
JUNE 13 - 14, 2023 | ANAHEIM, CA

IAM332-R

# Practical identity and access management: The basics of IAM on AWS

**Dario Goldfarb**
Principal Security Solutions Architect
AWS

**Arthur Mnev**
Senior Security Solutions Architect
AWS

# What you'll learn in this session

In this session you will learn how to build your IAM strategy, with opinionated prescriptive guidance

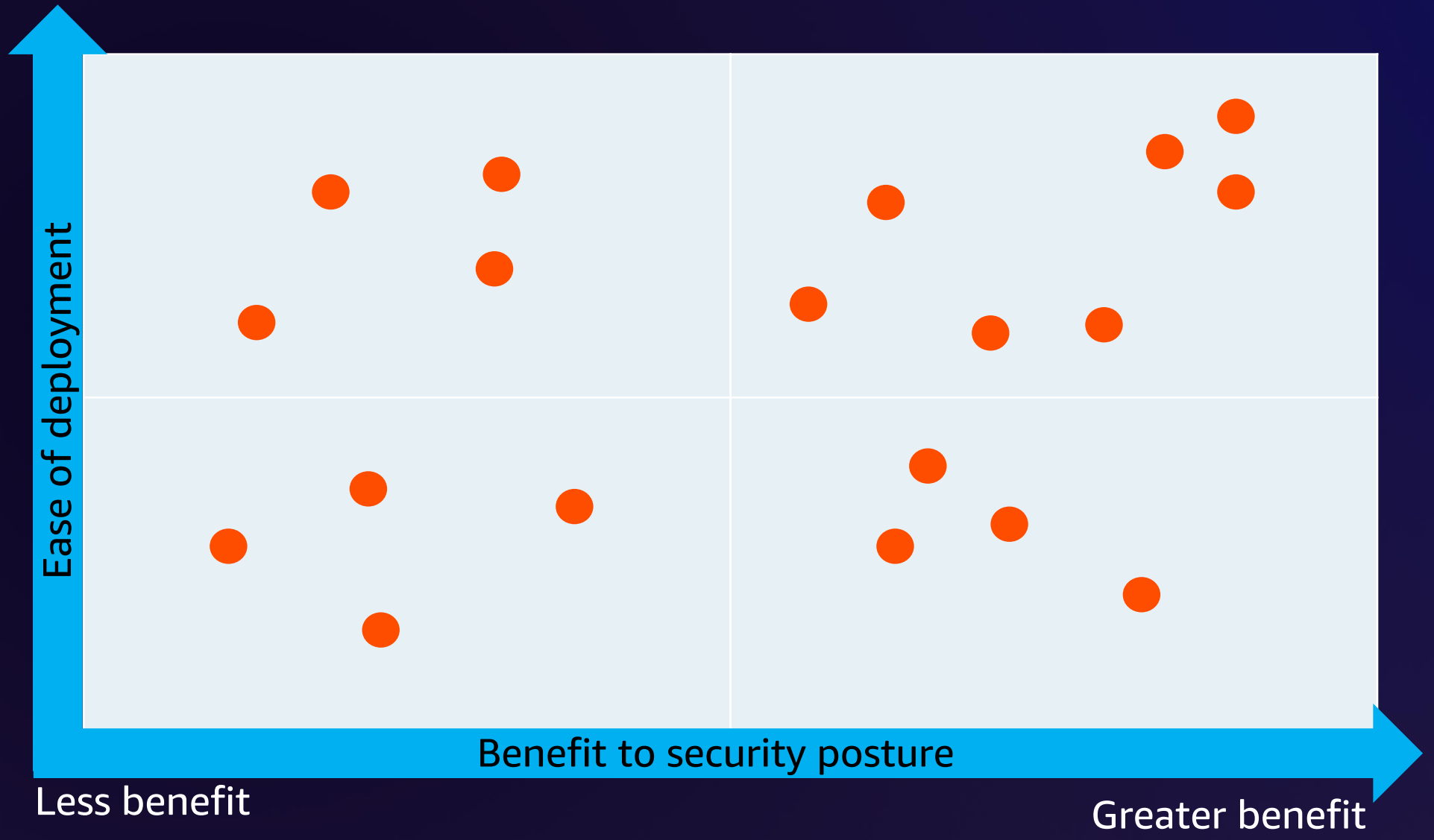
# Agenda

- AWS identity strategy
  - How to prioritize
  - Building your IAM journey
- Practical IAM recommendations for each phase:
  - Quick Wins
  - Foundational
  - Efficient
  - Optimized
- Conclusions

