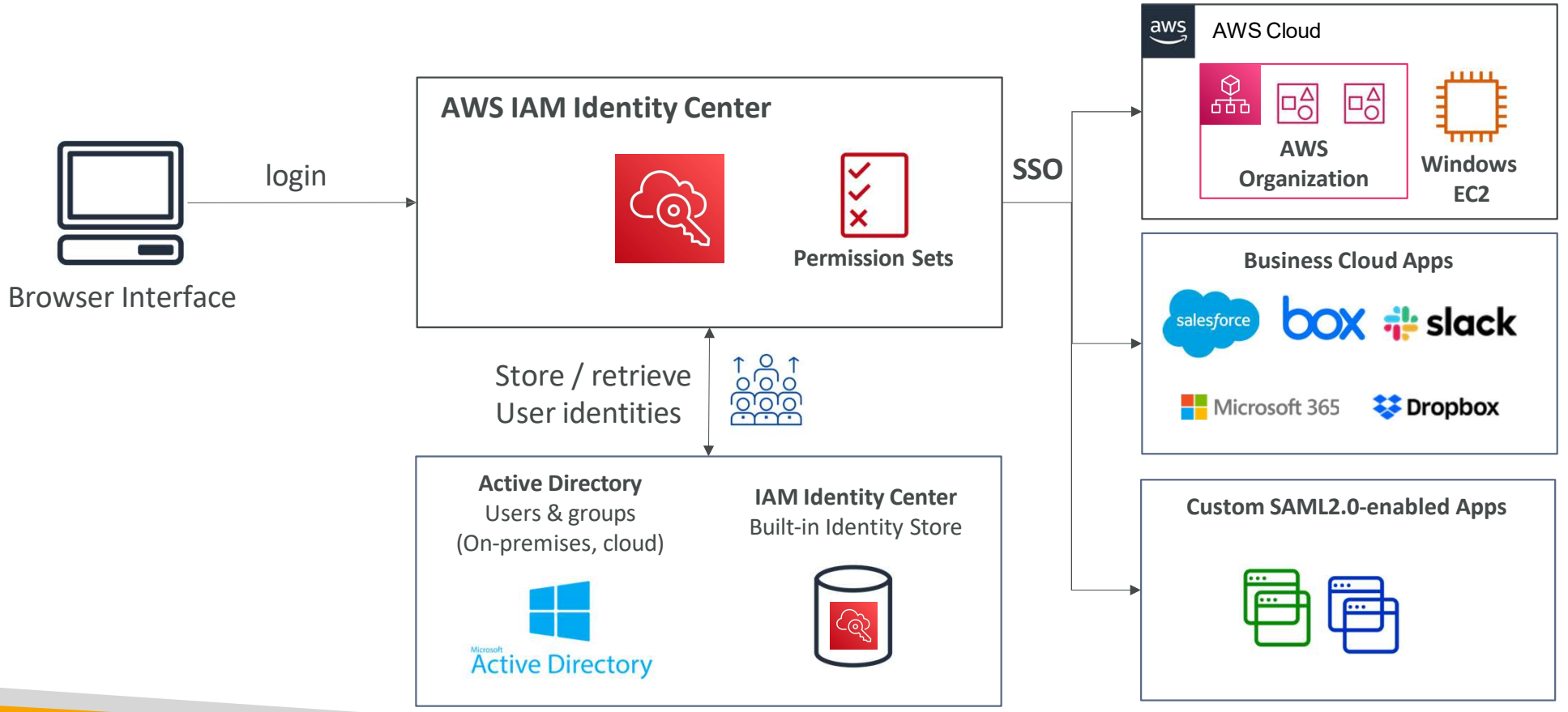
- Identity providers
  - Built-in identity store in IAM Identity Center
  - 3rd party: Active Directory (AD), OneLogin, Okta...



