Practical guidance to get to least privilege IAM at scale

Cassia Martin
Sr. Security SA

Liam Wadman
Security Solutions Architect

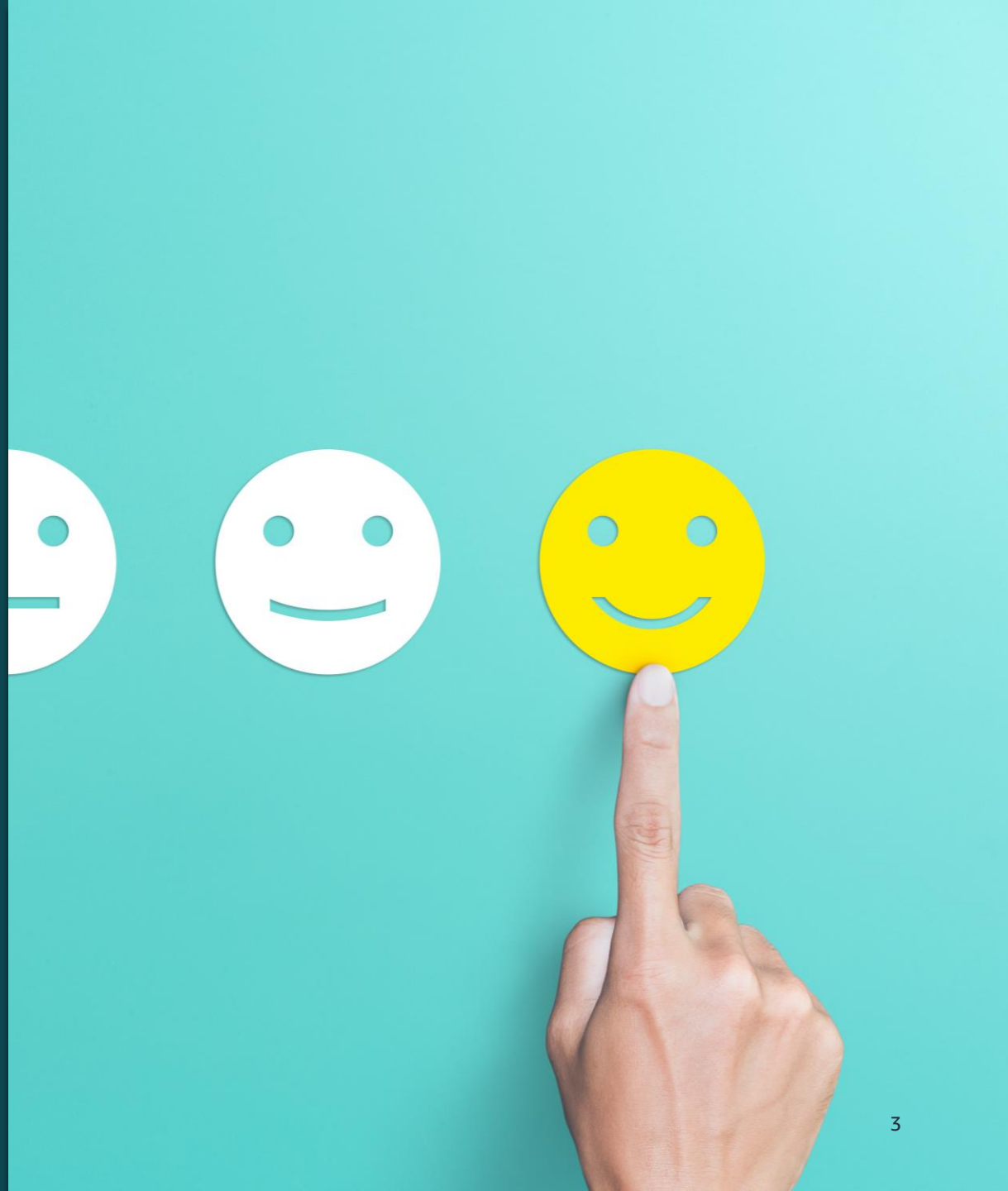
Agenda

- What problem are we trying to solve?
- Where do we start?
  - How do we set up my accounts and guardrails?
- How do we get better every day?
  - What tools are there to help?

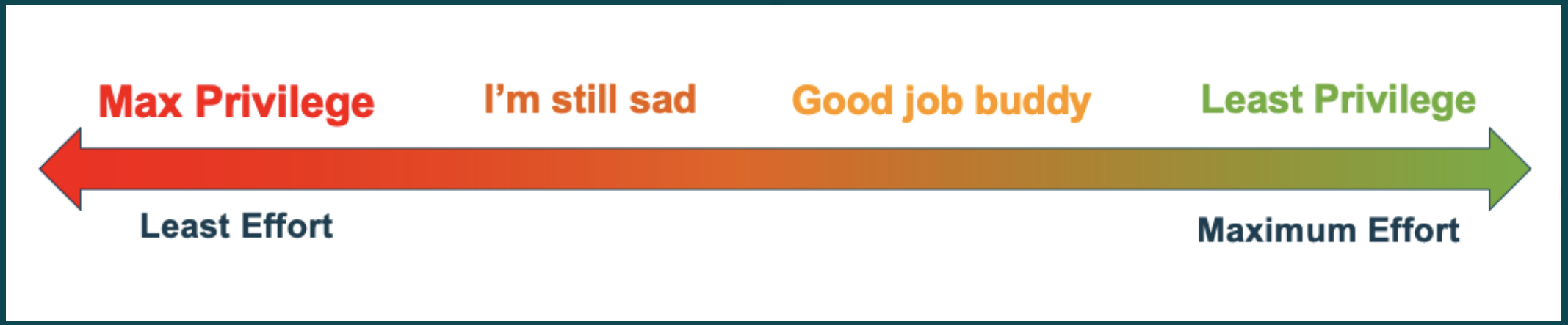
What is ”Least Privilege”?

Ensure that a principal only has the permissions required to perform the desired actions

- Humans
  - that build our software across the various environments they access
- Systems
  - that we build that need to call other systems & APIs



Is least privilege a binary choice?



Max Privilege

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

Absolute Least Privilege

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
  "Effect": "Allow",
    "Action": [
        "ec2:AttachVolume",
        "ec2:DetachVolume"
    ],
    "Resource": [
       "arn:aws:ec2:us-east-1:012345678901:volume/vol-11111111
       "arn:aws:ec2:us-east-1:012345678901:volume/vol-22222222
       "arn:aws:ec2:us-east-1:012345678901:volume/vol-33333333
       "arn:aws:ec2:us-east-1:012345678901:volume/vol-44444444
       "arn:aws:ec2:us-east-1:012345678901:instance/i-11111111
       "arn:aws:ec2:us-east-1:012345678901:instance/i-2222222”
       "arn:aws:ec2:us-east-1:012345678901:instance/i-33333333
    ],
    "Condition": {
      "ArnEquals": {"ec2:SourceInstanceARN": "arn:aws:ec2:us-e
1:012345678901:instance/i-33333333"}
            }
        }
```

Practical least Privilege

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachVolume",
                "ec2:DetachVolume"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:volume/*",
                "arn:aws:ec2:*:*:instance/*"
            ],
            "Condition": {
                "ArnEquals":
                  {"ec2:SourceInstanceARN":
                  "arn:aws:ec2:*:123456789012:instance/i-*"}
            }
        }
    ]
}
```

What does a good policy look like?