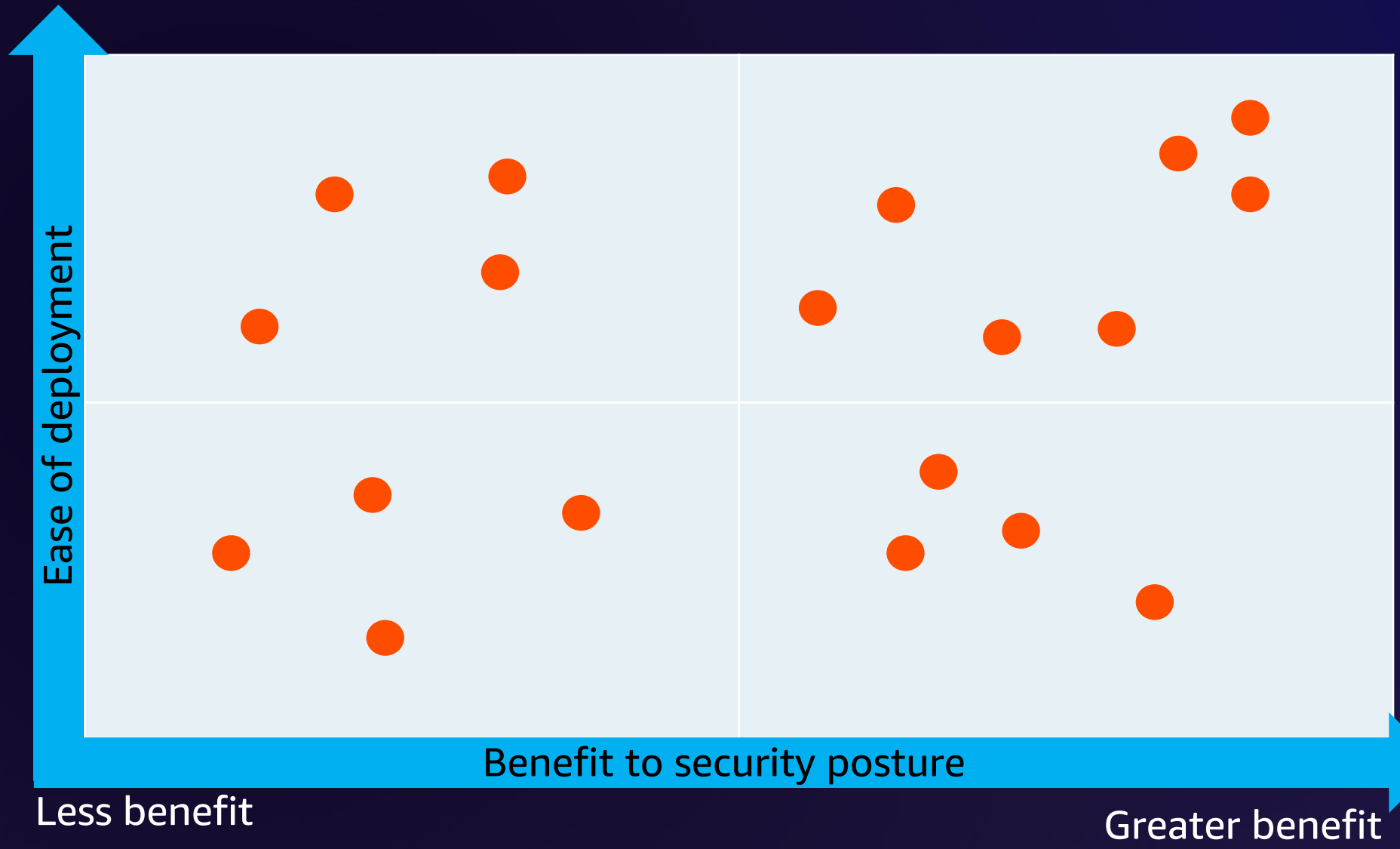
# AWS identity strategy

# Improving your posture is a journey

# AWS identity strategy: How to prioritize

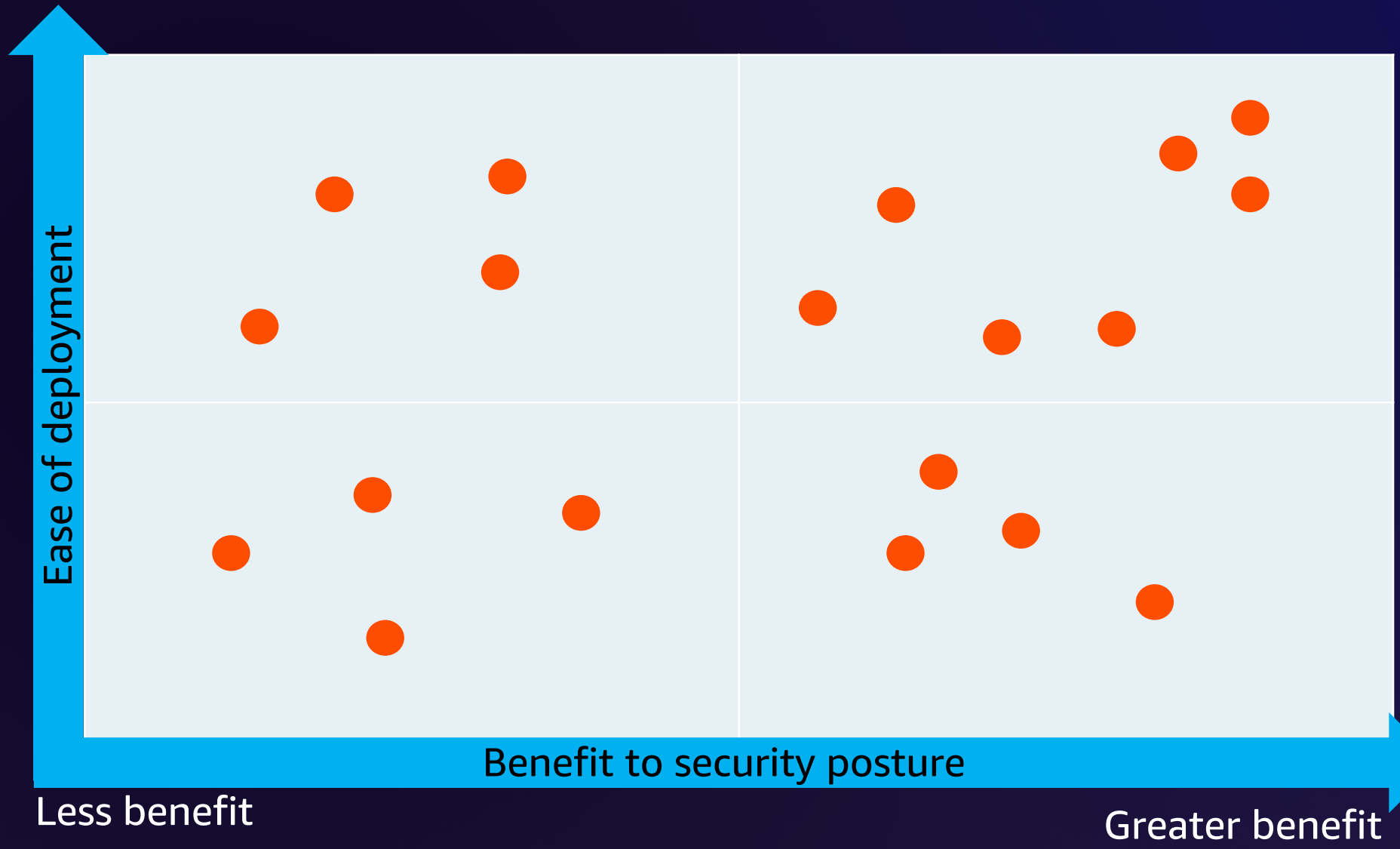


# AWS identity strategy: How to prioritize




**Easy**
1. Quick/easy to implement
2. Low effort
3. Low cost

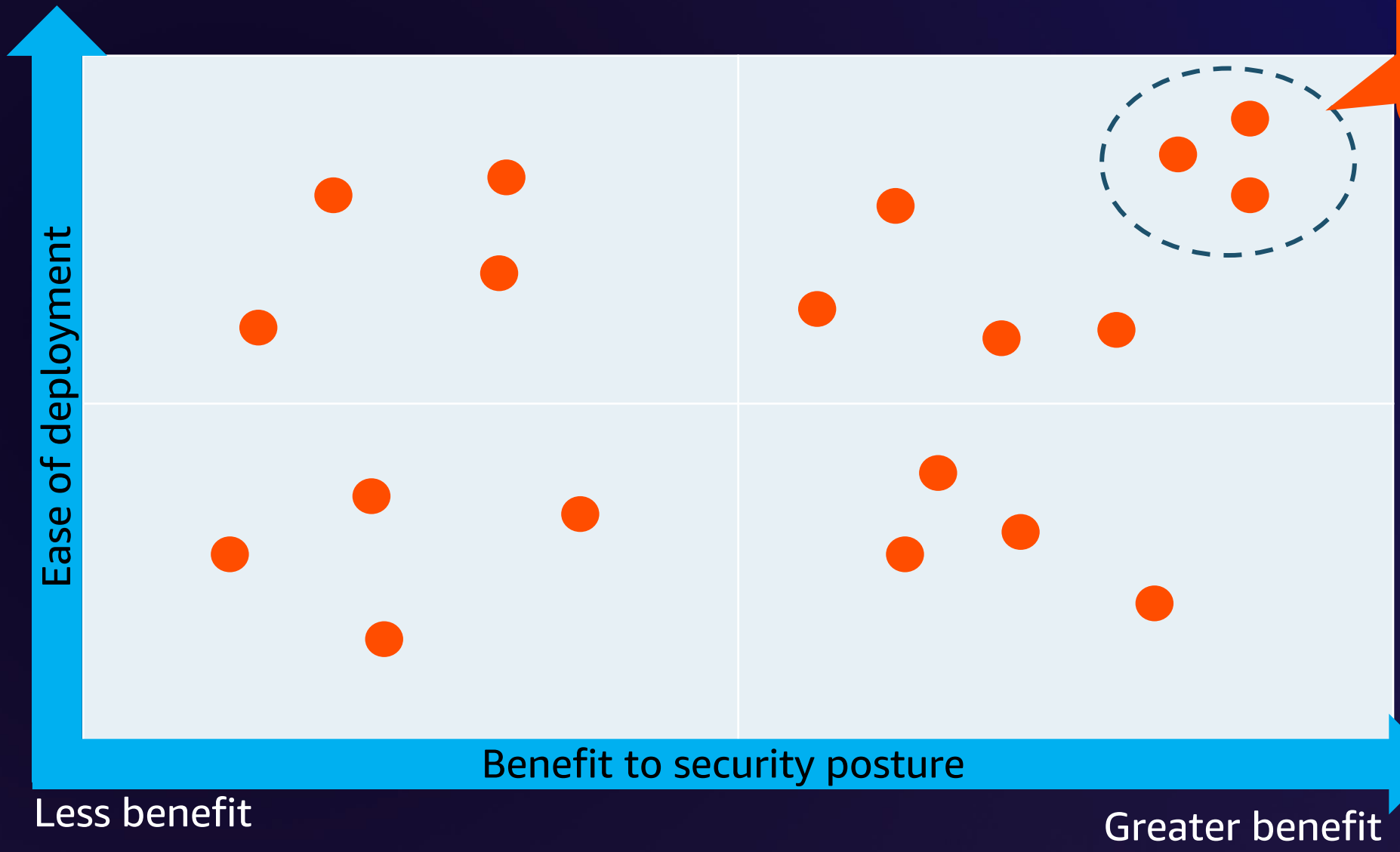
# AWS identity strategy: How to prioritize




**Easy**
1. Quick/easy to implement
2. Low effort
3. Low cost

**Benefit**
Mitigation of critical risks or risks with high likelihood of occurrence

# AWS identity strategy: How to prioritize

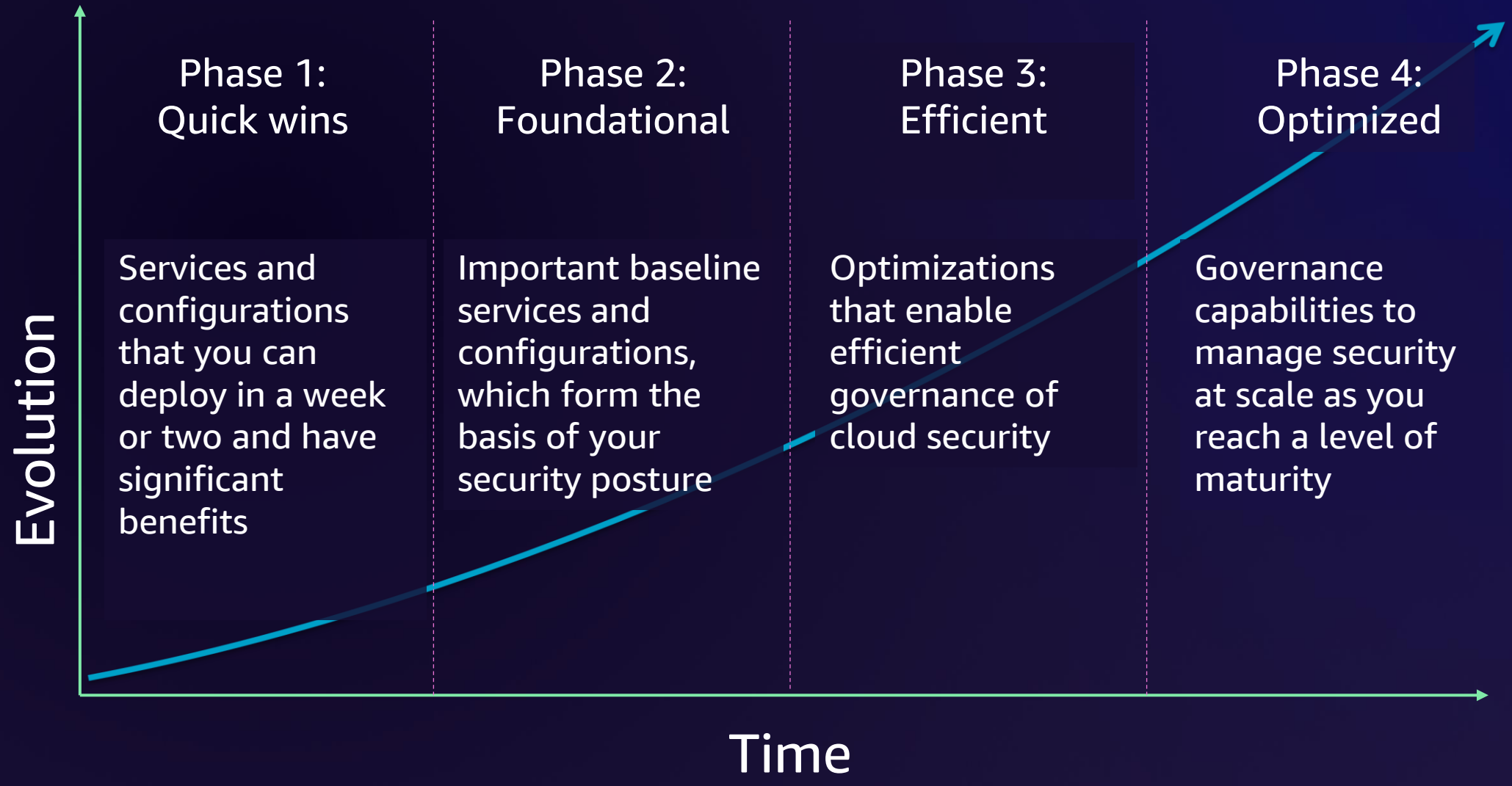



**Quick wins**
- All in 1 week
- Big benefit

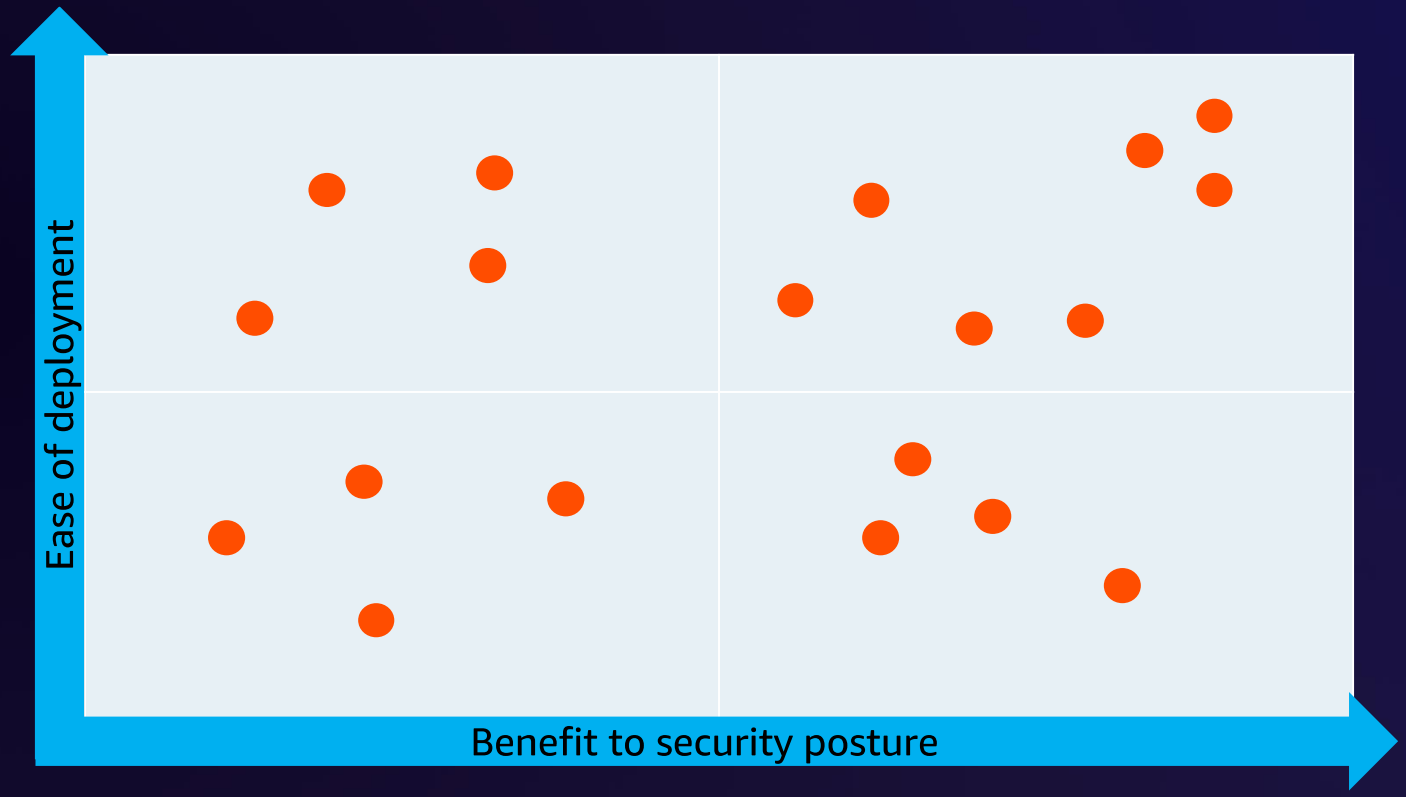
**Easy**
1. Quick/easy to implement
2. Low effort
3. Low cost