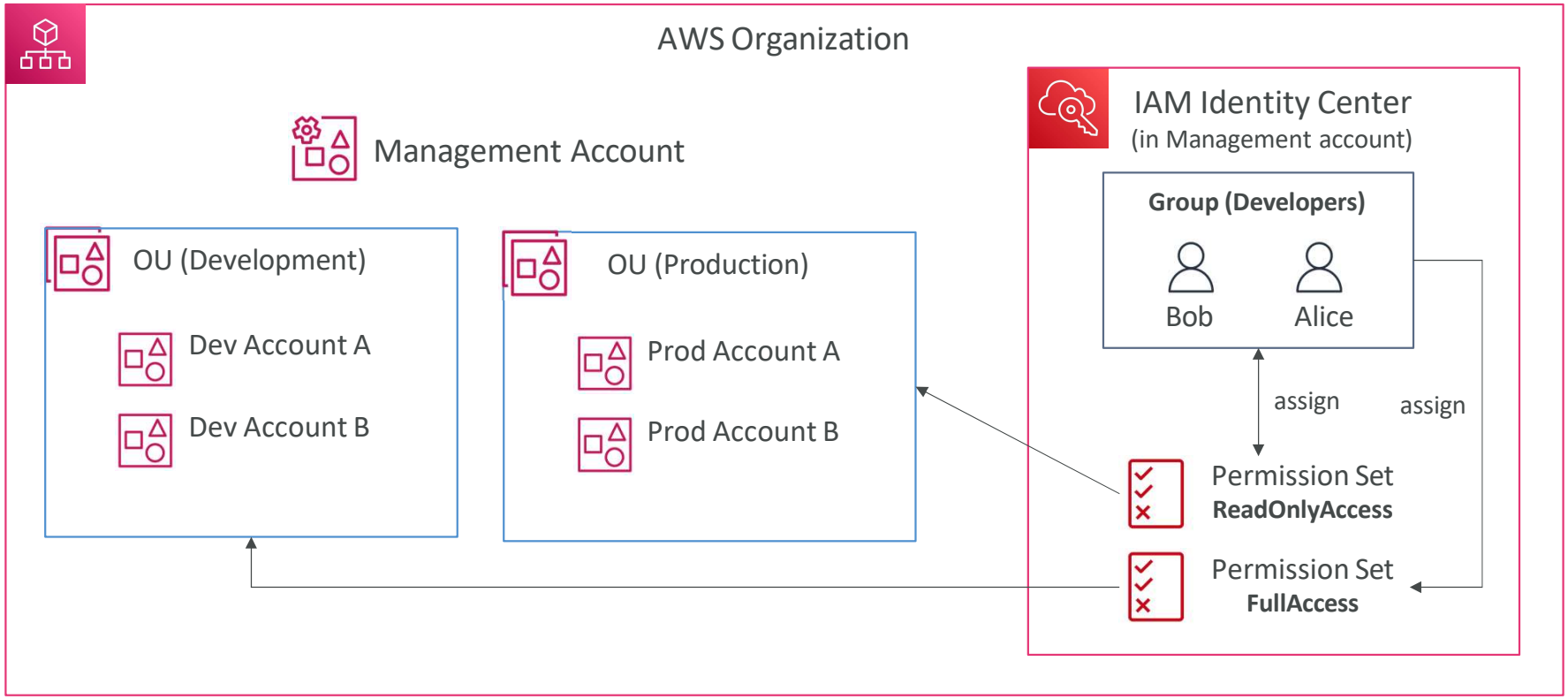
AWS IAM Identity Center - Login Flow



AWS IAM Identity Center

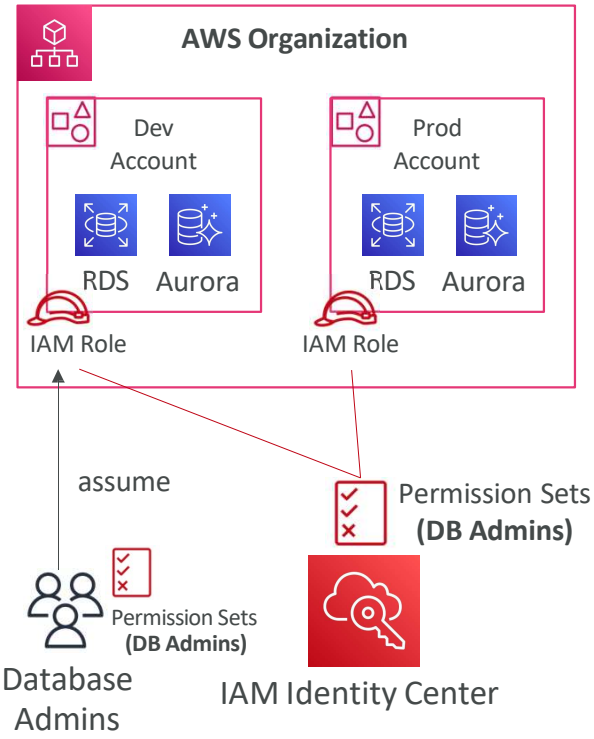


IAM Identity Center



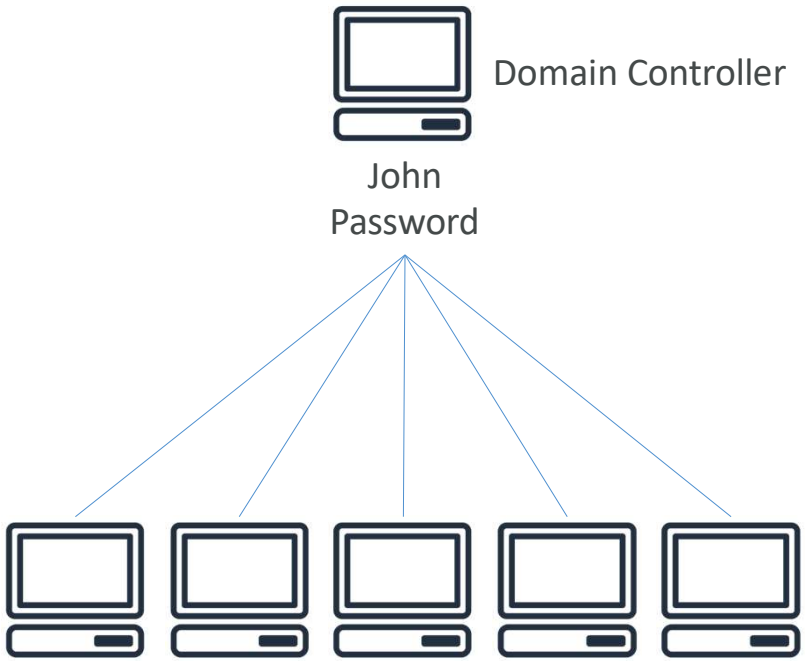
AWS IAM Identity Center
Fine-grained Permissions and Assignments

- Multi-Account Permissions
  - Manage access across AWS accounts in your AWS Organization
  - Permission Sets – a collection of one or more IAM Policies assigned to users and groups to define AWS access
- Application Assignments
