DevSecOps Days 2023

Washington, D.C.

Scaling Cloud Accounts and How to Secure Access to Multi-Account Estates

OCTOBER 12, 2023

Jason Kao
Head of Security Research and Solutions, CloudQuery

Carnegie Mellon University
Software Engineering Institute

whoami

Jason Kao
Head of Security Research and Solutions, CloudQuery

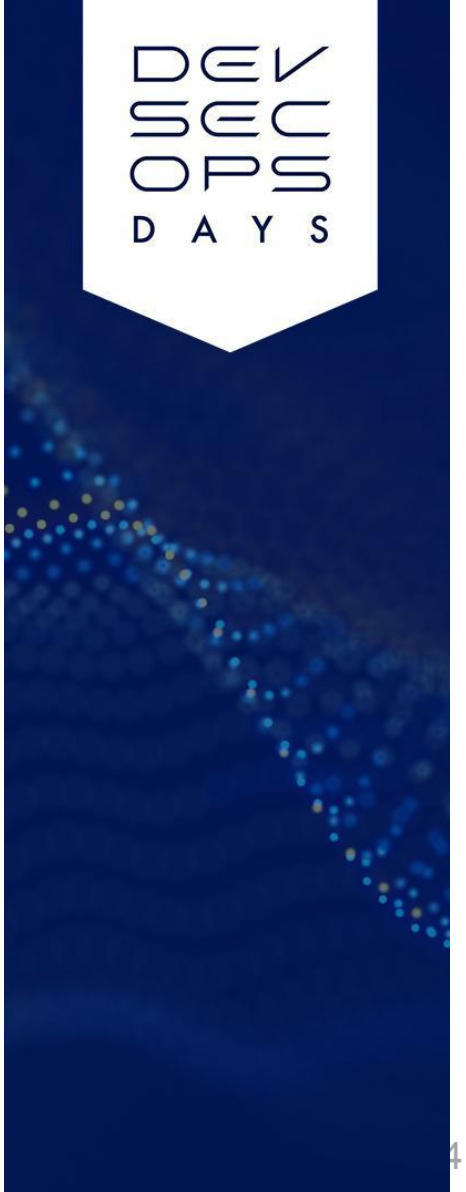
Email: jason@cloudquery.io

- 7+ years of both offensive and defensive cloud security experience including consulting with companies ranging from Fortune 100 companies to Startups across different industries.
- Speaker at conferences including AWS Re:Invent, AWS Re:Inforce, Mandiant mWise.
- Author on multiple Patents.
- Contributor to CIS Security Benchmarks.

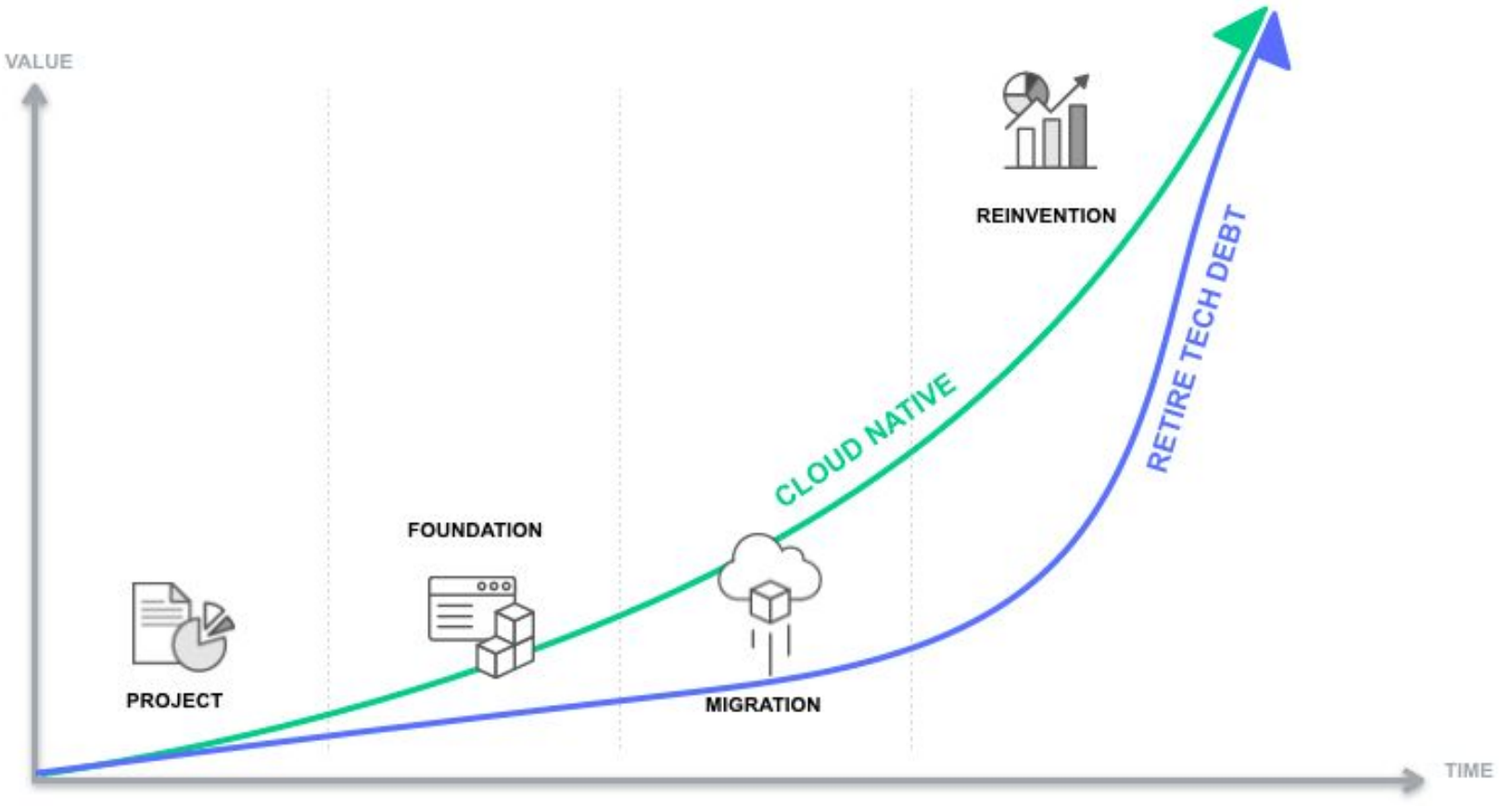
Agenda

- Cloud Adoption Trends
- Introduction to Multiple Accounts
- Security Risks of Multiple Accounts
- Securing Multi-Account Estates
- Platform Enablement and DevSecOps

Cloud Adoption Trends



Cloud Adoption Trends



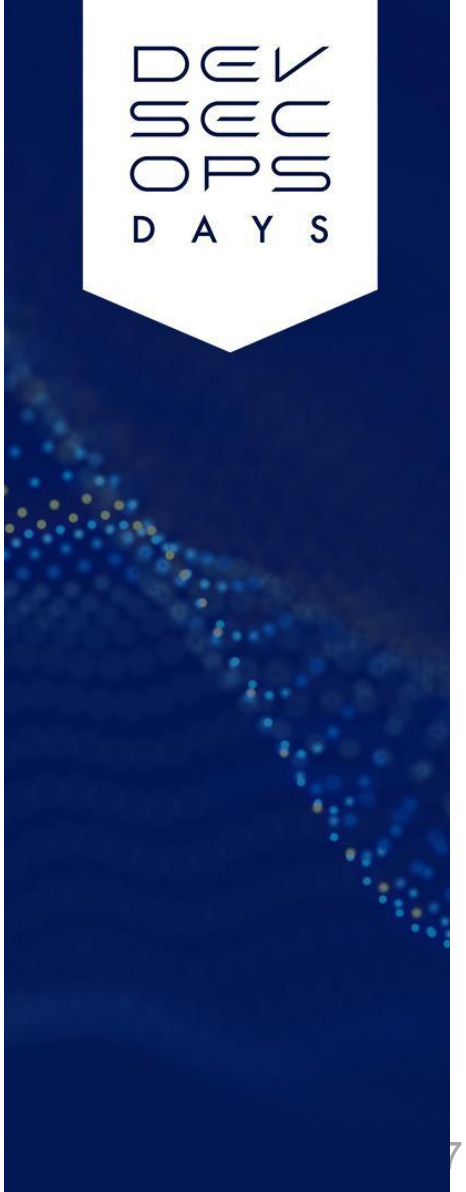
Source: AWS

Multiple Accounts

- **63%** of AWS users have multiple accounts.
- **63%** of Azure users have multiple accounts.
- **65%** of Google Cloud users have multiple accounts.