**Benefit**
Mitigation of critical risks or risks with high likelihood of occurrence

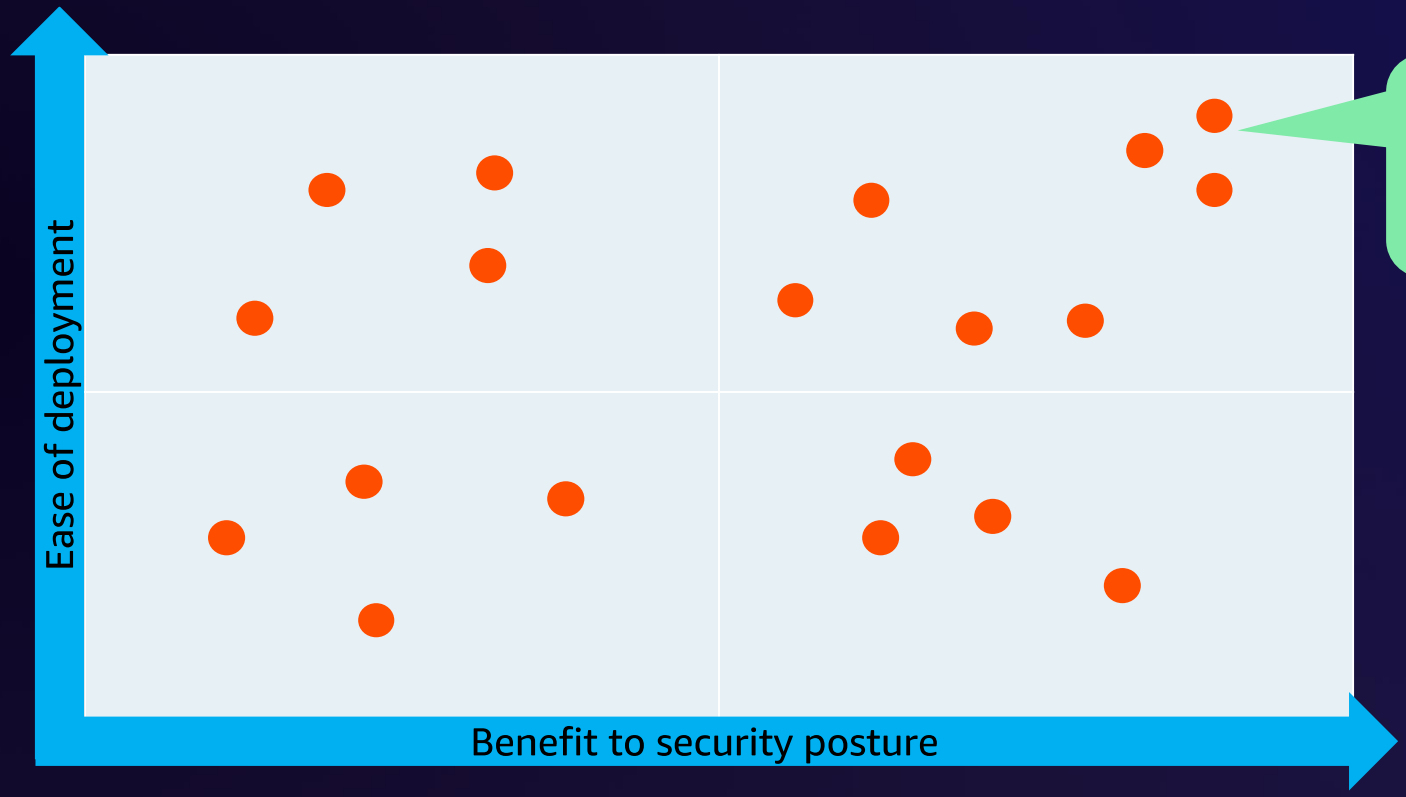
# Building your IAM journey



# AWS identity strategy: Examples

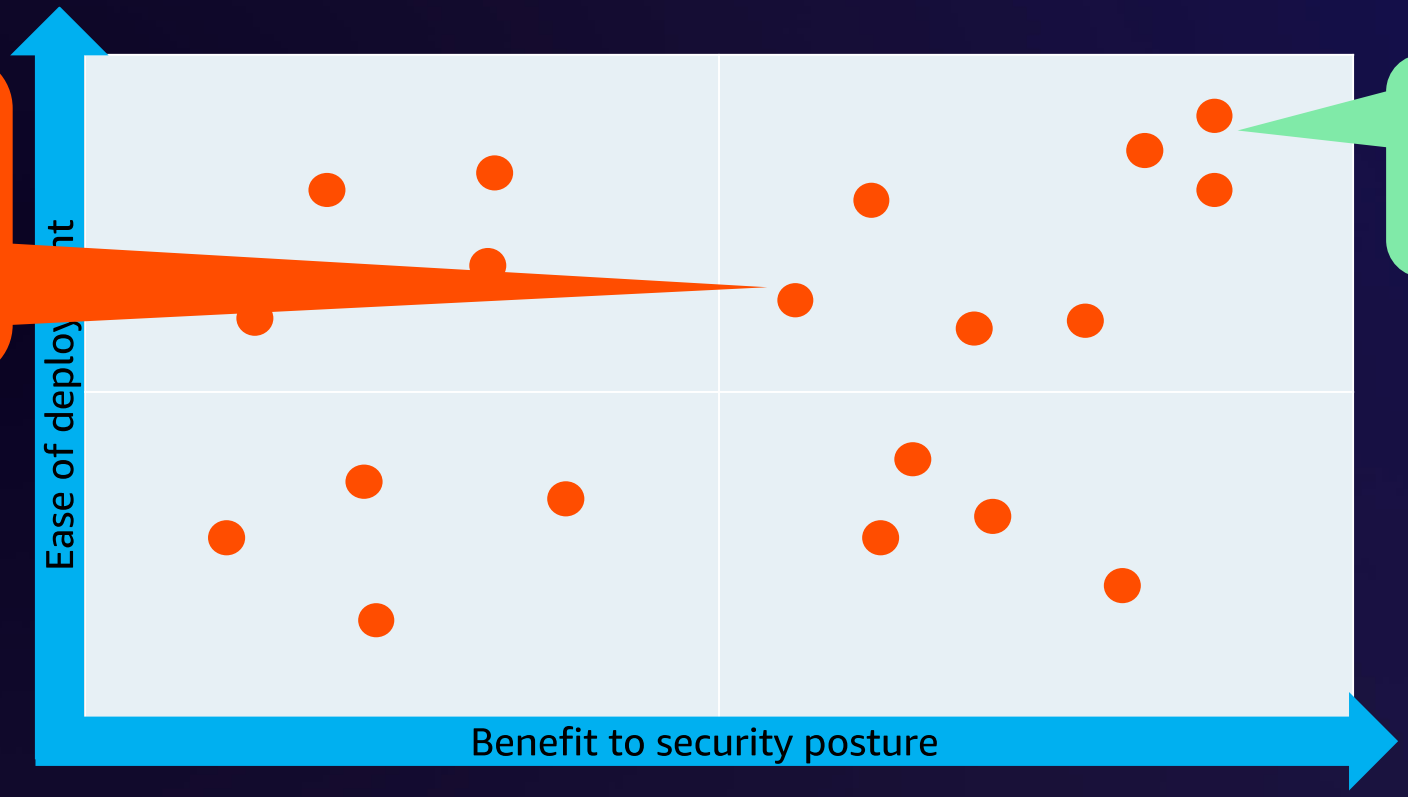


# AWS identity strategy: Examples



# AWS identity strategy: Examples

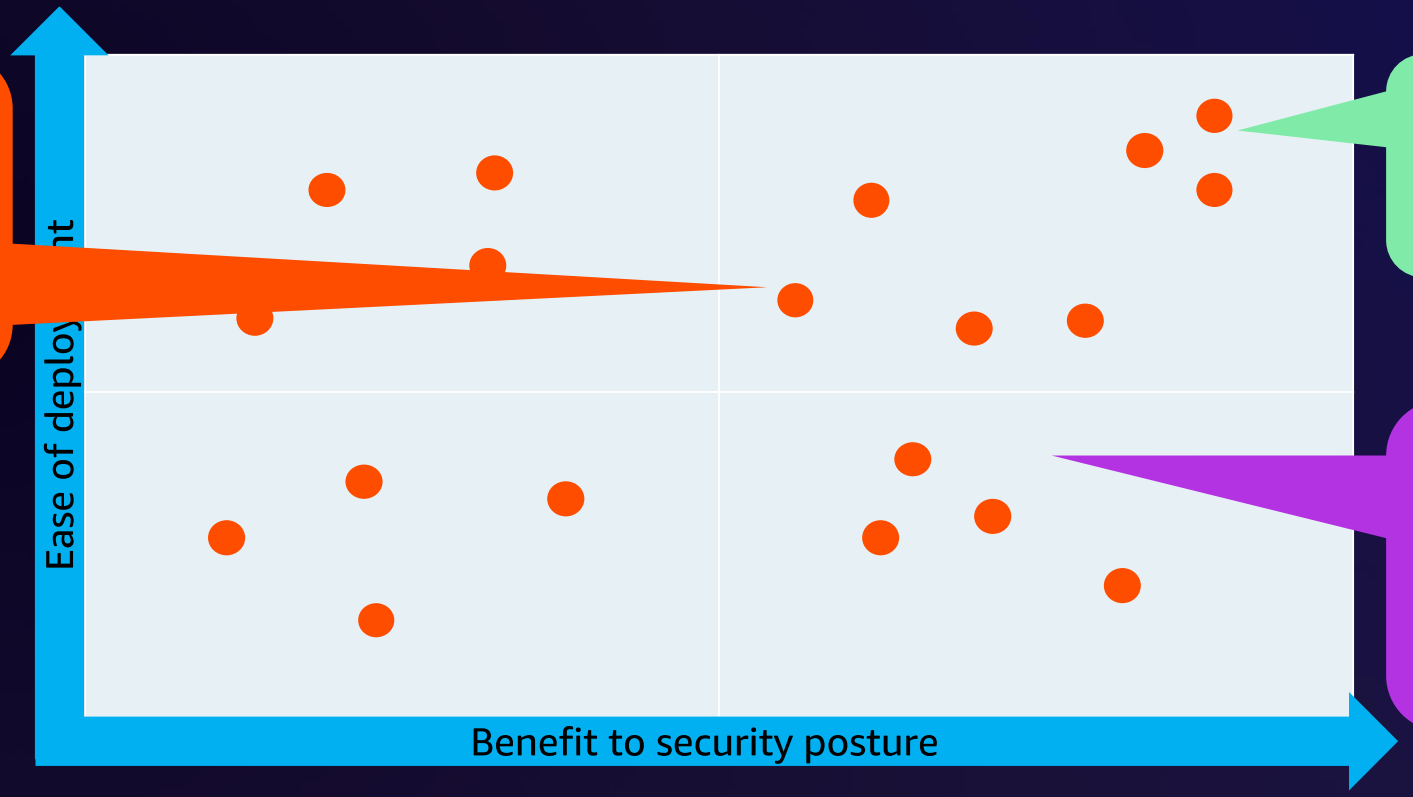
**2. Foundations**
- Organizations with AWS Control Tower
- Reduce usage of Access Keys