  - SSO access to many SAML 2.0 business applications (Salesforce, Box, Microsoft 365, ...)
  - Provide required URLs, certificates, and metadata
- Attribute-Based Access Control (ABAC)
  - Fine-grained permissions based on users' attributes stored in IAM Identity Center Identity Store
  - Example: cost center, title, locale, ...
  - Use case: Define permissions once, then modify AWS access by changing the attributes



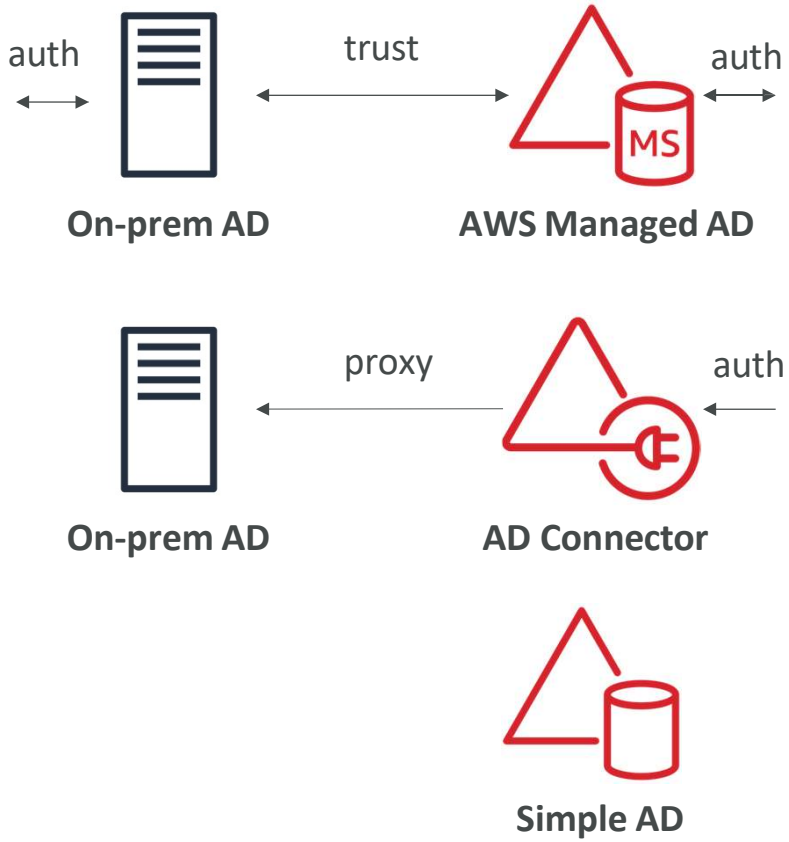
What is Microsoft Active Directory (AD)?