Source: Orca Cloud Security, March 2023

Introduction to Multiple Accounts



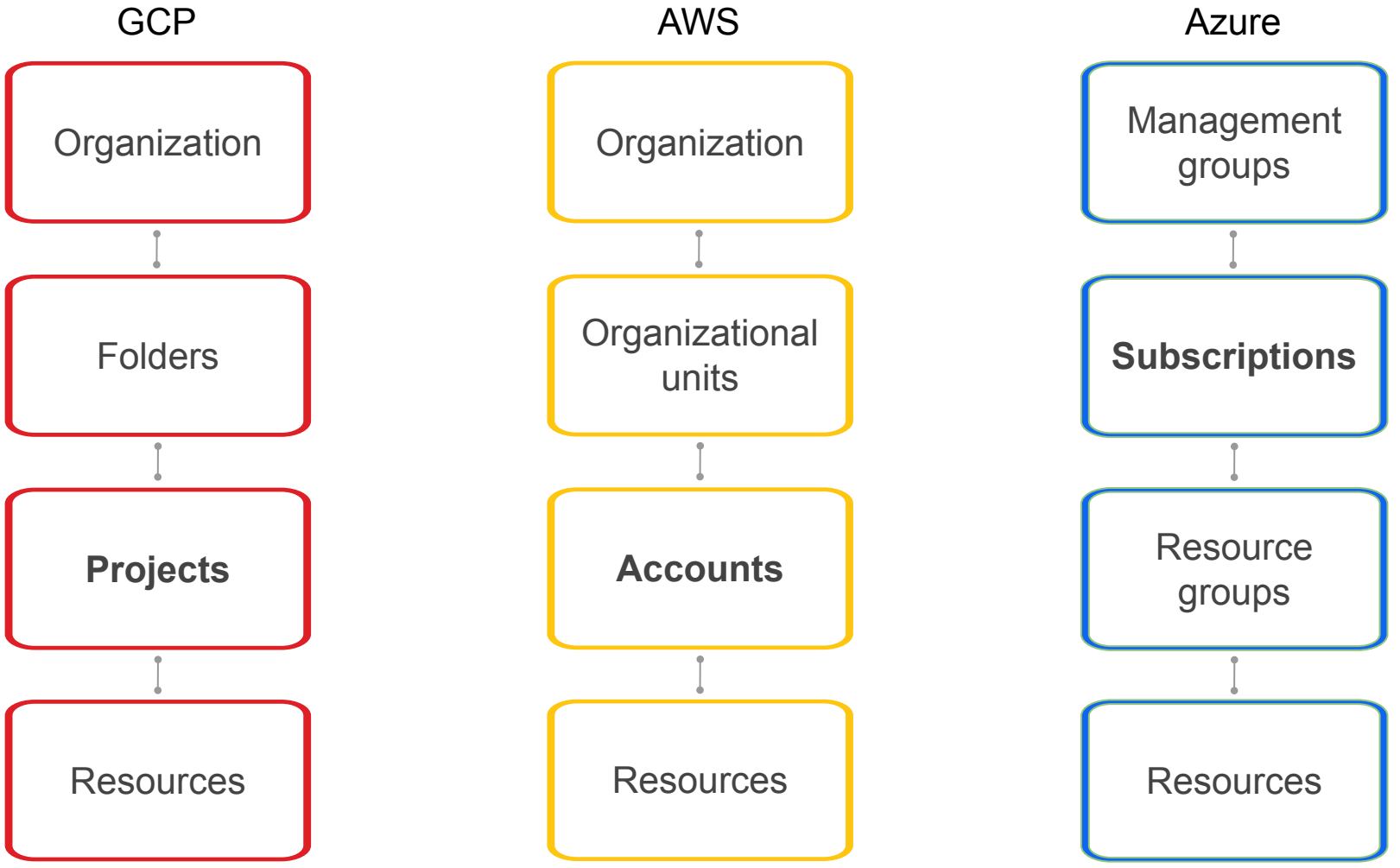
Account

An account is a logical boundary (container) used to contain resources. This logical boundary can be leveraged as a security boundary.

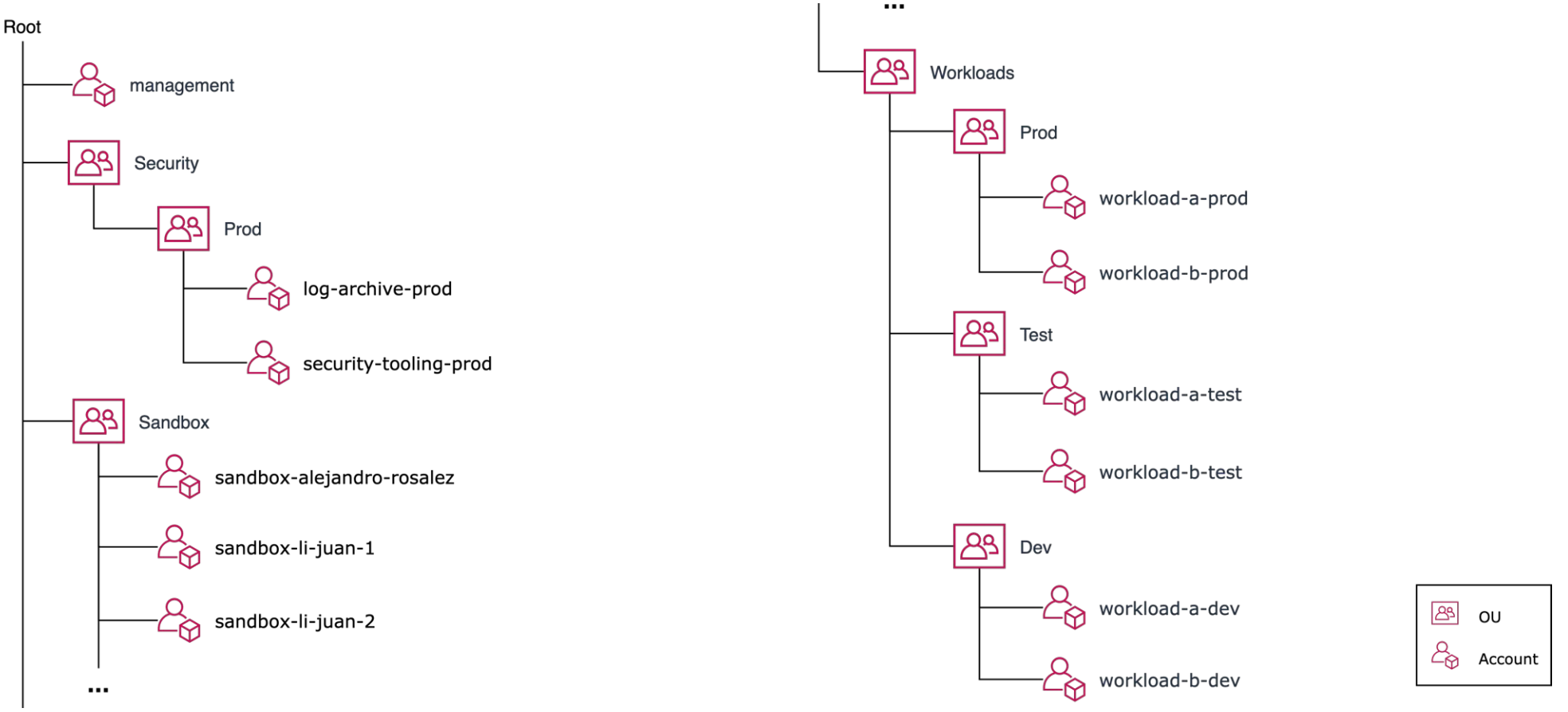
CSP Definitions:

- An AWS account is the basic container for all the AWS resources you create as an AWS customer.
- (Azure) Subscriptions logically associate user accounts with the resources that they create.
- Google Cloud projects form the basis for creating, enabling, and using all Google Cloud services including managing APIs, enabling billing, adding and removing collaborators, and managing permissions for Google Cloud resources.

Account Terminology



Reference Multi-Account Architecture



Source: AWS

Types of Accounts

Management Account:

The account used to create the organization. This account has certain privileges that can affect member accounts.