**1. Quick wins**
- MFA
- Identity federation

# AWS identity strategy: Examples

**2. Foundations**
- Organizations with AWS Control Tower
- Reduce usage of Access Keys



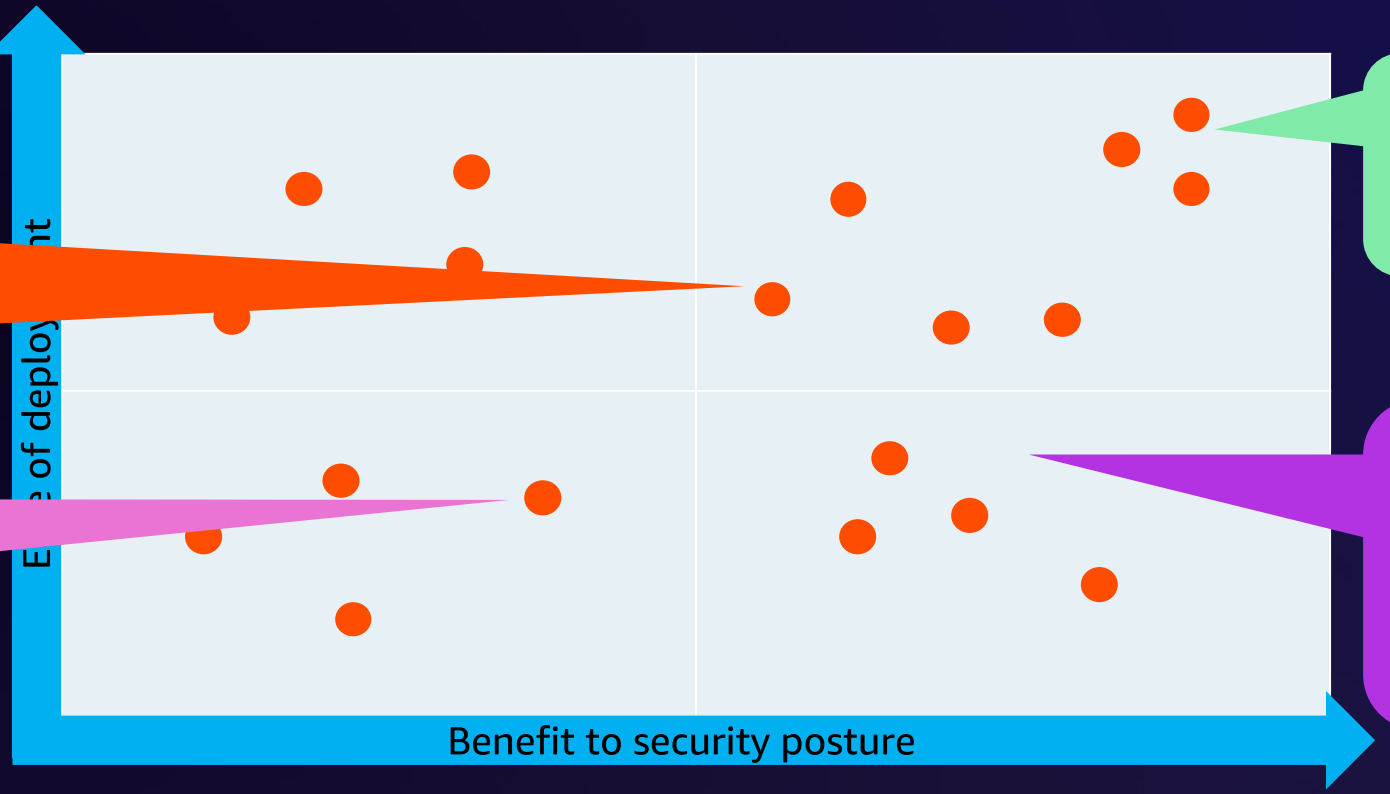

**1. Quick wins**
- MFA
- Identity federation

**3. Efficient**
- IAM Roles Anywhere
- Clean up roles with IAM Access Analyzer policy generation

# AWS identity strategy: Examples

**2. Foundations**
- Organizations with AWS Control Tower
- Reduce usage of Access Keys

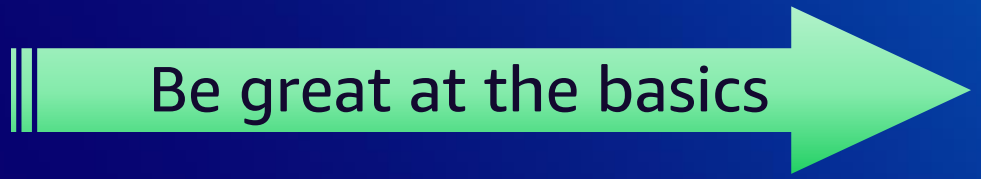
**1. Quick wins**
- MFA
- Identity federation




**4. Optimized**
- Delegate permission management

**3. Efficient**
- IAM Roles Anywhere
- Clean up roles with IAM Access Analyzer policy generation

# Phase 1: Quick Wins



# Phase 1: IAM Quick Wins

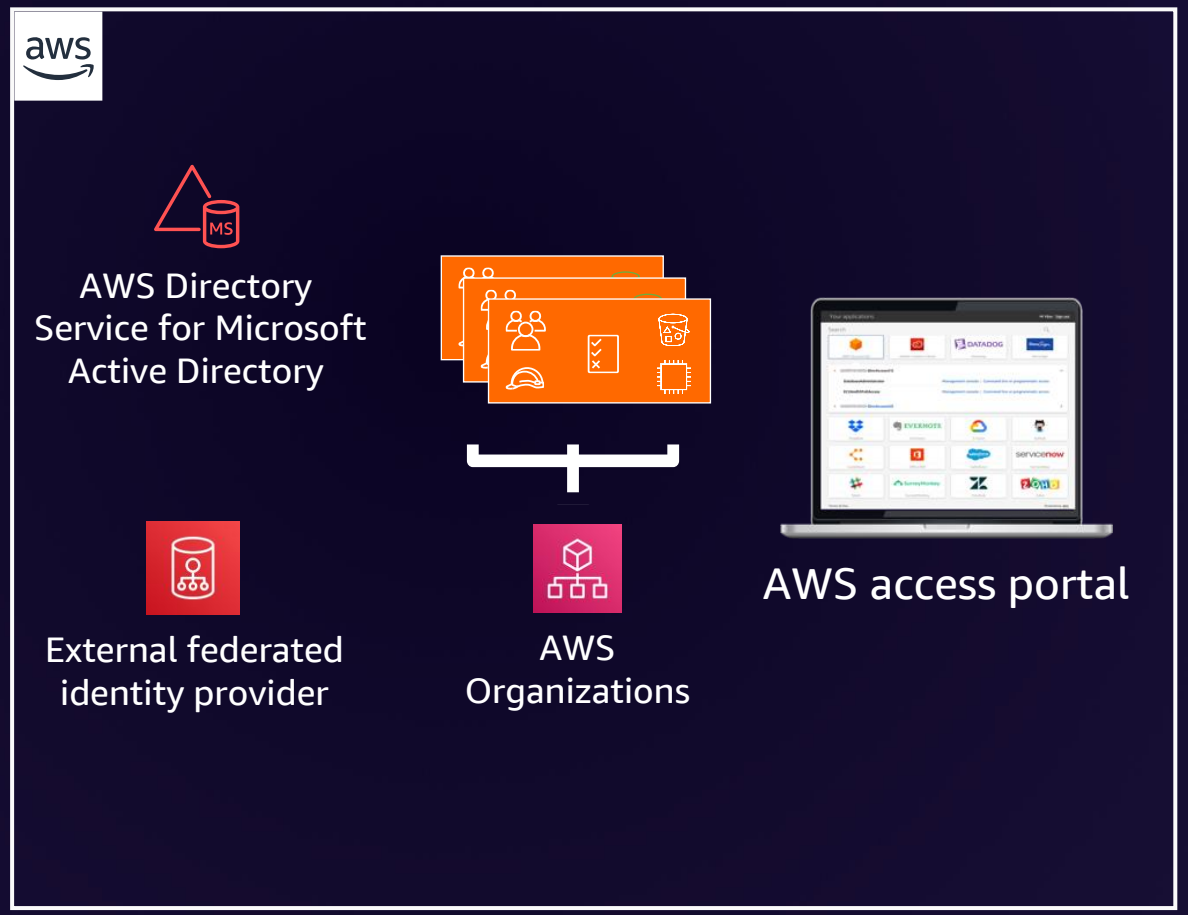
1. Implement AWS Control Tower with multiple accounts to segregate workloads
   - Federate your Identity Provider to IAM Identity Center
2. Enable MFA everywhere
   - Root, employees, contractors, customers (CIAM)
3. Avoid the use of root
   - Disable use of root on member accounts (with SCPs)

# Phase 1: IAM Quick Wins

1. Implement AWS Control Tower with multiple accounts to segregate workloads
    - Federate your identity provider to IAM Identity Center

# IAM Identity Center

NO ADDITIONAL COST




## Benefits

- Access portal to **multiple accounts**
- Central permission assignment
- Integrated with AWS **CLI v2**
- You choose your identity provider

okta · Microsoft Azure Active Directory · onelogin by ONE IDENTITY · Ping Identity · Microsoft Active Directory · jumpcloud · CYBERARK