- For human accounts
  - Enough permission to build
  - Not able to change the security posture of the environment
  - Not too constrained so that requests for additional permissions are frequent
  - Context aware – different in each environment
- For system roles
  - Repeatable and precise
  - Have permissions boundaries attached
  - Avoid * in policies - bound to specific resources

Multiple people are involved in this journey

Developer Team

Make good decisions at the point of development

Platform Team

Provide capabilities for early feedback & visibility

Security Team

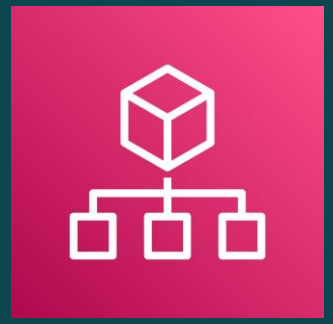
Communicate the expectations & enable the good decisions

How do we map this out?

- Build a secure environment
  - Separate accounts
  - Federate Identity
  - Construct guardrails
- Iterate over time
  - Allow experimentation
  - Use tools to reduce unused access
  - Build automation!

Build a Secure Foundation

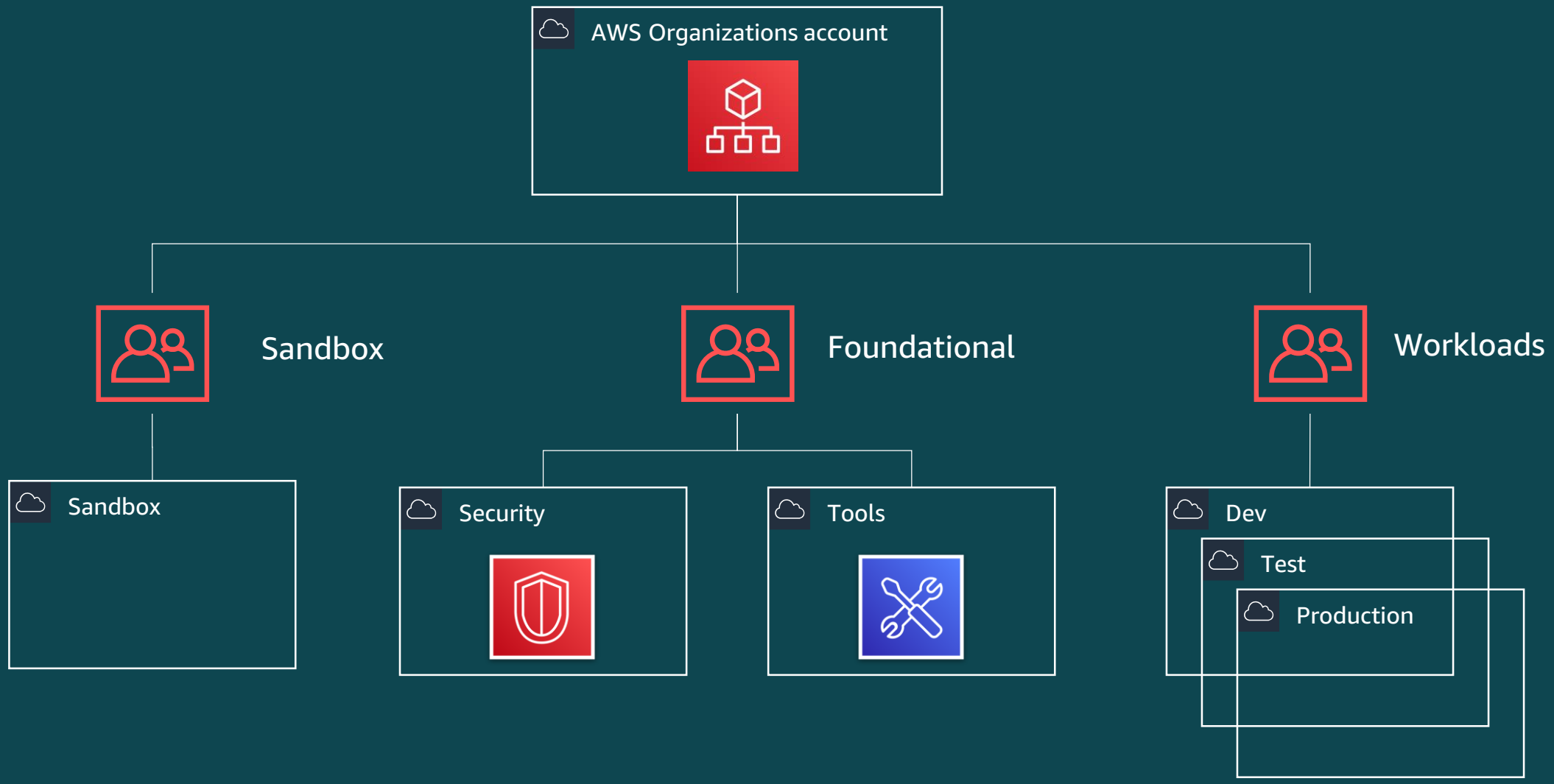
Federate Identity



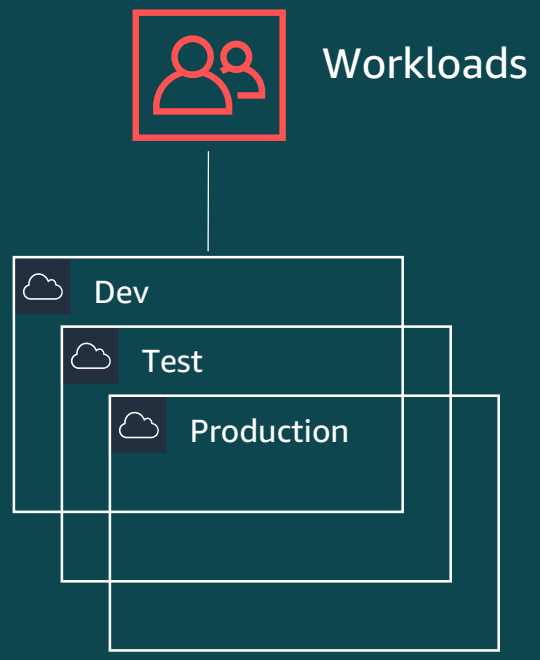
Separate Accounts

Construct guardrails

Separation of duty – Multi-account strategy



Workload least privilege - Context is important

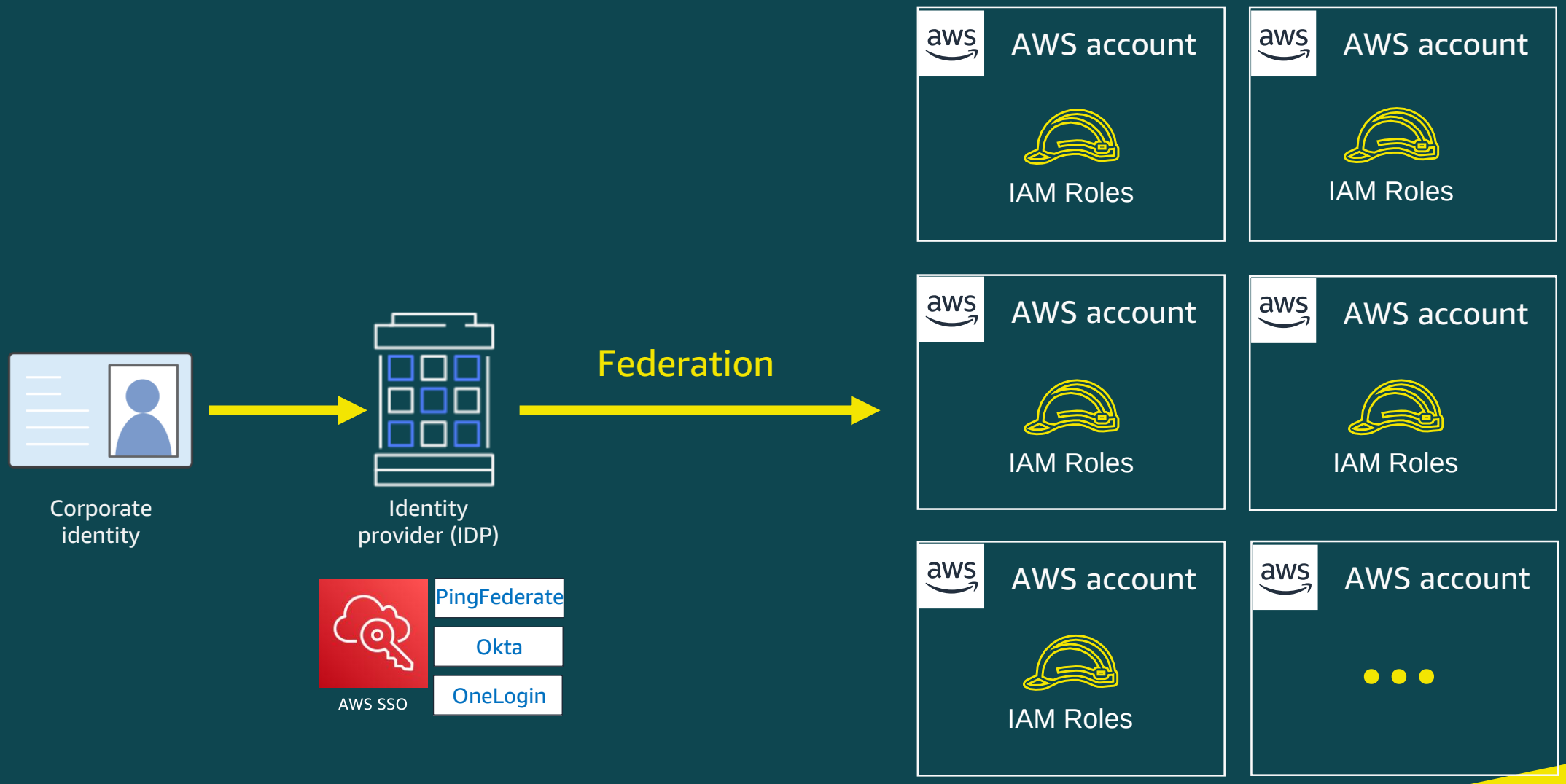


Dev – Least privilege means freedom to build and troubleshoot

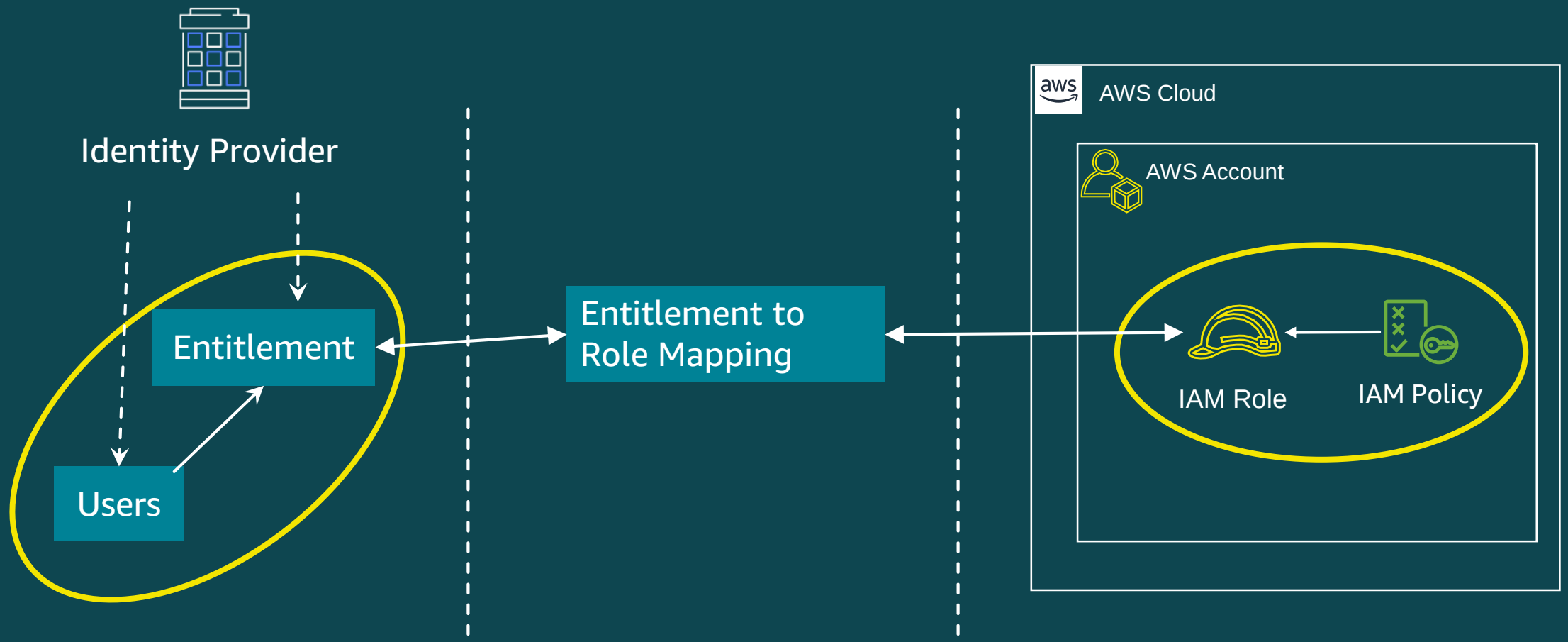
Test – Least privilege means Infrastructure as Code and automated feedback

Production - Least privilege means no human access to change the environment or access data