- Found on any Windows Server with AD Domain Services
- Database of objects: User Accounts, Computers, Printers, File Shares, Security Groups
- Centralized security management, create account, assign permissions
- Objects are organized in trees
- A group of trees is a forest



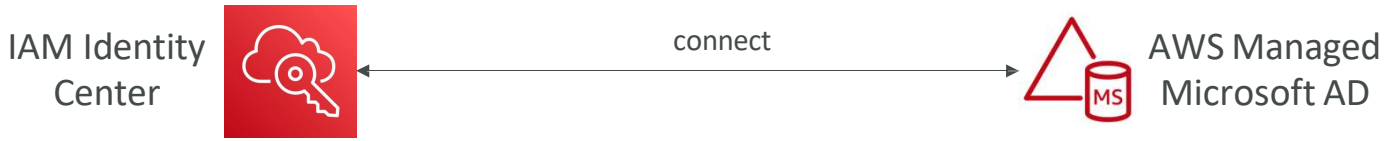
AWS Directory Services

- AWS Managed Microsoft AD
  - Create your own AD in AWS, manage users locally, supports MFA
  - Establish "trust" connections with your on-premises AD
- AD Connector
  - Directory Gateway (proxy) to redirect to on-premises AD, supports MFA
  - Users are managed on the on-premises AD
- Simple AD
  - AD-compatible managed directory on AWS
  - Cannot be joined with on-premises AD

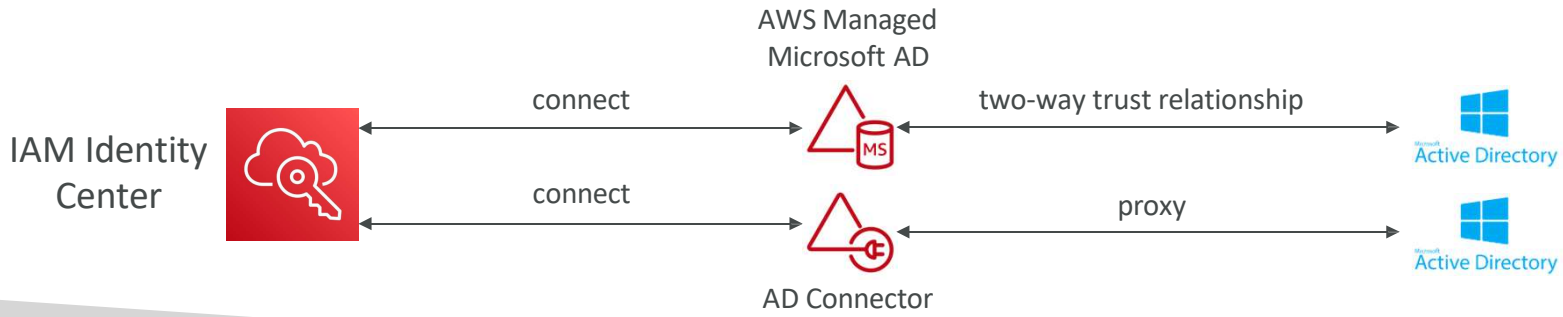


IAM Identity Center – Active Directory Setup

- Connect to an AWS Managed Microsoft AD (Directory Service)
  - Integration is out of the box



- Connect to a Self-Managed Directory
  - Create Two-way Trust Relationship using AWS Managed Microsoft AD
  - Create an AD Connector



AWS Control Tower

- Easy way to set up and govern a secure and compliant multi-account AWS environment based on best practices
- AWS Control Tower uses AWS Organizations to create accounts

- Benefits:
  - Automate the set up of your environment in a few clicks
  - Automate ongoing policy management using guardrails
  - Detect policy violations and remediate them
  - Monitor compliance through an interactive dashboard

AWS Control Tower - Guardrails

- Provides ongoing governance for your Control Tower environment (AWS Accounts)
- Preventive Guardrail - using SCPs (e.g., Restrict Regions across all your accounts)
- Detective Guardrail - using AWS Config (e.g., identify untagged resources)