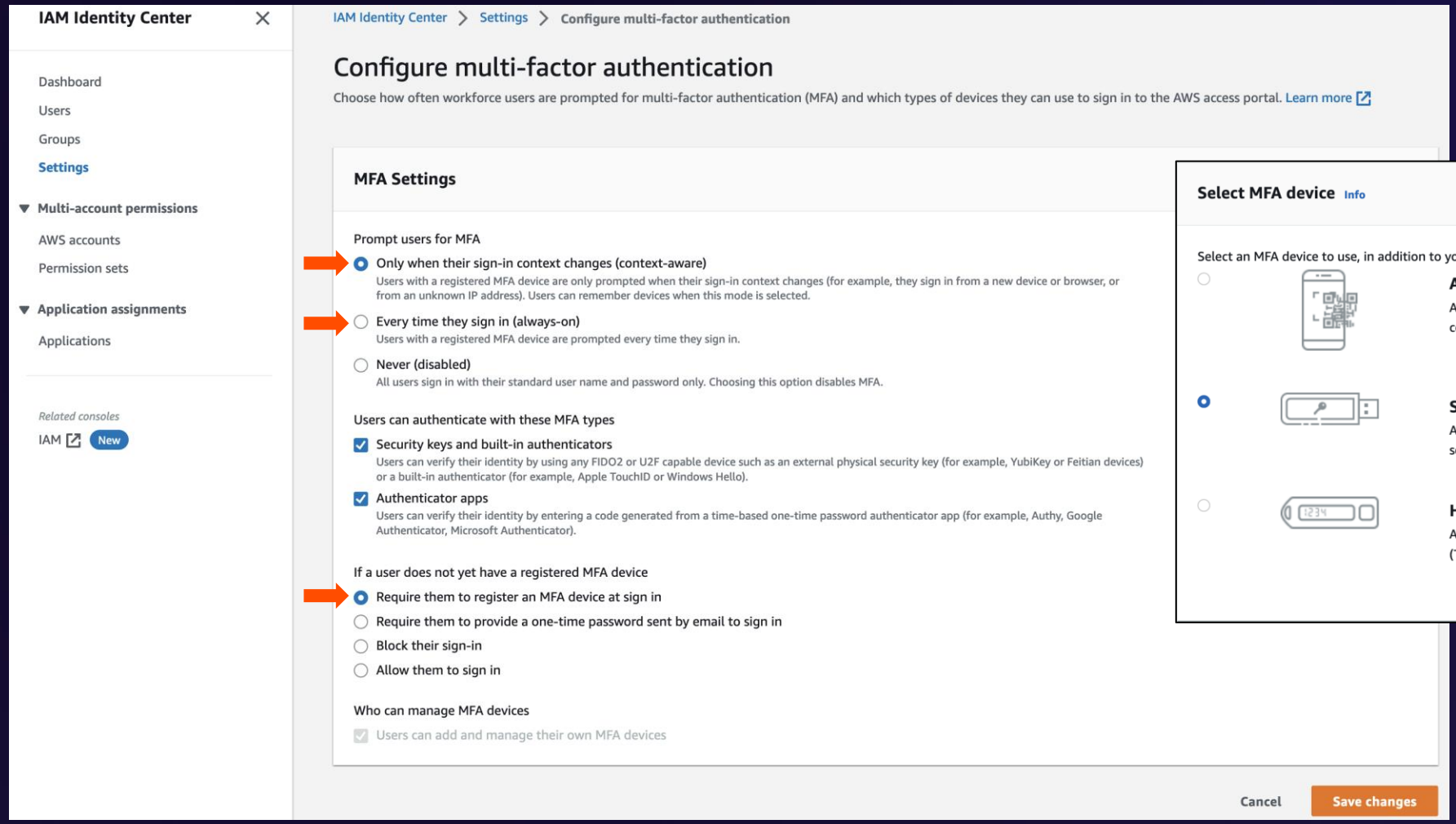
or use IAM Identity Center directory if you don't have one

# Phase 1: IAM Quick Wins

1. Implement AWS Control Tower with multiple accounts to segregate workloads
   - Federate your identity provider to IAM Identity Center
2. Activate MFA everywhere
   - Root, employees, contractors, customers (CIAM)

# MFA

## FREE MFA SECURITY KEY




https://aws.amazon.com/iam/features/mfa

# Phase 1: IAM Quick Wins

1. Implement AWS Control Tower with multiple accounts to segregate workloads
   - Federate your identity provider to IAM Identity Center
2. Activate MFA everywhere
   - Root, employees, contractors, customers (CIAM)
3. Avoid the use of root
   - Disable use of root on member accounts (with SCPs)

# Avoid the use of root

- Eliminate access keys on root
- Limit use to admin tasks that require root
- Disable the use of root with SCPs or in AWS Control Tower:

```
"Action": "*",
"Resource": "*",
"Effect": "Deny",
"Condition": {
    "StringLike": { "aws:PrincipalArn": [ "arn:aws:iam::*:root" ]}}
```

| Service | Name | Control objective | Implementation | Resource | Behavior | Release date | Guidance | Severity |
|---|---|---|---|---|---|---|---|---|
| AWS Identity and Access Management (IAM) | [AWS-GR_RESTRICT_ROOT_USER] Disallow actions as a root user | Enforce least privilege | Service control policy (SCP) | AWS::::Account | Preventive | July 30, 2019 | Strongly recommended | High |

# Phase 2: Foundations

# Phase 2: IAM Foundations

1. Enforce Security Invariants using Service Control Policies (SCPs)

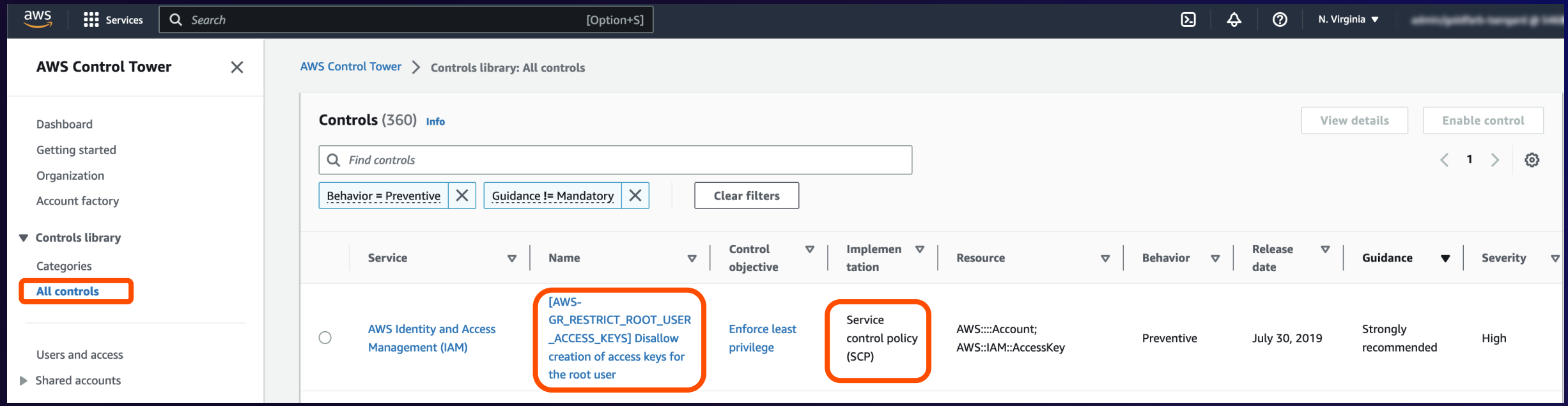
2. Limit the use of IAM Users and Access Keys to the absolute minimum
   - Use temporary credentials
   - Use IAM Roles for your compute resources (i.e., EC2 instance profile, Lambda execution roles)

3. Identify & delete unintended external permissions with IAM Access Analyzer

# Phase 2: IAM Foundations

1. Enforce security invariants using service control policies (SCPs)

# Enforce security invariants using SCPs



# Enforce security invariants using SCPs

```
"Effect": "Deny",
"Action": "ec2:RunInstances",
"Resource": ["arn:aws:ec2:*:*:instance/*"],
"Condition":
{ "StringNotEquals": {"ec2:InstanceType": "t2.micro" } }
```

Allow only this type of instances in developer sandbox account

```
"Effect": "Deny",
"Action": [ "organizations:LeaveOrganization" ],
"Resource": "*"
```

Accounts can't leave the AWS Organizations organization

https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies_scps_examples.html

# Phase 2: IAM Foundations

1. Enforce security invariants using service control policies (SCPs)

2. Limit the use of IAM users and access keys to the absolute minimum
   - Use temporary credentials
   - Use IAM roles for your compute resources (i.e., Amazon EC2 instance profile, AWS Lambda execution roles)

# Limit the use of IAM users and access keys

The amount of IAM users and access keys can become hard to manage; establish guardrails to prevent their use:

- Create SCP limiting who can create access keys

```
"Action": [ "iam:CreateUser", "iam:CreateAccessKey" ],
"Resource": [ "*" ],
"Effect": "Deny",
"Condition":
      { "StringNotEquals": { "aws:PrincipalARN": "arn:aws:iam::*:role/" } }
```

- Exceptions require tickets with justification and complete details – who the owner is, criticality, to which process it belongs, enforceable rotation
- Disable unused access keys; months later, if there are no complaints, eliminate

# Temporary credentials for AWS-native use

USE OF AWS TEMPORARY CREDENTIALS WITH AWS SERVICES

Application code on AWS

Operating system, containers

Example: Amazon EC2, Lambda

Access controlled by IAM policies (roles)

Application code automatically discovers and uses temporary credentials

AWS automatically delivers and rotates credentials

AWS resources

# Temporary credentials for AWS-native use

USE OF AWS TEMPORARY CREDENTIALS WITH AWS SERVICES

Application code on AWS

Operating system, containers

Example: Amazon EC2, Lambda

Access controlled by IAM policies (roles)

Application code automatically discovers and uses temporary credentials

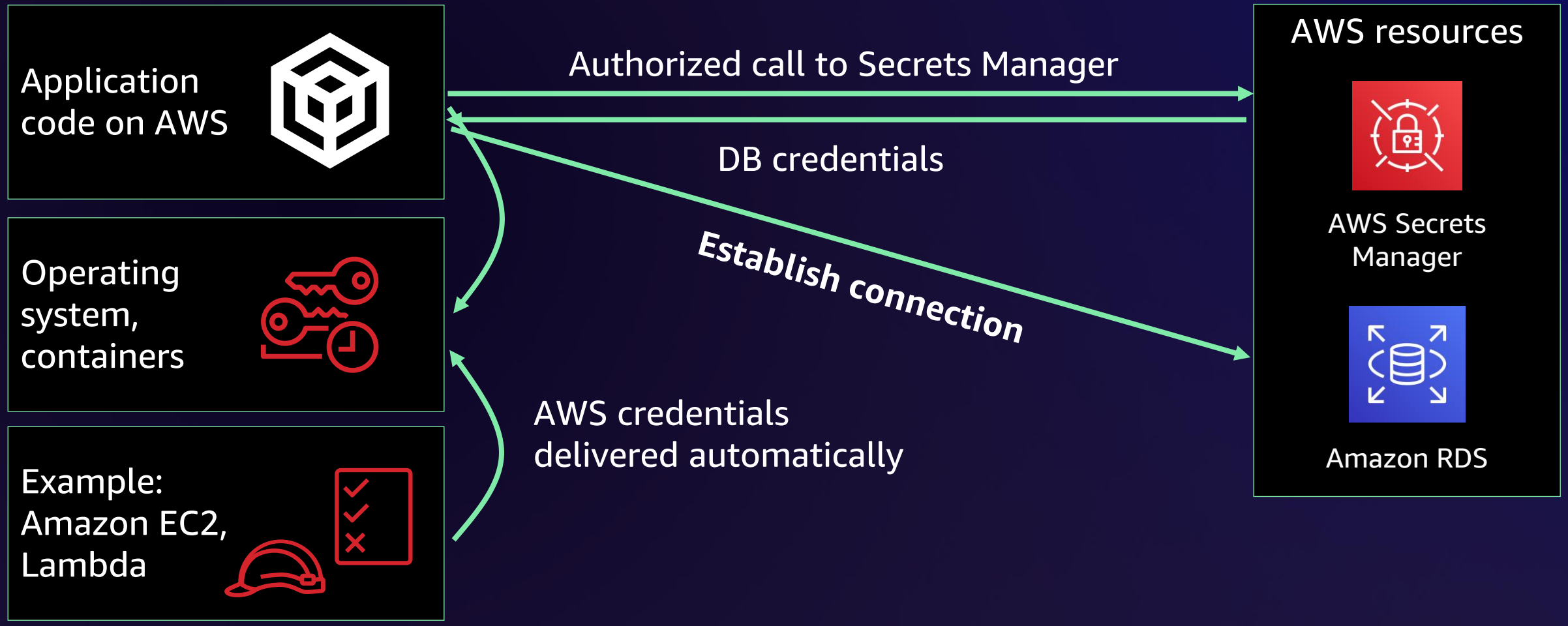
AWS automatically delivers and rotates credentials

Attribute roles to specific service actions – do not reuse roles for multiple services