Federate identity for human users



Federation – how to do mapping

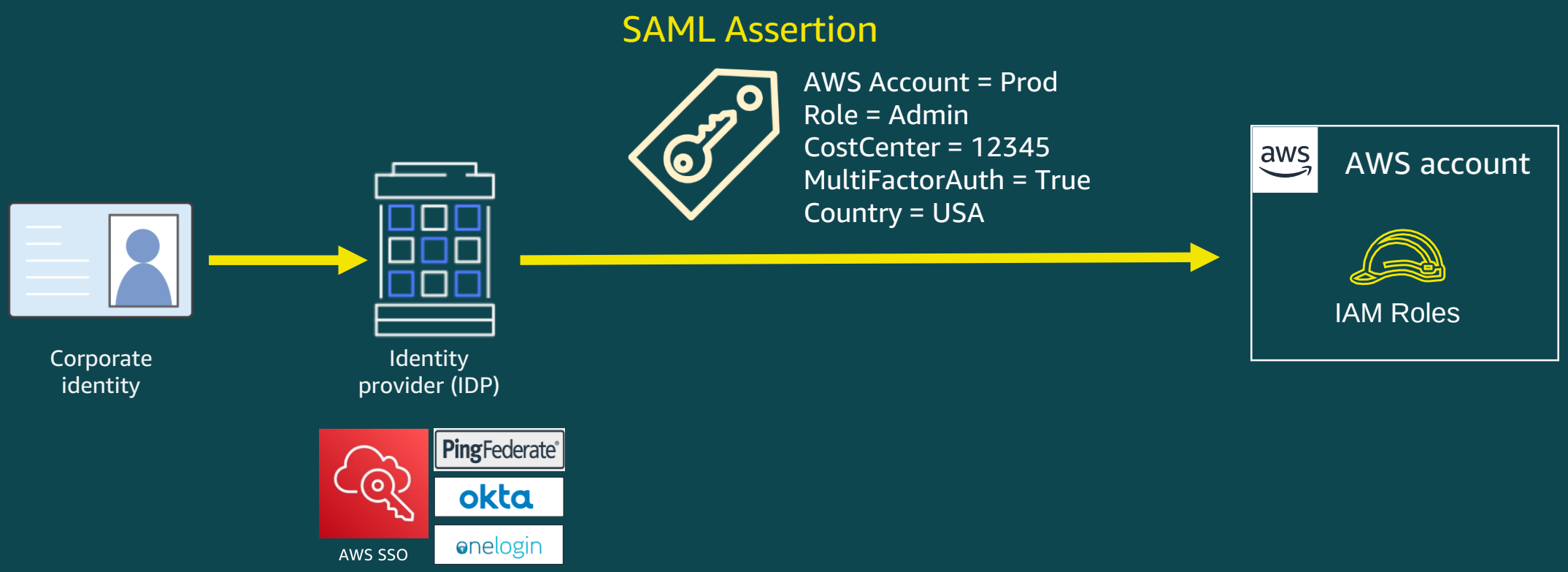


Who can use the role

Which entitlement corresponds to which IAM Role in which account

What the role can do

Federating beyond just identity



Build Guardrails

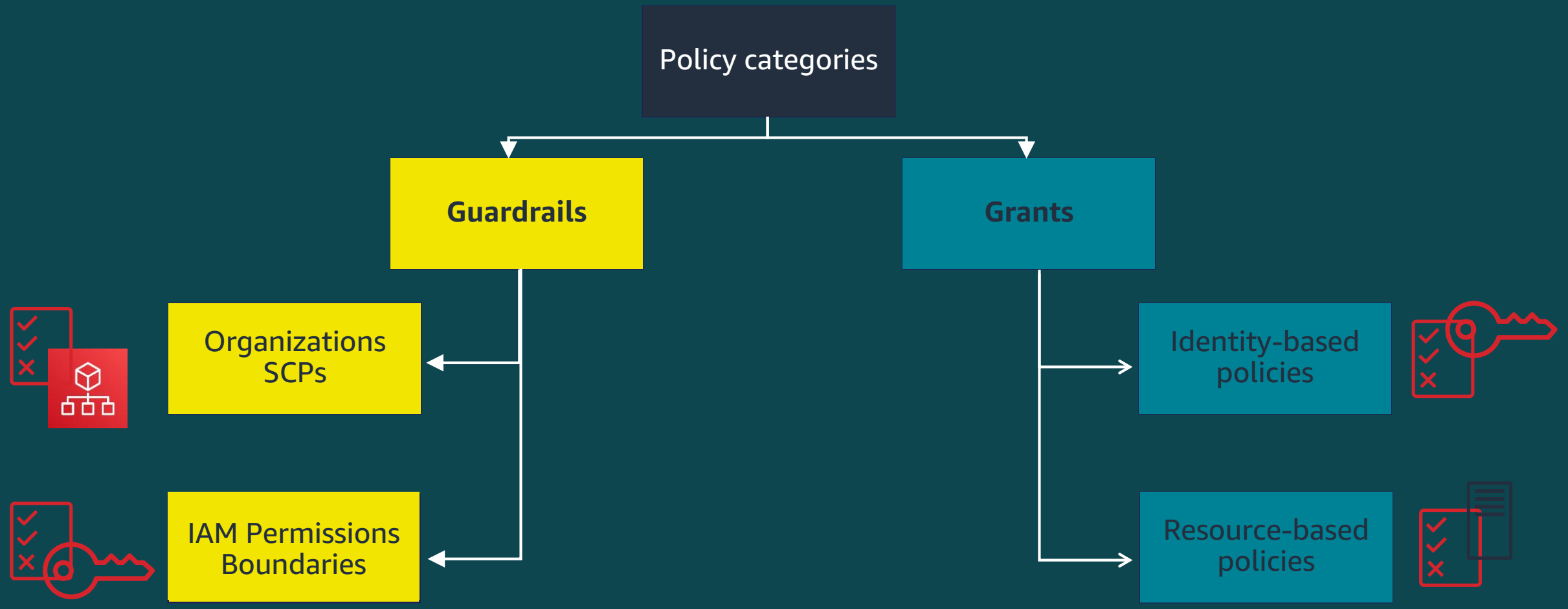
- Service control policies
  - What is an invariant?
- Identity policies
  - What am I granting?
- Permissions boundaries
  - What can delegated folks do?
- Resource policies
  - What do data owners grant?

Policies in AWS

Policies that set the maximum permission

Policies that give permission

Policies in AWS



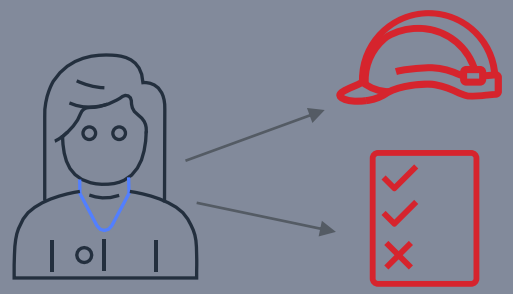
SCP guardrails

Service Control Policies (SCP) are applied account-wide. They cannot be superseded by <u>any</u> grant in an account to which they apply.

- Anything that is common across all accounts should be in an SCP, for example:
  - Restrict access to specific AWS Regions
  - Prevent deletion of common resources
  - Prevent non admins from disabling logging, monitoring
  - Prevent changes to S3 block public access
  - Restrict use of unapproved services
  - Restrict internet gateways from being attached where not authorized