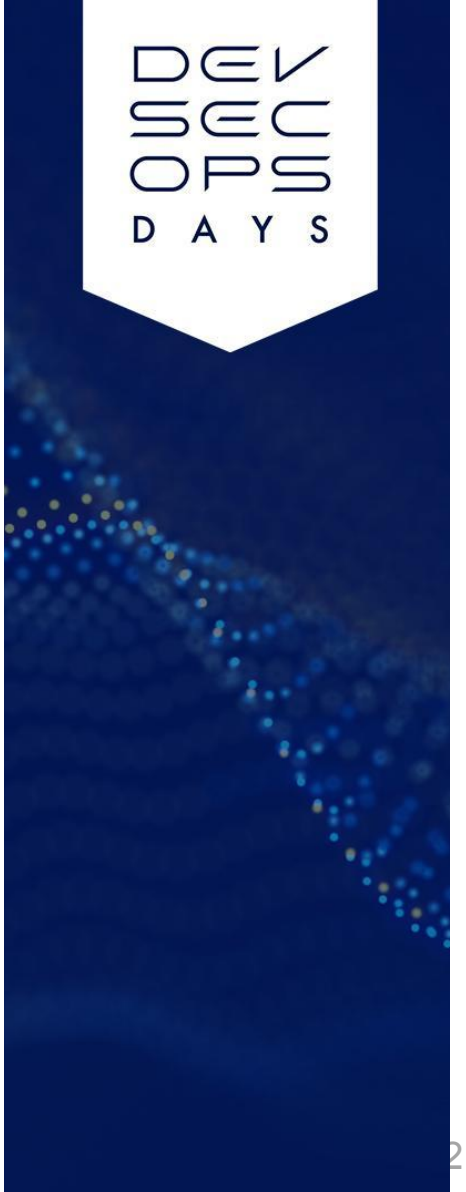
Member Account:

The non-management accounts in the Organization. Only can belong to 1 organization at a time.

Delegated Administrator:

Member accounts can be designated as a delegated administrator to administer and manage tasks and AWS services that may span across accounts and integrate with Organizations.

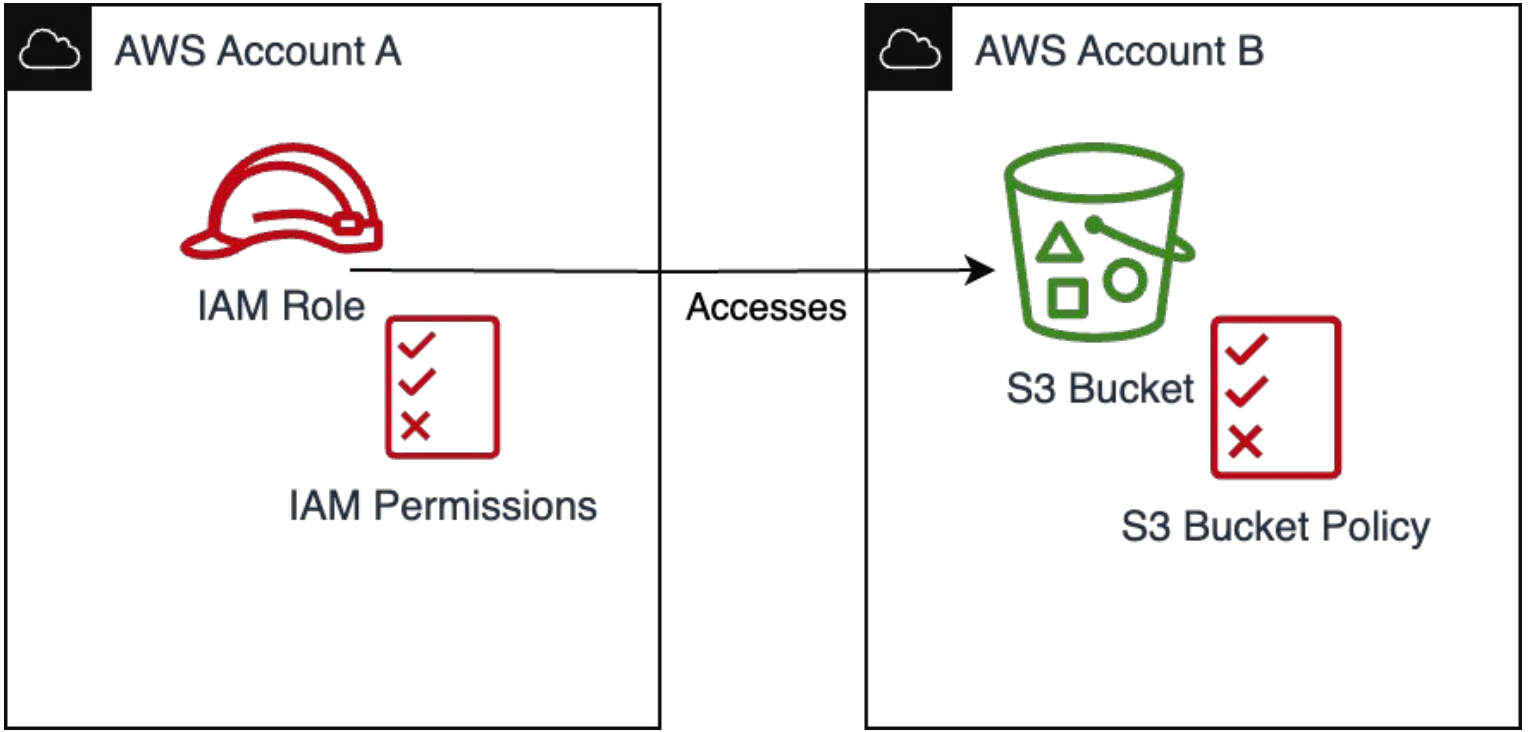
Security Risks of Multiple Accounts



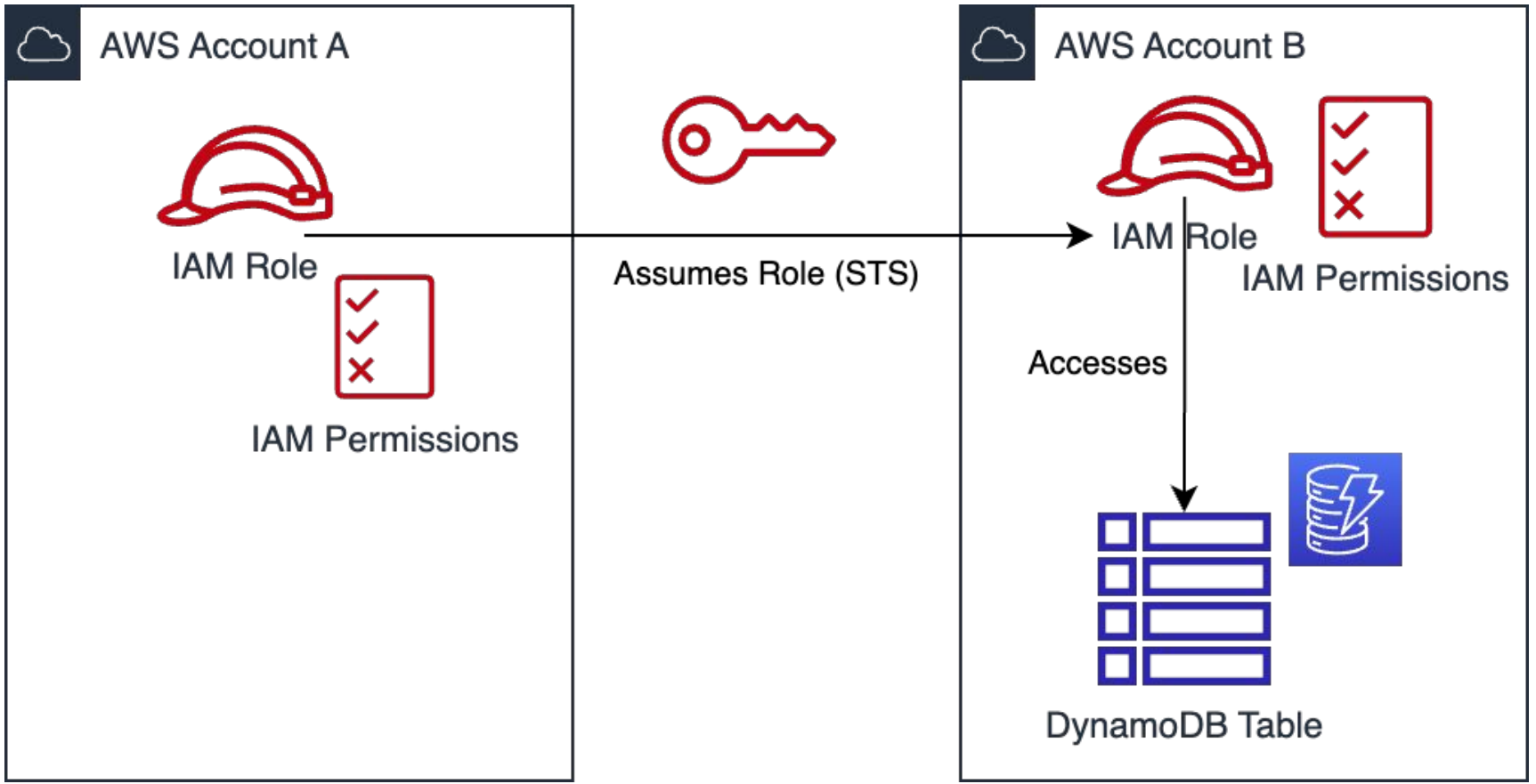
Security Risks of Multi-Accounts

- Identity and Access Management Risks
- Cloud Service Risks
- Reconnaissance Risks