AWS resources

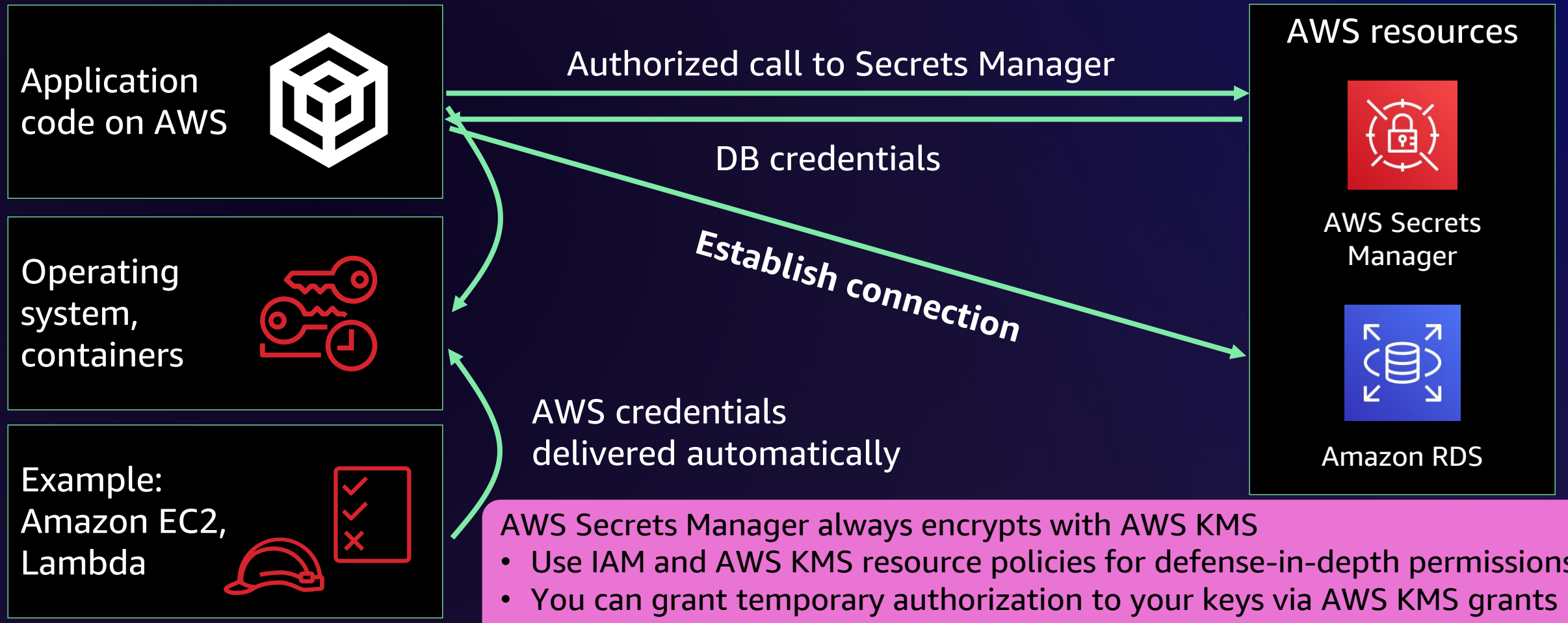
# Credentials for non-AWS services

USING SHORT-TERM AWS CREDENTIALS FOR NON AWS SERVICES WITH AWS SECRETS MANAGER



# Credentials for non-AWS services

USING SHORT-TERM AWS CREDENTIALS FOR NON AWS SERVICES WITH AWS SECRETS MANAGER




AWS Secrets Manager always encrypts with AWS KMS

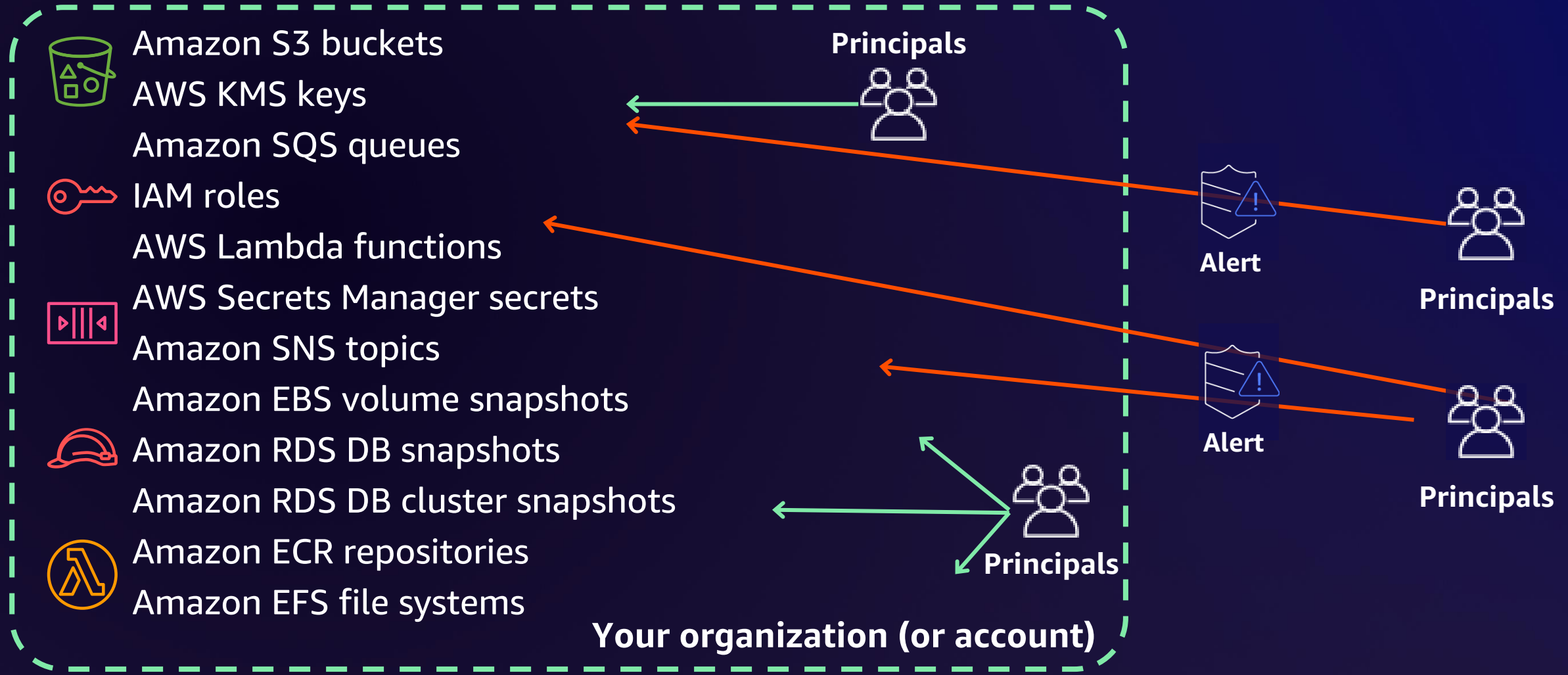
- Use IAM and AWS KMS resource policies for defense-in-depth permissions
- You can grant temporary authorization to your keys via AWS KMS grants without modifying IAM policies

# Phase 2: IAM Foundations

1. Enforce security invariants using service control policies (SCPs)
2. Limit the use of IAM users and access keys to the absolute minimum
   - Use temporary credentials
   - Use IAM roles for your compute resources (i.e., Amazon EC2 instance profile, AWS Lambda execution roles)
3. Identify and delete unintended external permissions with IAM Access Analyzer

# Identify & delete unintended external access

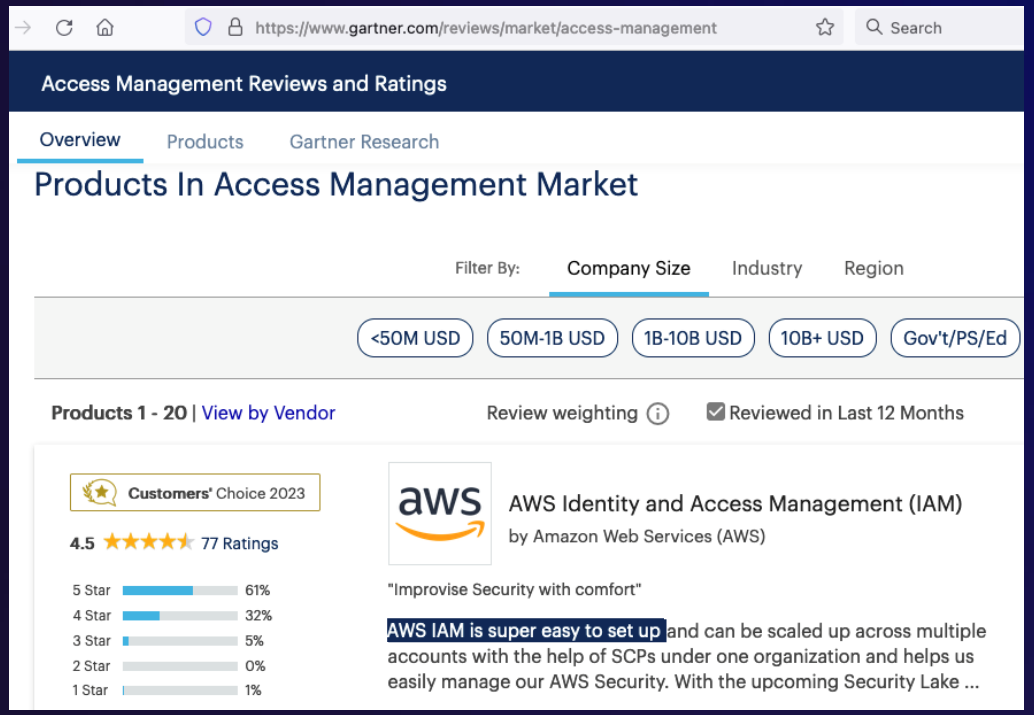
IAM ACCESS ANALYZER (NO ADDITIONAL COST)



# Phase 3: Efficient

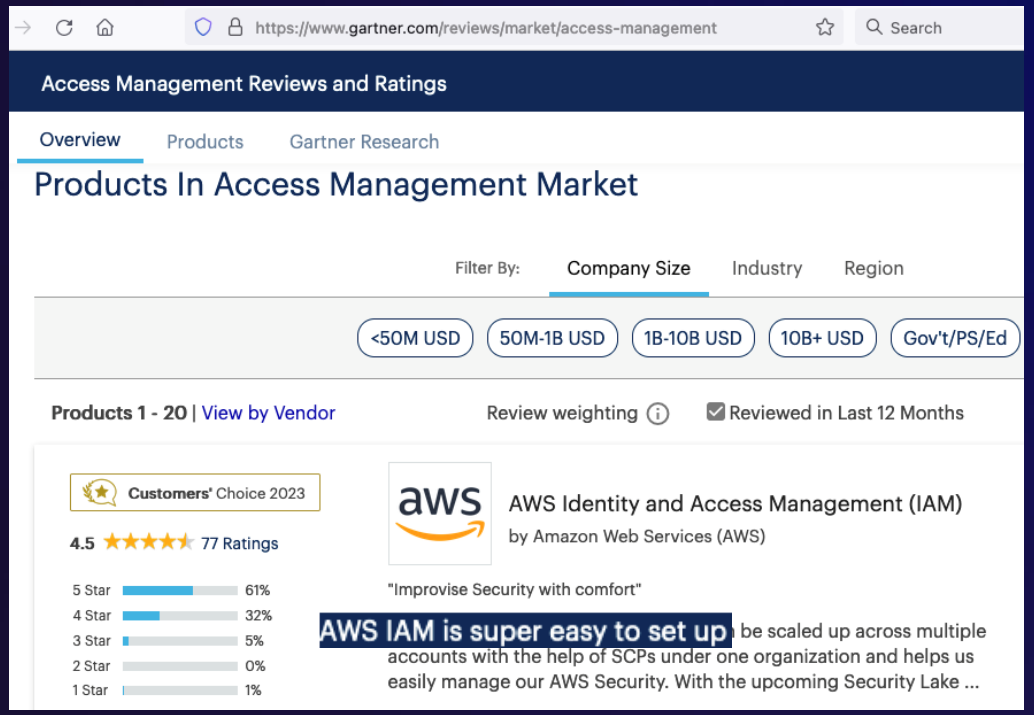
# Phase 3: What are we talking about?