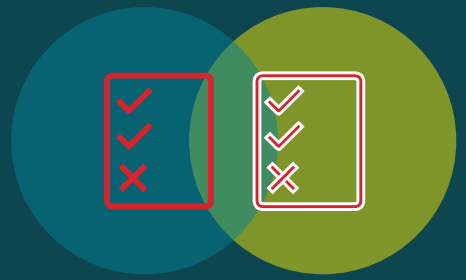
Permission Boundaries

Grant Alice the permission to create policies and roles

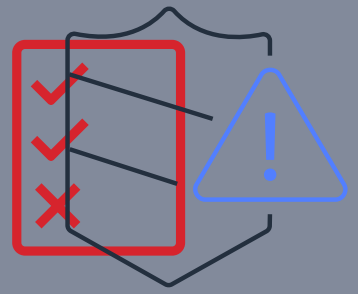


Require that another policy (permissions boundary) is also attached to the role

The effective permission of the role is the intersection of the two policies



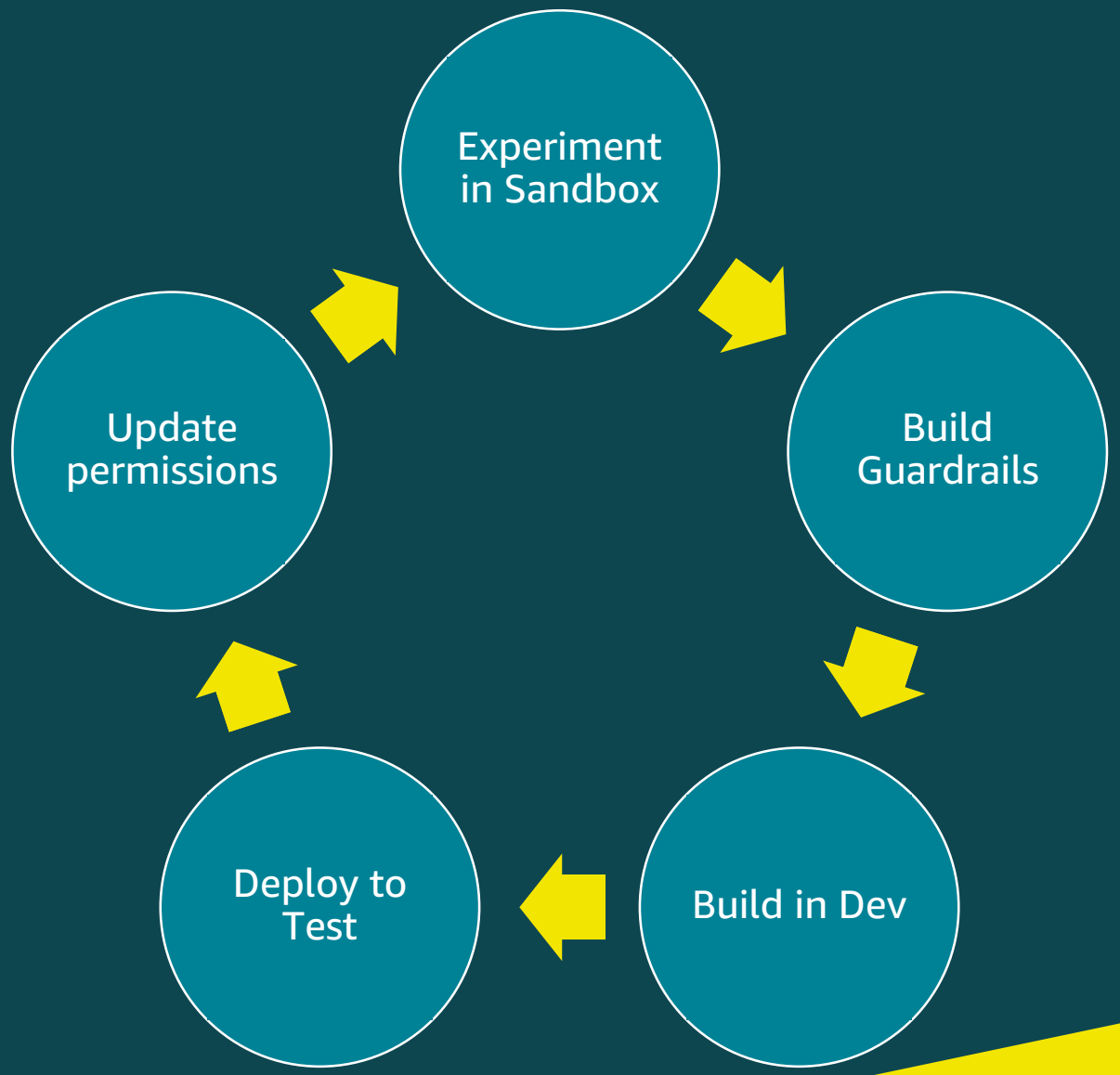
In this way, you can set the maximum permission of the roles that Alice creates—the roles are bound.



Iterate over time

- Iterate over time
  - Allow experimentation
  - Use tools to reduce unused access
  - Build automation!

- Least privilege for the appropriate context



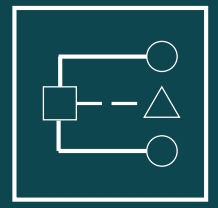
So what is the lifecycle of least privilege?

- Builder looks at a service they've not used before
  - In sandbox account
- Building workloads in dev with policies attached to roles
  - Interactive – make calls to IAM Access Analyzer or use console
  - Via pipeline – automated checks
- Deploy to test & run the workload for some time
  - Validate access based on activity
- Perform access review
  - Update permissions based on policy recommendation