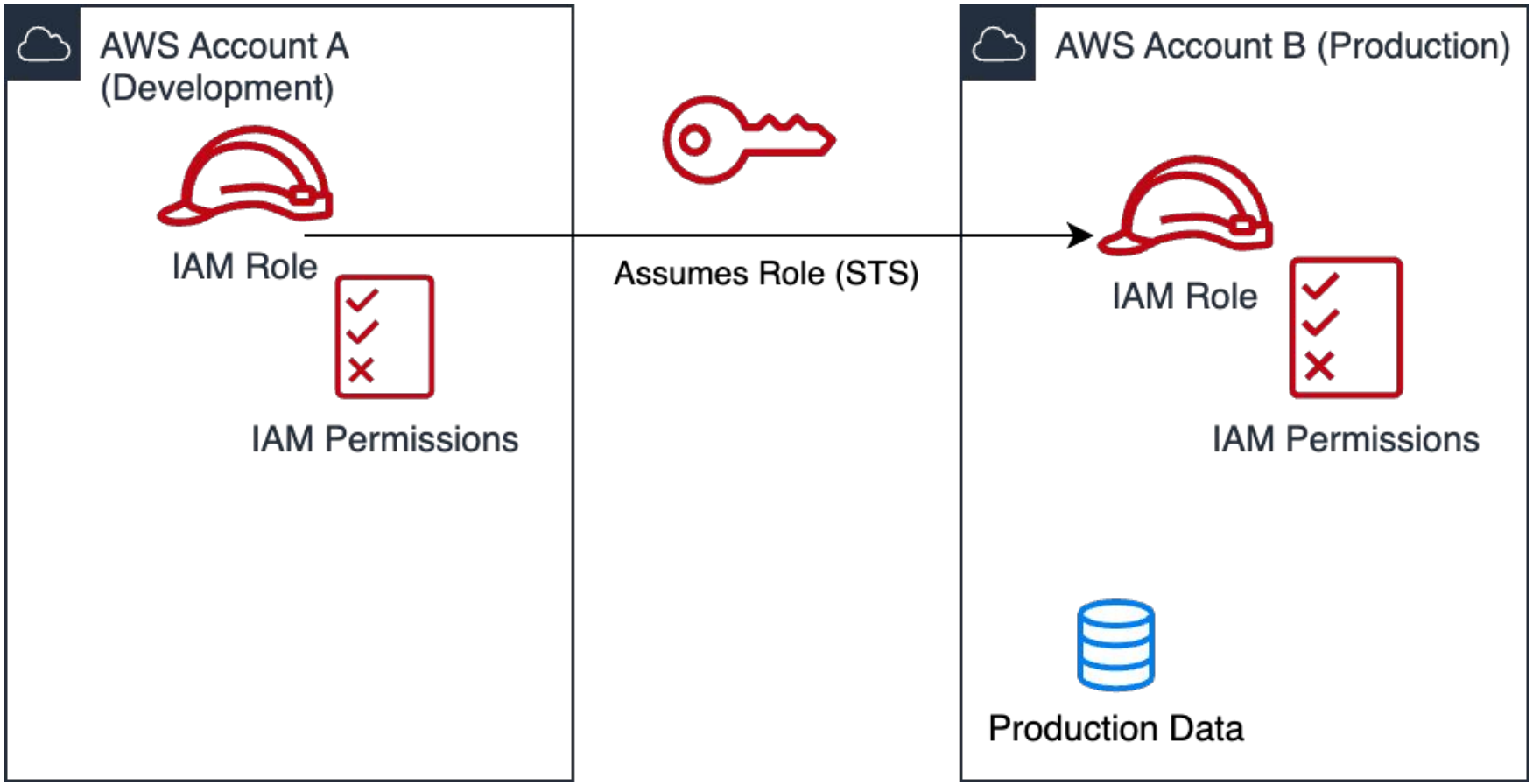
IAM Risks: Cross-Account Access via Resource-Based Policies