# Phase 3: What are we talking about?




Source: Gartner, 2023. Products in Access Management Market

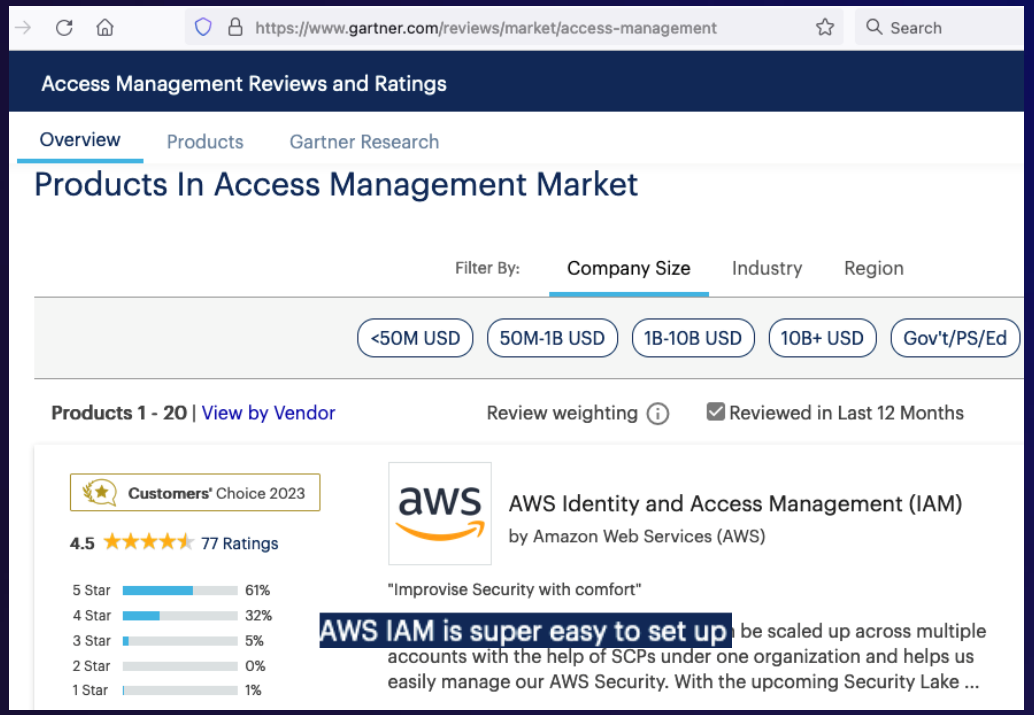
# Phase 3: What are we talking about?




Source: Gartner, 2023. Products in Access Management Market

# Phase 3: What are we talking about?

Ease of use is directly proportionate to your knowledge and comfort with the tools you can use.

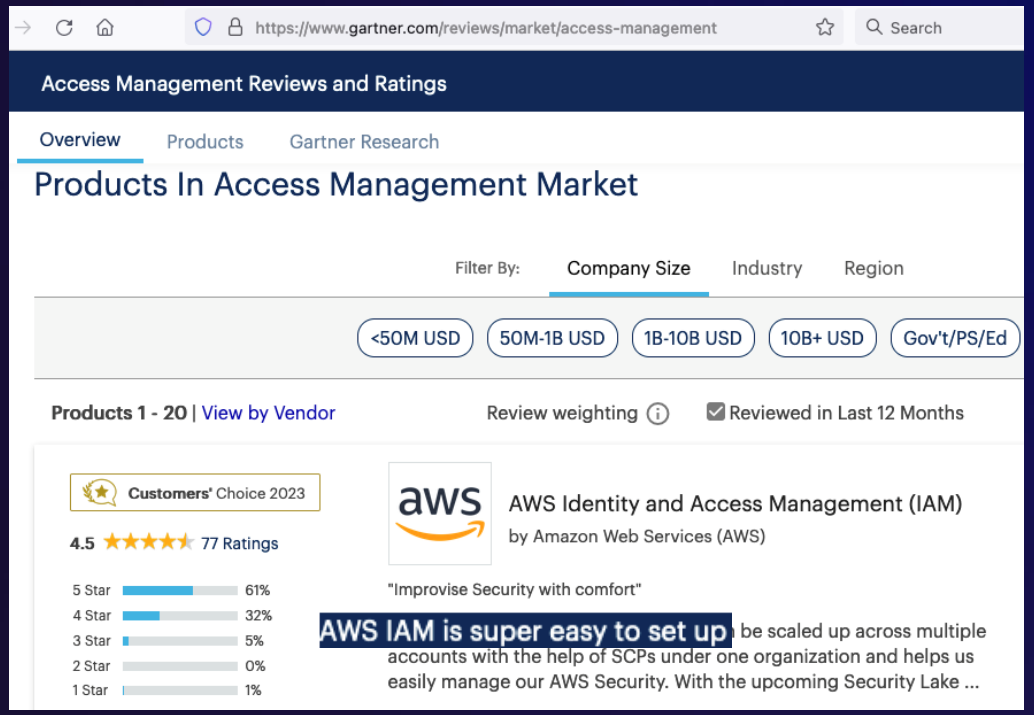



Source: Gartner, 2023. Products in Access Management Market

# Phase 3: What are we talking about?

Ease of use is directly proportionate to your knowledge and comfort with the tools you can use.

AWS security tools will help you with the undifferentiated heavy lift and work an "easy button" in your security efficiency journey.

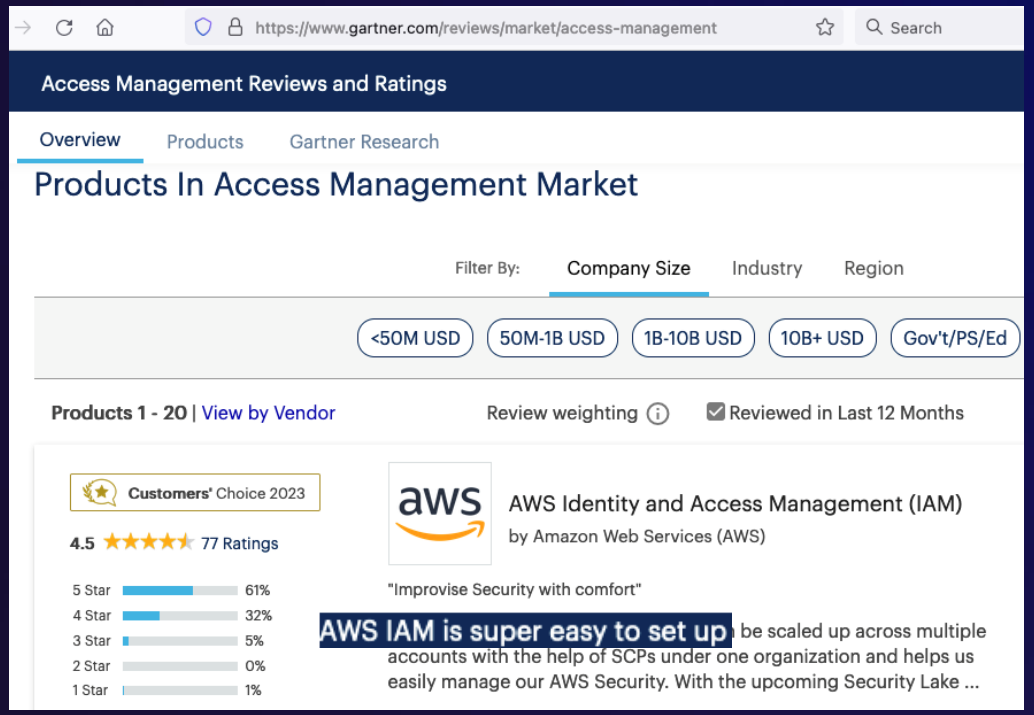



Source: Gartner, 2023. Products in Access Management Market

# Phase 3: What are we talking about?

Ease of use is directly proportionate to your knowledge and comfort with the tools you can use.

AWS security tools will help you with the undifferentiated heavy lift and work an "easy button" in your security efficiency journey.




Source: Gartner, 2023. Products in Access Management Market

**Efficiency**:

- Resources expended in relation to the accuracy and completeness with which users achieve their goals (NIST 800-55 R1)
- The ratio of useful work performed by a machine [...] (Oxford Dictionary)

# Phase 3: IAM Efficient

1. Clean up what is not being used
   - Use IAM Access Advisor to remove unused access keys/permissions/roles
2. Clean up excessive permissions
   - Perform privilege reviews (work toward least privilege)
   - Leverage IAM Access Analyzer Policy Generation
3. Temporary credentials for external access
   - IAM Roles Anywhere
4. IAM as Code
   - Leverage versioning and infrastructure as code for IAM Policies

# Phase 3: IAM Efficient

1. Clean up what is not being used
   - ➢ Use IAM access advisor to remove unused access keys/permissions/roles

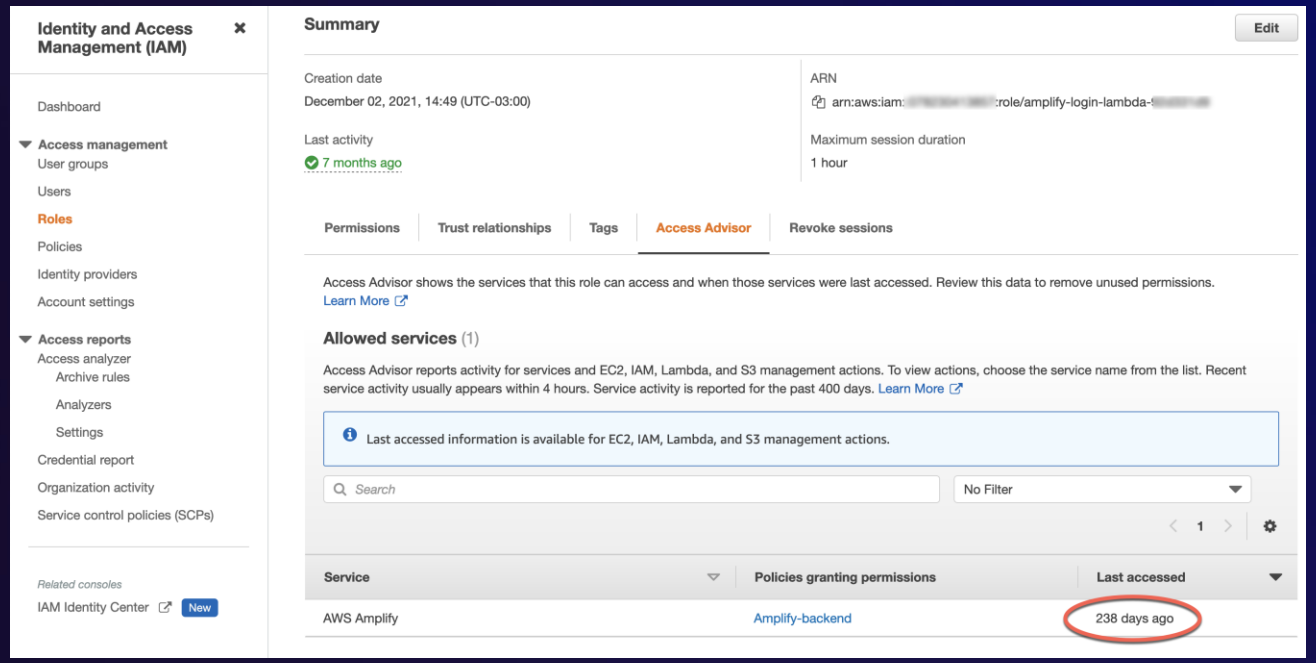
# IAM access advisor: Delete unused permissions