Tools to help you make good decisions

IAM Features

AWS Config



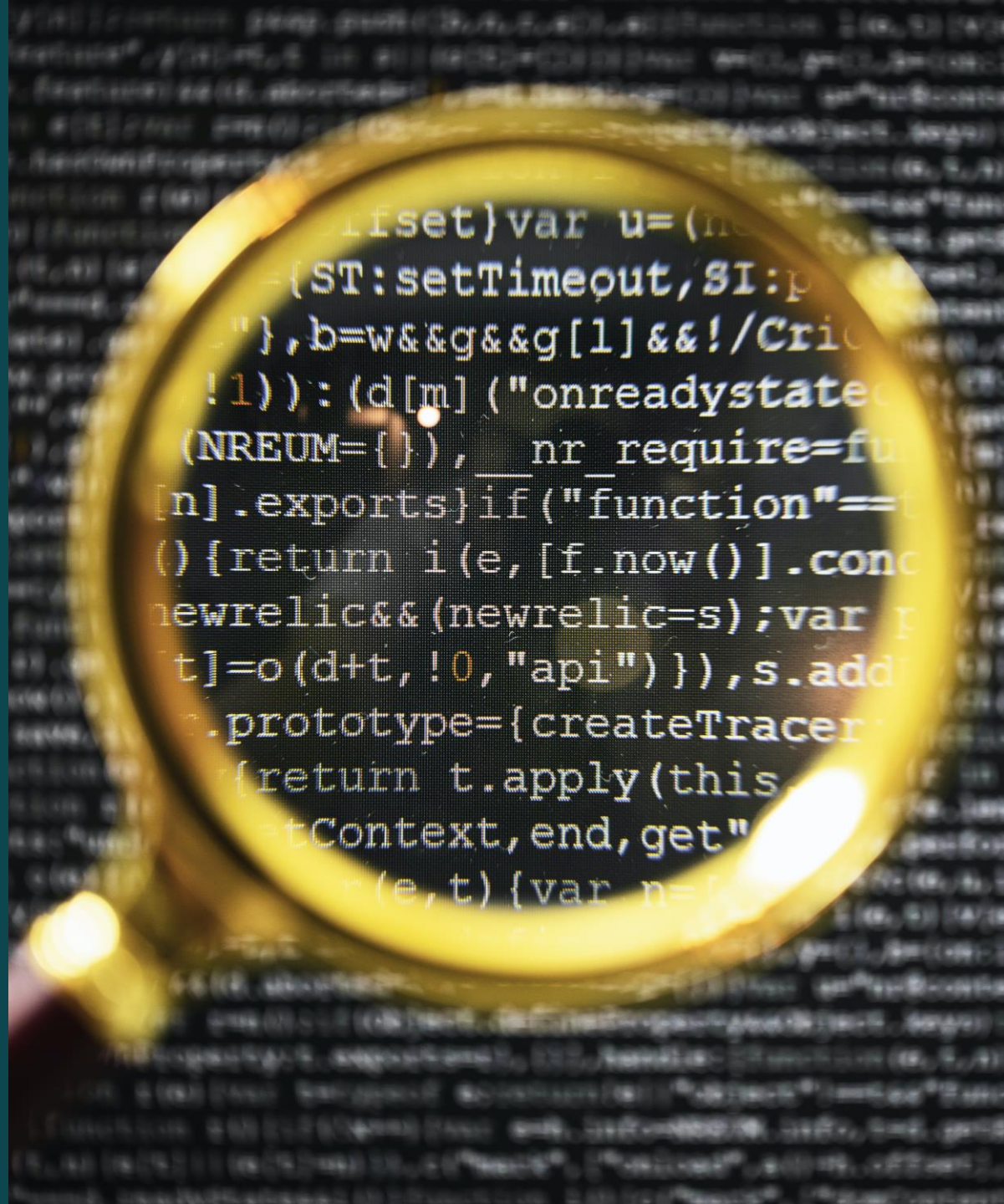
IAM Access Analyzer

IAM Policy Generator
IAM Policy Simulator
IAM Access Advisor

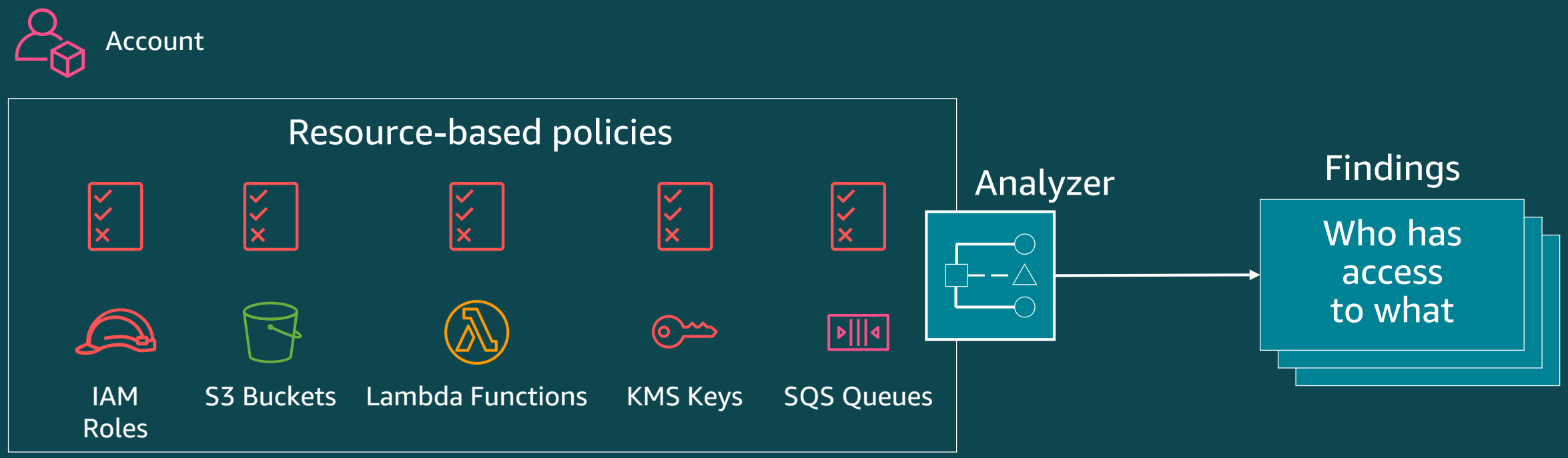
Security Hub

IAM Access Analyzer

- Console
  - IAM access analyzer policy analyzer
- Programmatic
  - AWS access analyzer validate-policy
  - Integrated in pipeline or commit hook



IAM Access Analyzer - Findings



IAM Access Analyzer – Policy Validation

Visual editor | JSON | Import managed policy

```
1  {
2      "Version": "2012-10-17",
3      "Statement": [{
4          "Effect": "Allow",
5          "Action": [
6              "iam:GetRole",
7              "iam:PassRole"
8          ],
9          "Resource": "*"
10     }]
11 }
```

Security: 1 | Errors: 0 | Warnings: 0 | Suggestions: 0

Search security warnings | Learn more | Feedback

Ln 7, Col 12

PassRole With Star In Resource: Using the iam:PassRole action with wildcards (*) in the resource can be overly permissive because it allows iam:PassRole permissions on multiple resources. We recommend that you specify resource ARNs or add the iam:PassedToService condition key to your statement. Learn more

IAM A

Ger

Generat

Time

Select

Start d

2021

End da

Generated policy

Review and create managed policy

Review the permissions summary, add tags, and create the generated policy as a customer managed policy

Name*

TestRolePolicy

Use alphanumeric and '+=,.@-_' characters. Maximum 128 characters.