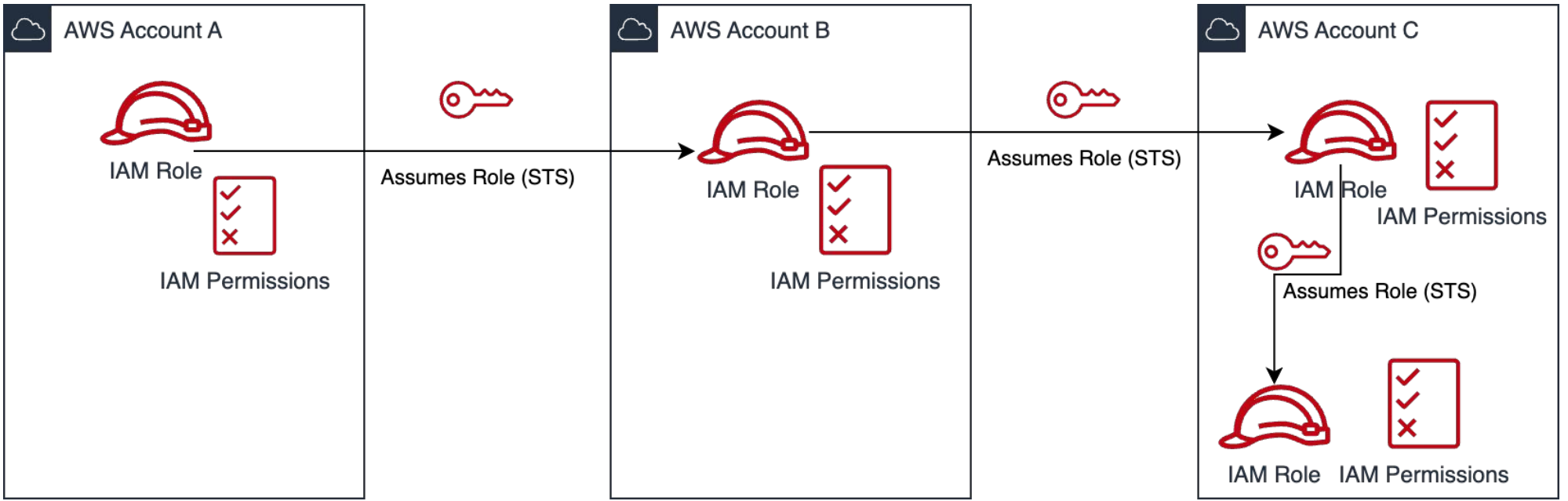
IAM Risks: Cross-Account Access via IAM



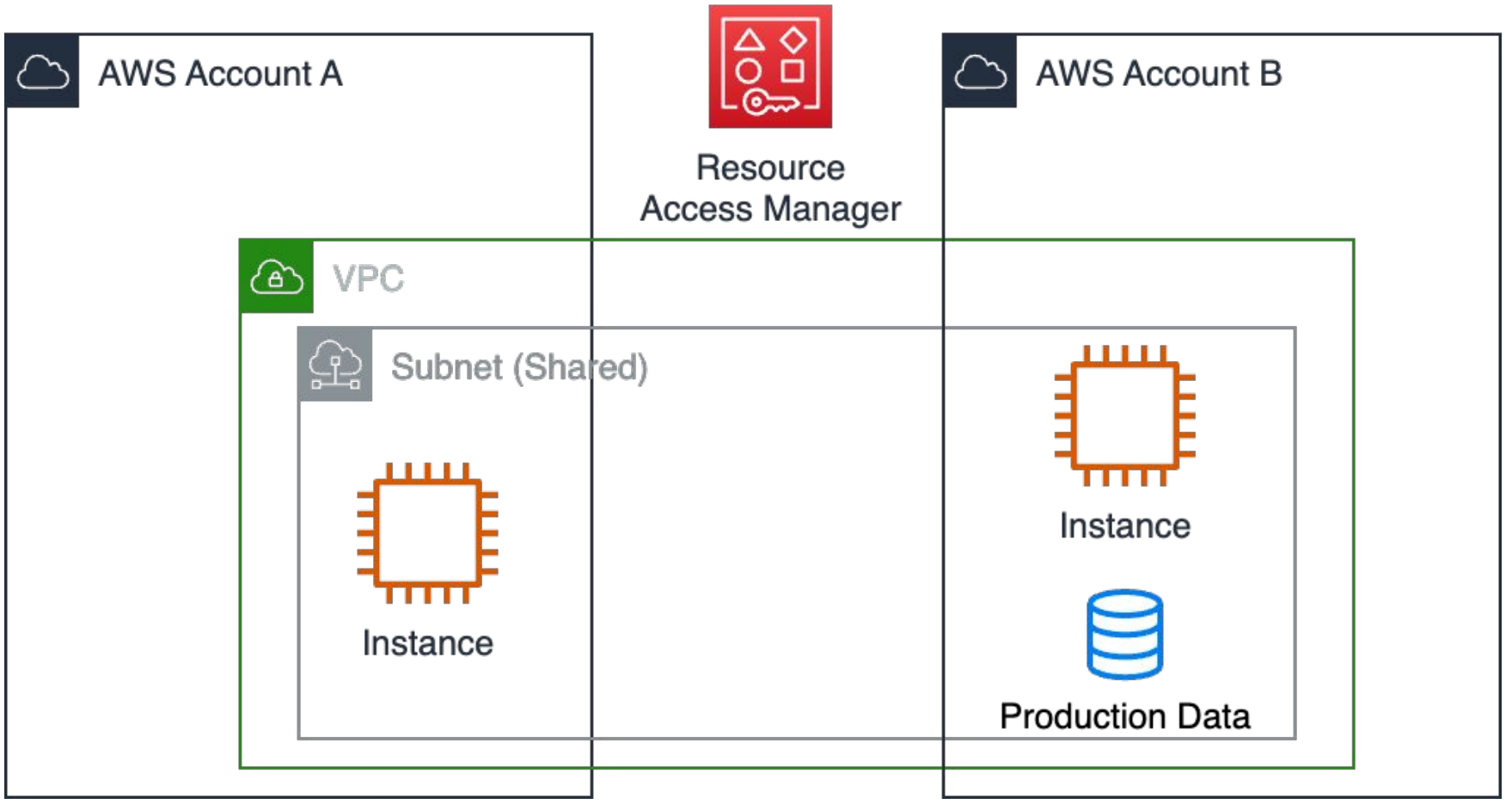
IAM Risks: Lateral Movement Across Environments



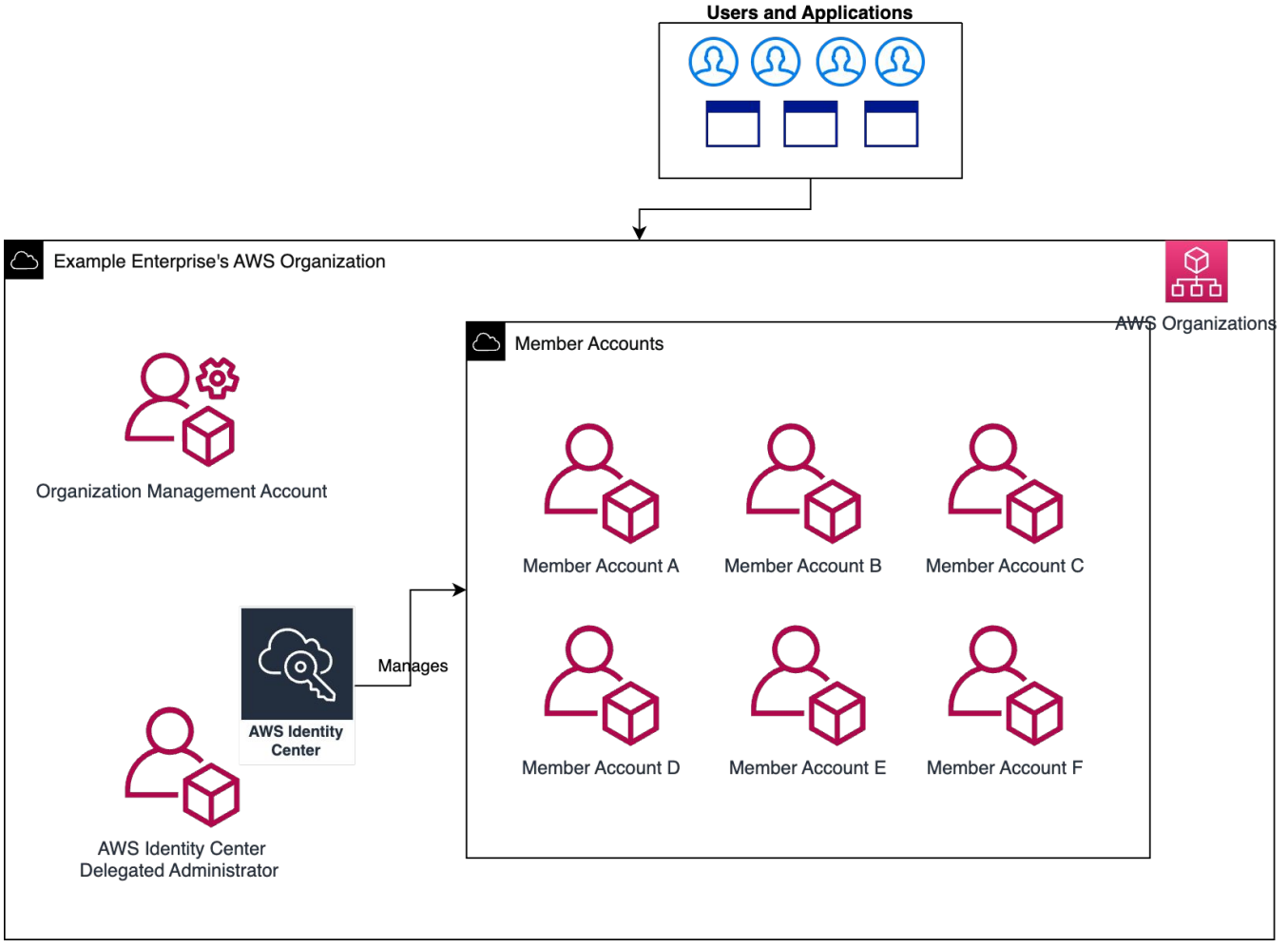
IAM Risks: Role Chaining



Cloud Service Risks: Resource Sharing



Cloud Service and IAM Risks: Identity Center (SSO)



Cloud Service and IAM Risks: Identity Center (SSO)

Permission Set Modification

- `sso:ProvisionPermissionSet`

Adding Permissions

- `sso:AttachManagedPolicyToPermissionSet`
- `sso:AttachCustomerManagedPolicyReferenceToPermissionSet`
- `sso:PutInlinePolicyToPermissionSet`
- `sso:DetachManagedPolicyFromPermissionSet`
- `sso:DetachCustomerManagedPolicyReferenceFromPermissionSet`
- `sso:DeleteInlinePolicyFromPermissionSet`
- `sso:DeletePermissionBoundaryFromPermissionSet`

Cloud Service and IAM Risks: Identity Center (SSO)

Scope Change via Account Assignment

- `sso:CreateAccountAssignment`

Membership Modification

- `identitystore:CreateGroupMembership`
- `sso-directory:AddMemberToGroup`

Cloud Service and IAM Risks: Identity Center (SSO)

Permission Modification
- `sso:PutPermissionBoundaryToPermissionSet`
- Permissions from Adding Permissions