Description

Sample Policy Generated

Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

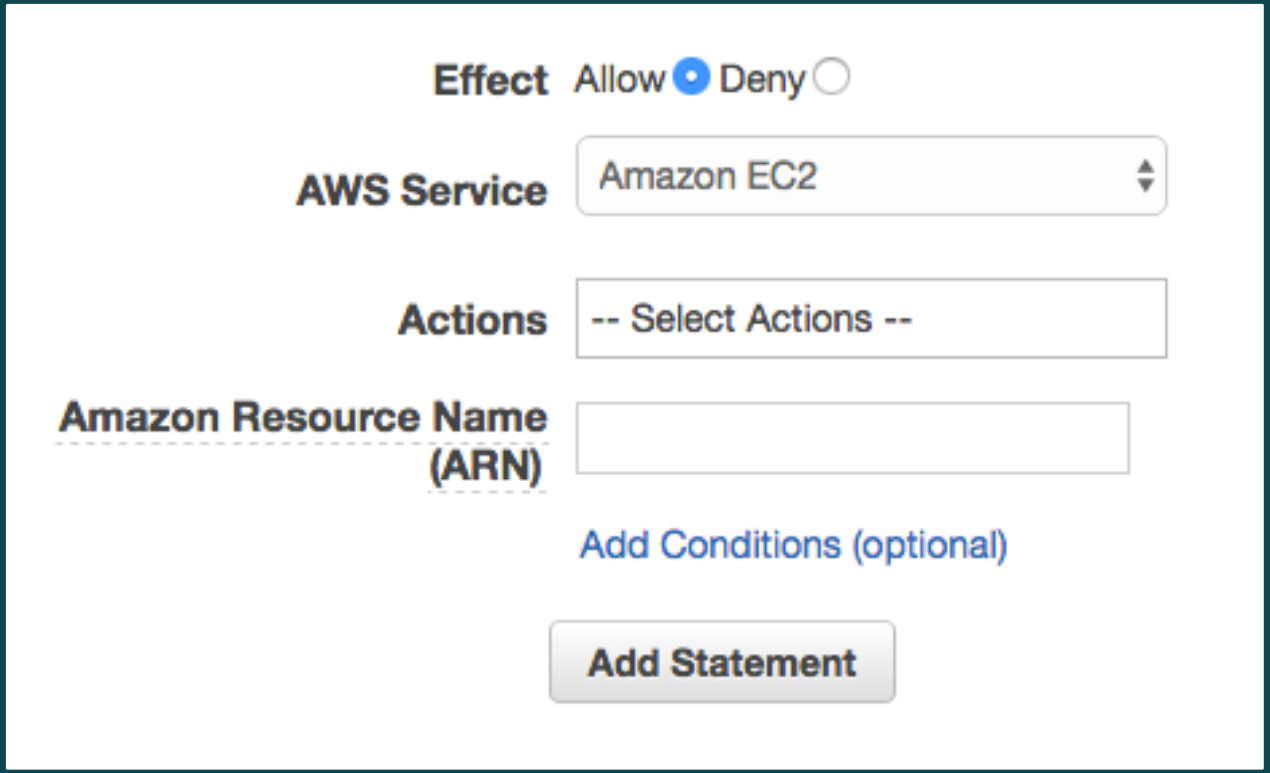
Summary

Filter

Service	Access level	Resource	Request condition
Allow (5 of 276 services) Show remaining 271			
EC2	**Limited**: List, Read, Write	All resources	None
IAM	**Limited**: List, Read, Write, Permissions management	All resources	None
Lambda	**Limited**: List, Read	All resources	None
S3	**Limited**: Write	All resources	None
SQS	**Limited**: Read, Write	All resources	None

IAM Policy Generator

The AWS Policy Generator has a simple GUI that helps you build your IAM and Resource policies.



https://awspolicygen.s3.amazonaws.com/policygen.html

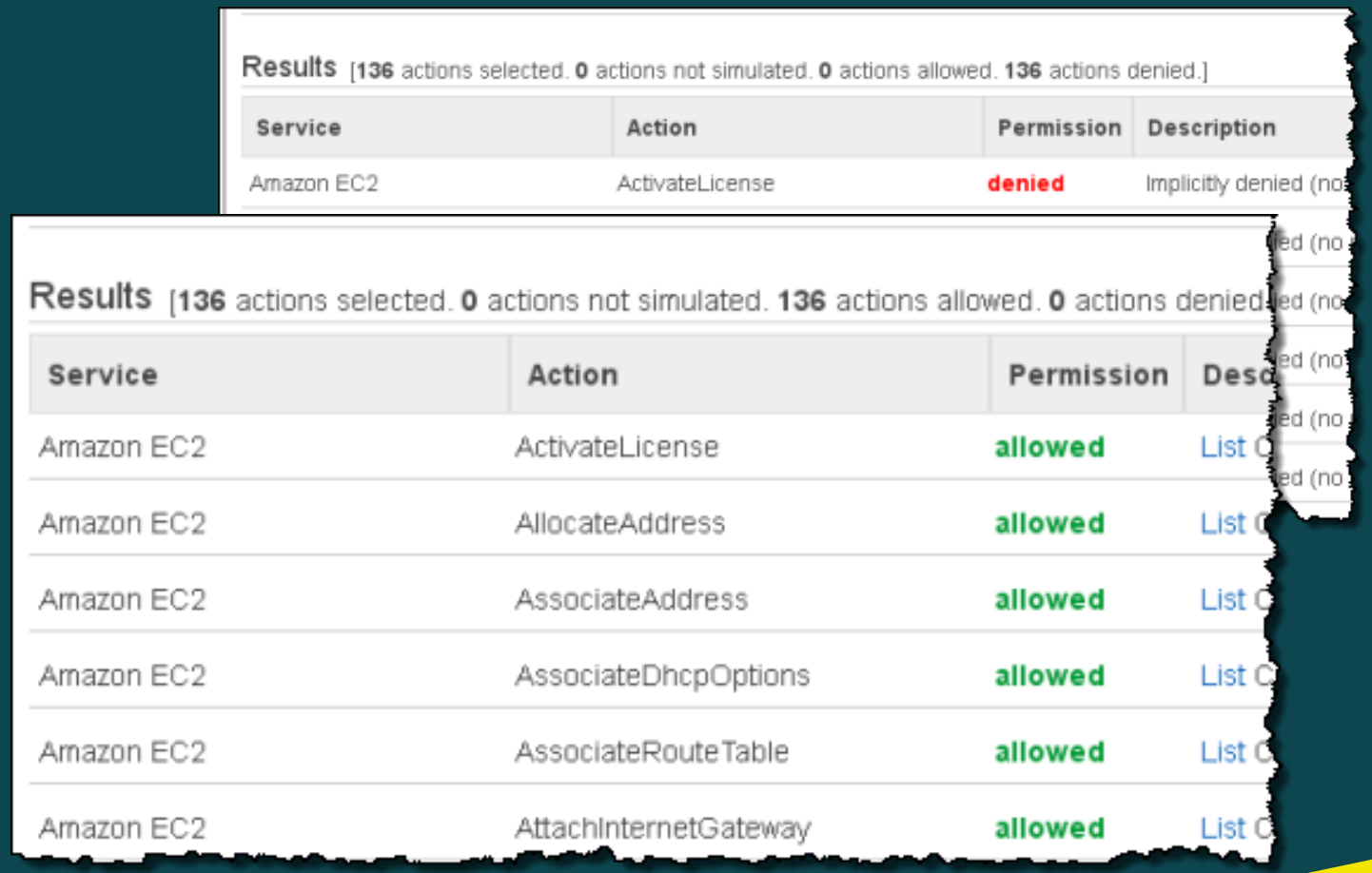
IAM Policy Generator - JSON

https://awspolicygen.s3.amazonaws.com/js/policies.js

```
....},
        "AWS Artifact": {
        "ARNFormat": "arn:aws:artifact::<resource>",
        "ARNRegex": "^arn:aws:artifact::.+",
        "Actions": [
                "AcceptAgreement",
                "DownloadAgreement",
                "Get",
                "TerminateAgreement"
        ],
        "HasResource": true,
        "StringPrefix": "artifact"
}, ....
```

IAM Policy Simulator

The IAM policy simulator allows you to test policies against resources in your account



Results [136 actions selected. 0 actions not simulated. 0 actions allowed. 136 actions denied.]

Service	Action	Permission	Description
Amazon EC2	ActivateLicense	denied	Implicitly denied (no

Results [136 actions selected. 0 actions not simulated. 136 actions allowed. 0 actions denied.]

Service	Action	Permission	Desc
Amazon EC2	ActivateLicense	allowed	List C
Amazon EC2	AllocateAddress	allowed	List C
Amazon EC2	AssociateAddress	allowed	List C
Amazon EC2	AssociateDhcpOptions	allowed	List C
Amazon EC2	AssociateRouteTable	allowed	List C
Amazon EC2	AttachInternetGateway	allowed	List C

IAM Access Advisor – Policy view

Permissions | Groups | Tags (1) | Security credentials | Access Advisor

Access Advisor shows the services that this user can access and when those services were last accessed. Review this data to remove unused permissions. Learn More

Allowed services (89)

Access Advisor reports activity for services and EC2, IAM, Lambda, and S3 management actions. To view actions, choose the service name from the list. Recent service activity usually appears within 4 hours. Service activity is reported for

Last accessed information is available for EC2, IAM, Lambda, and S3 management actions.