Access Disruption
- `sso:DeletePermissionSet`
- `sso:DeleteAccountAssignment`

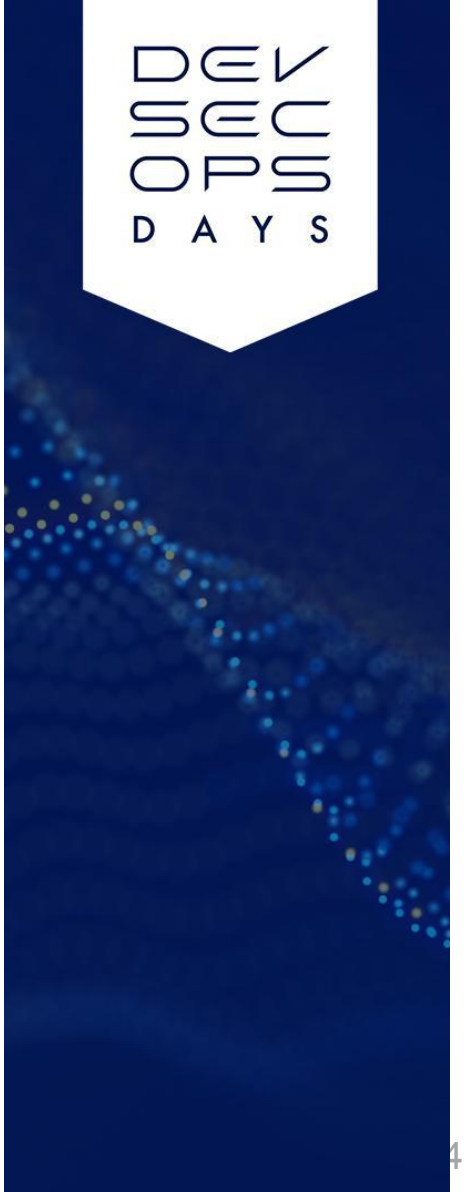
Reconnaissance: Account Enumeration

```
jkao@jkaombpro ~ % aws organizations list-accounts --profile noaccess

An error occurred (AccessDeniedException) when calling the ListAccounts operation: You don't have permissions to access this resource.

jkao@jkaombpro ~ % aws organizations list-accounts --profile ihaveaccess
{
{
    "Accounts": [
        {
            "Id": "123412341234",
            "Arn": "arn:aws:organizations::123412341234:account/o-mwise12345/123412341234",
            "Email": "managementaccount@company.com",
            "Name": "sample-account-1",
            "Status": "ACTIVE",
            "JoinedMethod": "INVITED",
            "JoinedTimestamp": "2022-01-11T21:29:45.040000-05:00"
        },
        {
            "Id": "432143214321",
            "Arn": "arn:aws:organizations::123412341234:account/o-mwise12345/432143214321",
            "Email": "productionaccount@company.com",
            "Name": "sample-account-2prod",
            "Status": "ACTIVE",
            "JoinedMethod": "CREATED",
            "JoinedTimestamp": "2022-03-05T23:50:16.776000-05:00"
        }
    ]
}
```

Securing Multi-Account Estates

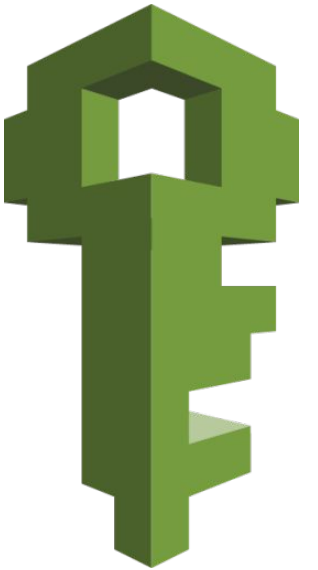


Security Best Practices

- Identity and Access Management Best Practices
- Infrastructure Design & Architecture Best Practices
- Monitoring Best Practices

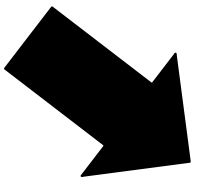
Identity and Access Management Best Practices

- Reduce blast radius
- Eliminate/reduce role chaining
- Setting boundaries to reduce overprivilege
- Least privilege
- Reduce unnecessary access



IAM Role Trust Policies

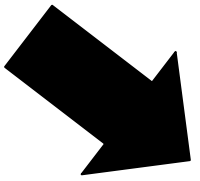
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123412341234:root"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```



```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123412341234:role/my-first-role"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

IAM Permissions

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

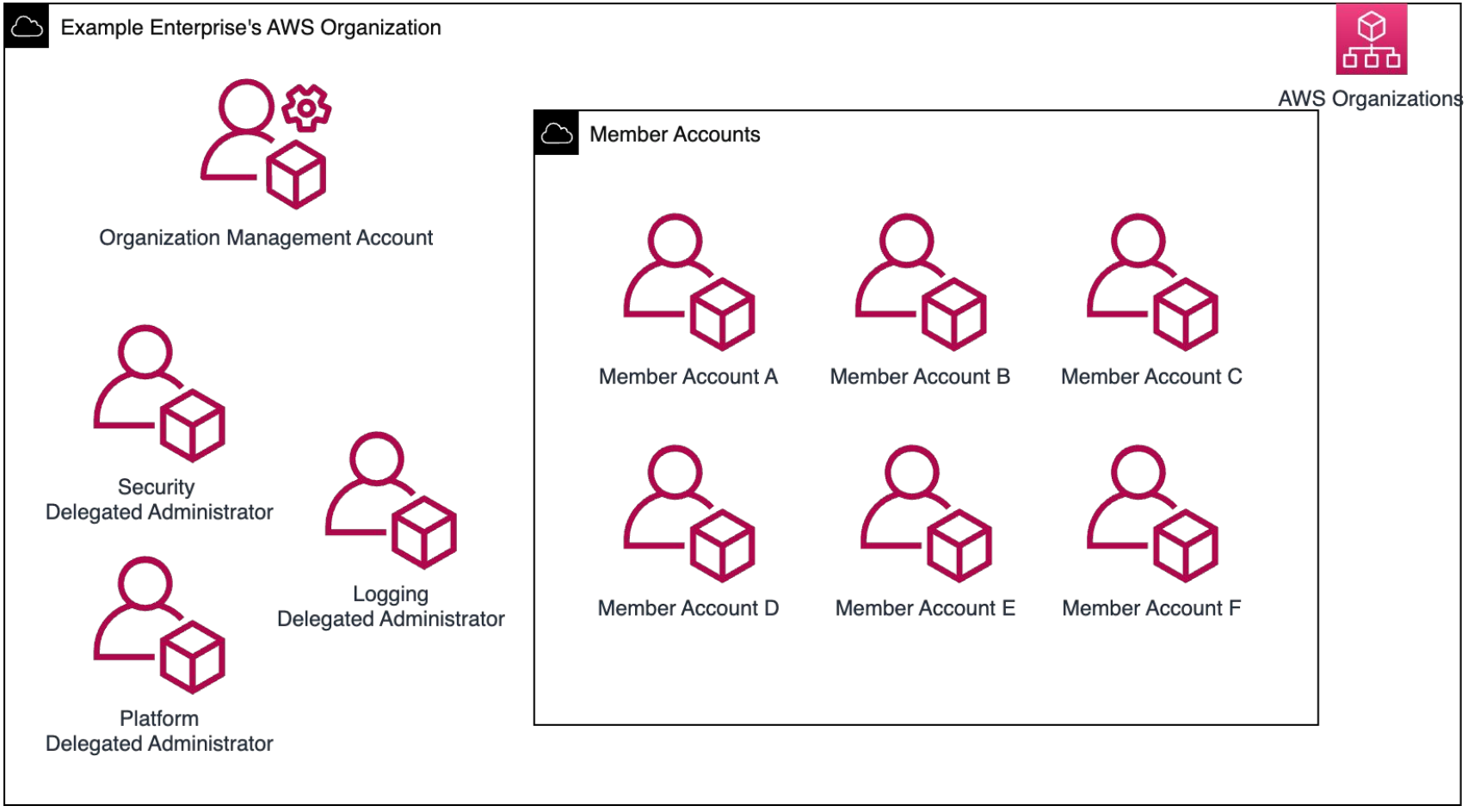