Search | No Filter | 1 2

Service	Policies granting permissions	Last accessed
Amazon S3	ViewOnlyAccess	154 days ago
AWS Certificate Manager	ViewOnlyAccess	Not accessed in the tracking period
Amazon Athena	ViewOnlyAccess	Not accessed in the tracking period
Amazon EC2 Auto Scaling	ViewOnlyAccess	Not accessed in the tracking period
AWS Marketplace	ViewOnlyAccess	Not accessed in the tracking period
AWS Batch	ViewOnlyAccess	Not accessed in the tracking period
Amazon Cloud Directory	ViewOnlyAccess	Not accessed in the tracking period
AWS CloudFormation	ViewOnlyAccess	Not accessed in the tracking period
Amazon CloudFront	ViewOnlyAccess	Not accessed in the tracking period

IAM Access Advisor – management actions drilldown

Allowed management actions for Amazon S3 (61)

Access Advisor reports management action activity that is logged by CloudTrail for this service. Recent activity usually appears within 4 hours. Yo
logged since 4/12/2020. To view all of the role's events, see AWS CloudTrail. Learn More

Search

No Filter

1 2 3 4 5

Action	Last accessed	Region accessed
GetBucketLocation	8 days ago	US East (N. Virginia) us-east-1
ListAllMyBuckets	8 days ago	US East (N. Virginia) us-east-1
DeleteBucketPolicy	8 days ago	US East (N. Virginia) us-east-1
GetBucketVersioning	9 days ago	US East (N. Virginia) us-east-1
GetAccountPublicAccessBlock	9 days ago	US East (N. Virginia) us-east-1
GetBucketAcl	9 days ago	US East (N. Virginia) us-east-1
GetBucketCORS	9 days ago	US East (N. Virginia) us-east-1

Organization Activity Report

Service access report

Review access activity to learn when a principal within the organizational entity last accessed a service. Data is available for services that are allowed by directly attach

Search services

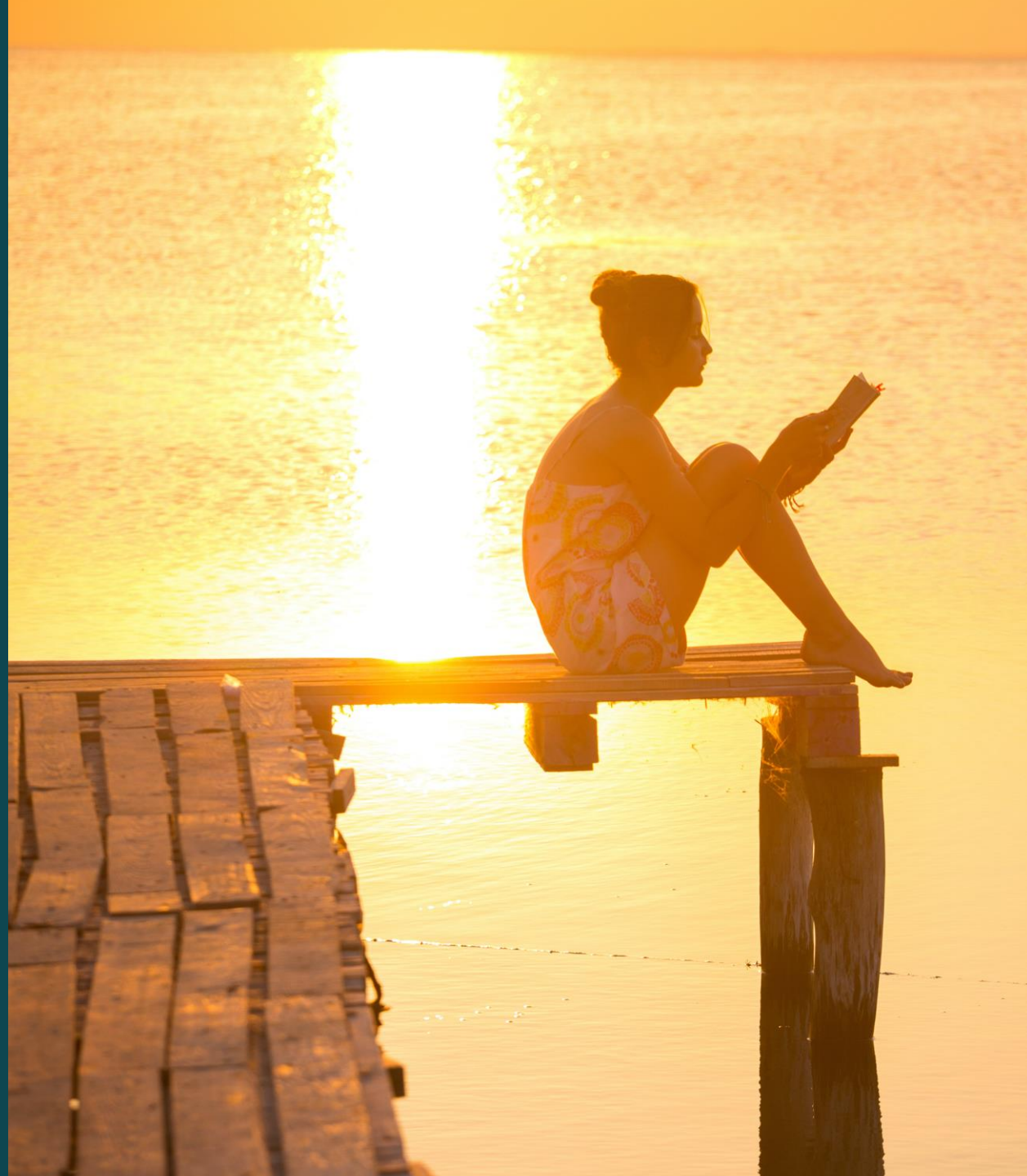
Service	Last accessed
Amazon Message Delivery Service	Today
Amazon CloudWatch Logs	Today
AWS Systems Manager	Today
AWS Resource Groups	Today
Amazon S3	Today
Amazon EC2	Today
Amazon EC2 Auto Scaling	Today
AWS Backup	Today

What if automation is not enough?

- You will need a mechanism to provision access outside the typical
  - Ticketing is your friend
  - Least privilege may mean expanding to get LP for what needs to be done
- Ticket history is data
  - Over time you'll find common needs
  - Helps focus your security program

Principles

- Least privilege is iterative
- You want to balance access for people to do the work with managing risk
- Allow experimentation
- Build Guardrails for invariants
- Automation & tooling helps with analysis



Activities to Support Principles (1/2)

- Use multiple AWS accounts to separate workloads
  - Least Privilege depends on context of AWS Accounts
- Authenticate users centrally & use roles to provide access
- Build identity guardrails in layers
  - SCPs, identity policies, permissions policies, resource policies
  - ABAC to scope down further within accounts

Activities to Support Principles (2/2)

- Use IAM features to verify and generate policies
- Build pipelines for automated checks
  - If you have specific requirements you can manually build checks with lambda
- Ticketing mechanism to get a human who can understand context

Call to action

- If you are not already, make sure you are federating into AWS
- If you are not already, make sure you separate dev/test/prod
- SCPs for invariants/Guardrails
- Use tooling (IAM Access Analyzer etc.) to:
  - Give Fast feedback to devs
  - Generate Data to make your least privilege journey easier

Further Viewing

Becky Weiss	Quint Van Deman	Josh Du Lac	Brigid Johnson
Enforcing Security Invariants youtube.com/watch?v=W30sx0hpY0Y	Mastering Identity at each Layer of the Cake youtube.com/watch?v=vbjFjMNVEpc	Choosing the right mix of AWS IAM policies for scale youtube.com/watch?v=o1bfA0SIxBk	Next Generation Permission Management youtube.com/watch?v=8vsD_aTtuTo

Thank you!