```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "sts:AssumeRole",
        "Resource": [
            "arn:aws:iam::123412341234:role/example-role"
        ]
    }
}
```

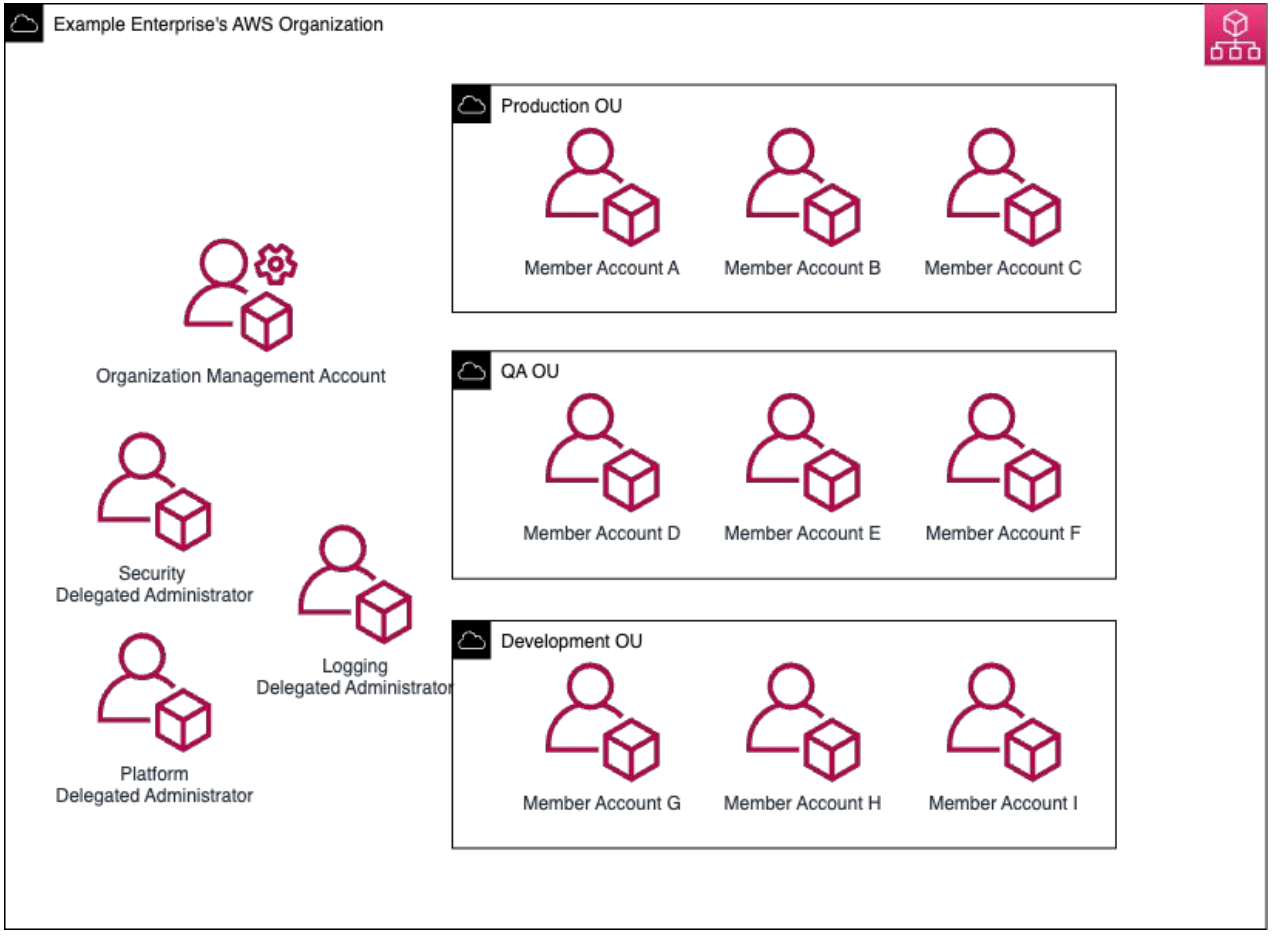
Infrastructure Design and Architecture Best Practices

- AWS Organizations and Multiple Accounts
- Reduce Usage of the Management Account
  - SCPs do not apply to the Management Account
  - Created Member Accounts come with `OrganizationalAccountAccessRole`
- Leverage Delegated Administrator Accounts
- Reduce Direct Access to Delegated Administrator and Management Accounts

Infrastructure Best Practices: Delegation



Infrastructure Best Practices: Separation



Monitoring and Logging Best Practices

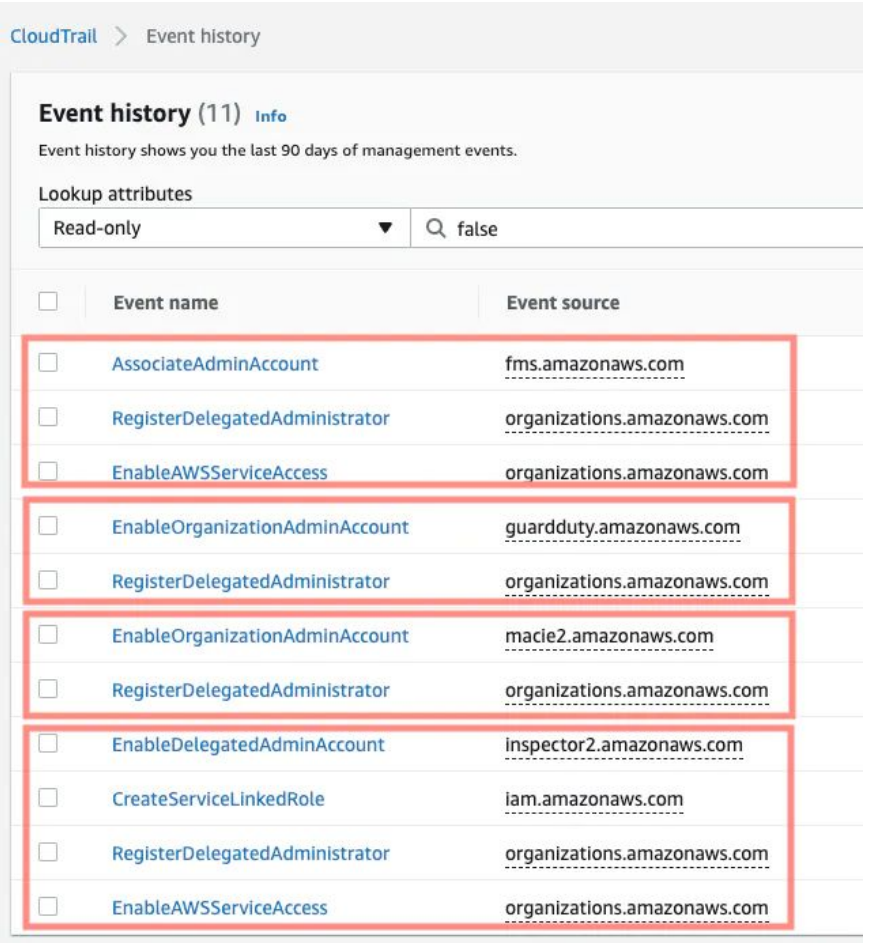
- Maintain Chain of Access within Cloud Environment
  - Linking Cross-Account Sessions Together
  - Tagging Sessions
- Monitor Identity Center Changes
- Monitor Account & Organization Changes

Monitoring via Tagging: Transitive Session Tags

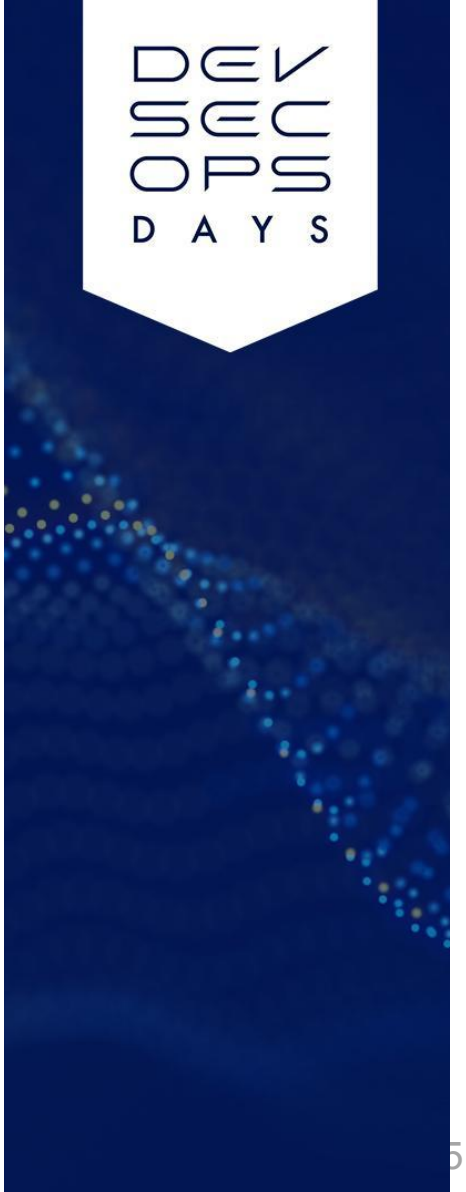
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAssumeRole",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Principal": {"AWS": "arn:aws:iam::123412341234:role/my-first-role"},
            "Condition": {
              "StringLike": {
                  "aws:RequestTag/Project": "*"
              }
            }
        },
        {
            "Sid": "AllowTag",
            "Effect": "Allow",
            "Action": "sts:TagSession",
            "Principal": {"AWS": "arn:aws:iam::123412341234:role/my-first-role"},
            "Condition": {
                "StringLike": {
                    "aws:RequestTag/Project": "*"
                },
                "ForAllValues:StringEquals": {
                    "sts:TransitiveTagKeys": [
                        "Project"
                    ]
                }
            }
        }
    ]
}
```

```
aws sts assume-role \
--role-arn arn:aws:iam::123412341234:role/my-first-role \
--role-session-name named-session \
--tags Key=MyKey,Value=HereIsTheKey \
--transitive-tag-keys MyKey
```

Monitoring: AWS Management APIs via CloudTrail



Platform Enablement and DevSecOps



Security Controls

Prevent

Guardrails

Goal: Prevent an event or misconfiguration from occurring.

- IAM Guardrails
- Account Design and Architecture
- Developer Tools in the Development Lifecycle
- Secure Configuration

Detect

Monitoring and Logging

Goal: Detect, log, and alert after an event has occurred.

- Monitoring and Logging
- Misconfiguration Scans
- Alerting from security controls or monitoring.

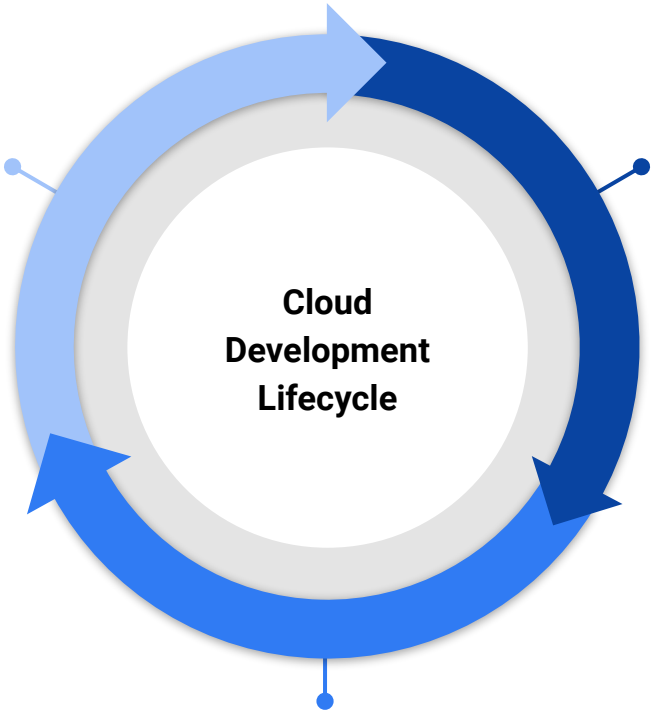
Respond

Remediation

Goal: Remediate adverse events or fix issues after they occur or are detected.

- Remediation Controls
- Automated Responses to issues.
- Runbooks for security incidents.
- Auto Remediation (Ex: Public S3 Buckets).

Controls: Development Lifecycle



Development

Empower Developers with Cloud (IAM, Accounts, etc).

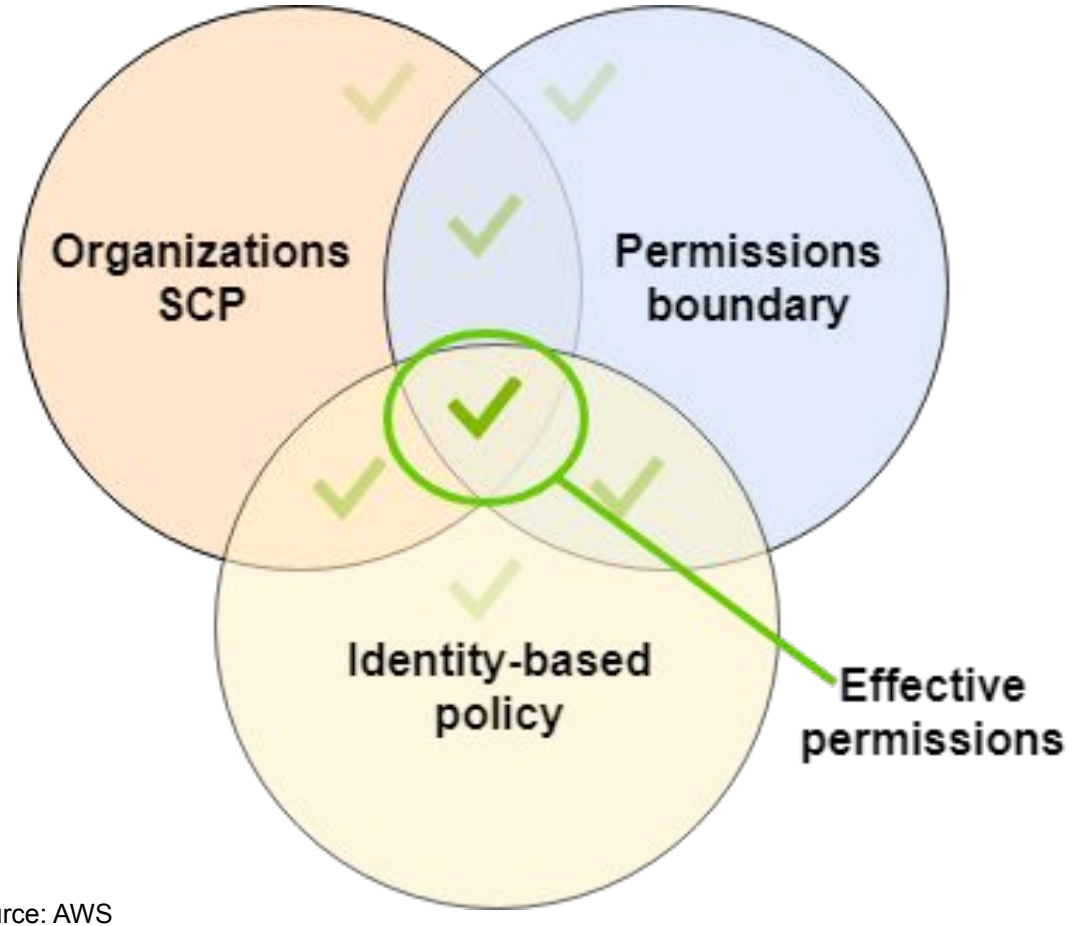
Testing & Controls

Provide Tooling to help with Cloud Development. Testing IAM, Accounts, etc.

Guardrails & Feedback

Provide Boundaries for Developers and take feedback to adjust as needed.

Guardrails: IAM Boundaries



Source: AWS

Guardrails: IAM Deny Policies

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyIdentityCenterManagement",
            "Effect": "Deny",
            "Action": [
                "sso:ProvisionPermissionSet",
                "sso:DeletePermissionSet",
                "sso:DeletePermissionsPolicy",
                "sso:DeletePermissionsBoundaryFromPermissionSet",
                "sso:CreateAccountAssignment",
                "sso:DeleteInlinePolicyFromPermissionSet",
                "sso:PutInlinePolicyToPermissionSet",
                "sso:DeleteAccountAssignment",
                "sso:DetachCustomerManagedPolicyReferenceFromPermissionSet",
                "sso:DetachManagedPolicyFromPermissionSet",
                "sso:AttachManagedPolicyToPermissionSet",
                "sso:CreatePermissionSet",
                "sso:UpdatePermissionSet",
                "sso:AttachCustomerManagedPolicyReferenceToPermissionSet",
                "sso:PutPermissionsPolicy",
                "sso:PutPermissionsBoundaryToPermissionSet",
                "identitystore:CreateGroupMembership",
                "sso-directory:AddMemberToGroup"
            ],
            "Resource": "*"
        }
    ]
}
```

Conclusion

- Multiple Accounts is the way.
- Reduce the possibility of risky multi-account IAM combinations.
- Empower Developers by providing secure guardrails via:
  - IAM Boundaries
  - Secure Account Architecture
- Fortify Detective Controls
  - Monitor and Maintain Chain of Access.
  - Sensitive Cloud Services

Contact Me

Jason Kao
Head of Security Research and Solutions, CloudQuery

Email: jason@cloudquery.io

CloudQuery

CloudQuery.io

https://www.linkedin.com/in/kaojason/