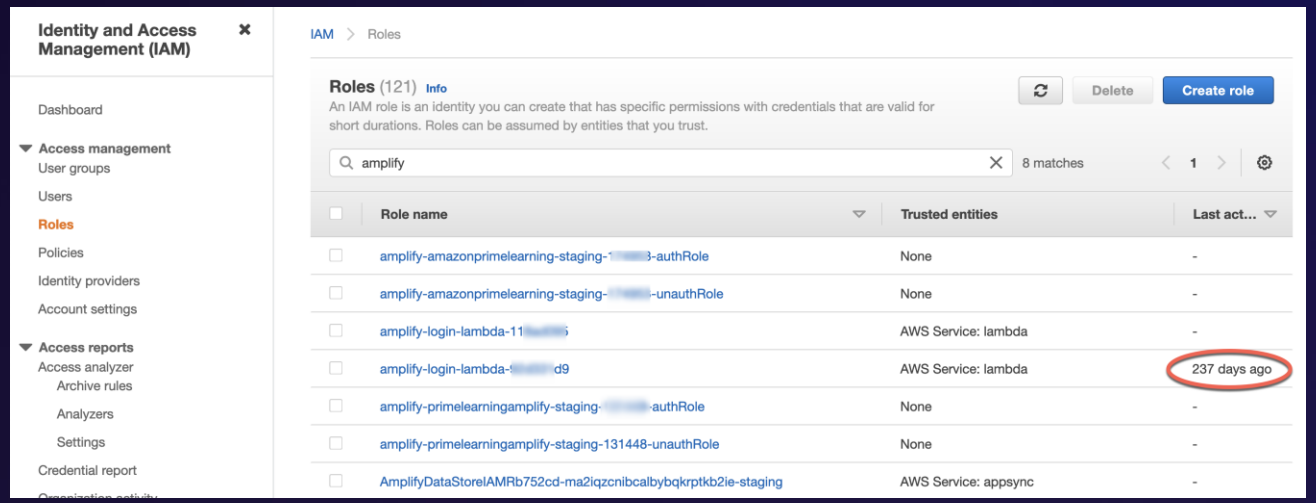
**Step 1:** Identify unused access keys, roles and permissions

- Review the last activity date

**Step 2:** Disable/delete unused access keys, roles, and permissions

| Access key ID | Created | Last used |
|---|---|---|
| AKIA[redacted]I | 2021-04-26 18:10 UTC-0300 | 2021-06-14 [redacted]:05 |



# Phase 3: IAM efficient

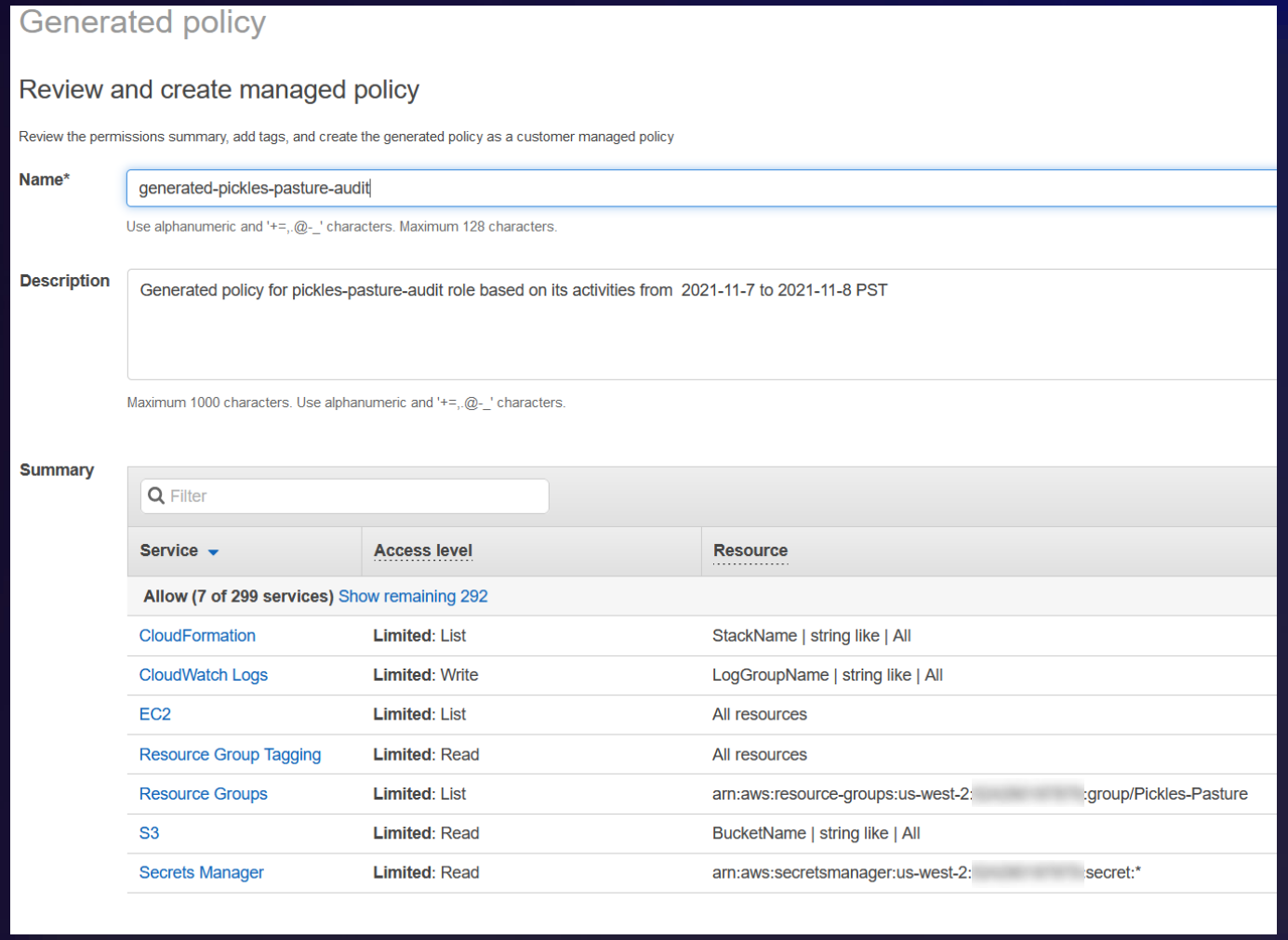
1. Clean up what is not being used
   - Use IAM access advisor to remove unused access keys/permissions/roles
2. Clean up excessive permissions
   - Perform privilege reviews (work toward least privilege)
   - Use IAM Access Analyzer policy generation

# Automatic activity-based policy generation

IAM ACCESS ANALYZER (NO ADDITIONAL COST)

Scenario: Development team doesn't create strong permissions

1. Configuration (e.g., 90 days)
2. Generate JSON policies from events in AWS CloudTrail
3. Review and customize the policy with minimum privilege
4. Create and associate the policy

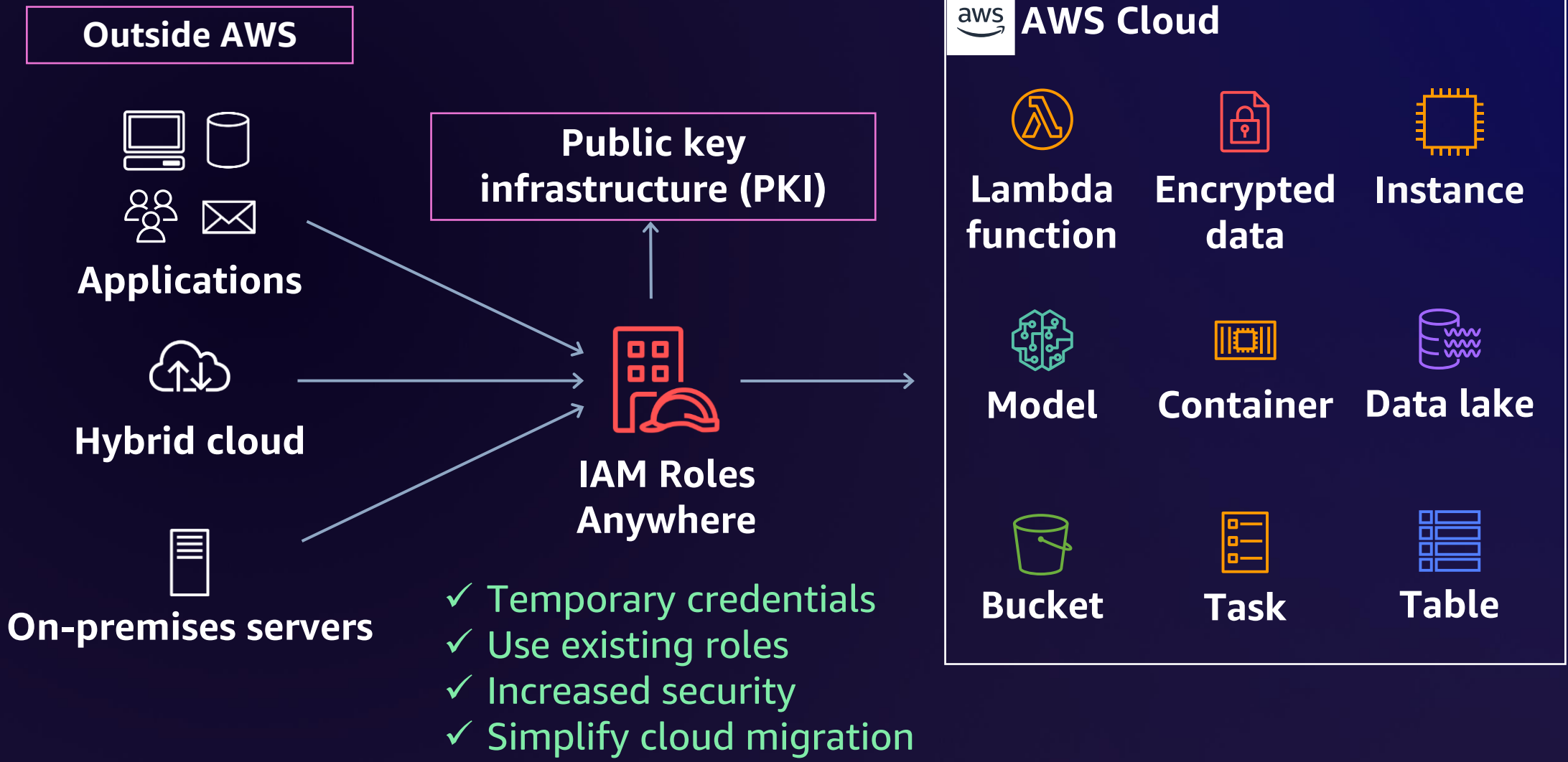


# Phase 3: IAM efficient

1. Clean up what is not being used
   - Use IAM access advisor to remove unused access keys/permissions/roles
2. Clean up excessive permissions
   - Perform privilege reviews (work toward least privilege)
   - Use IAM Access Analyzer policy generation
3. Temporary credentials for external access
   - IAM Roles Anywhere
   - AWS IoT Core

# IAM Roles Anywhere

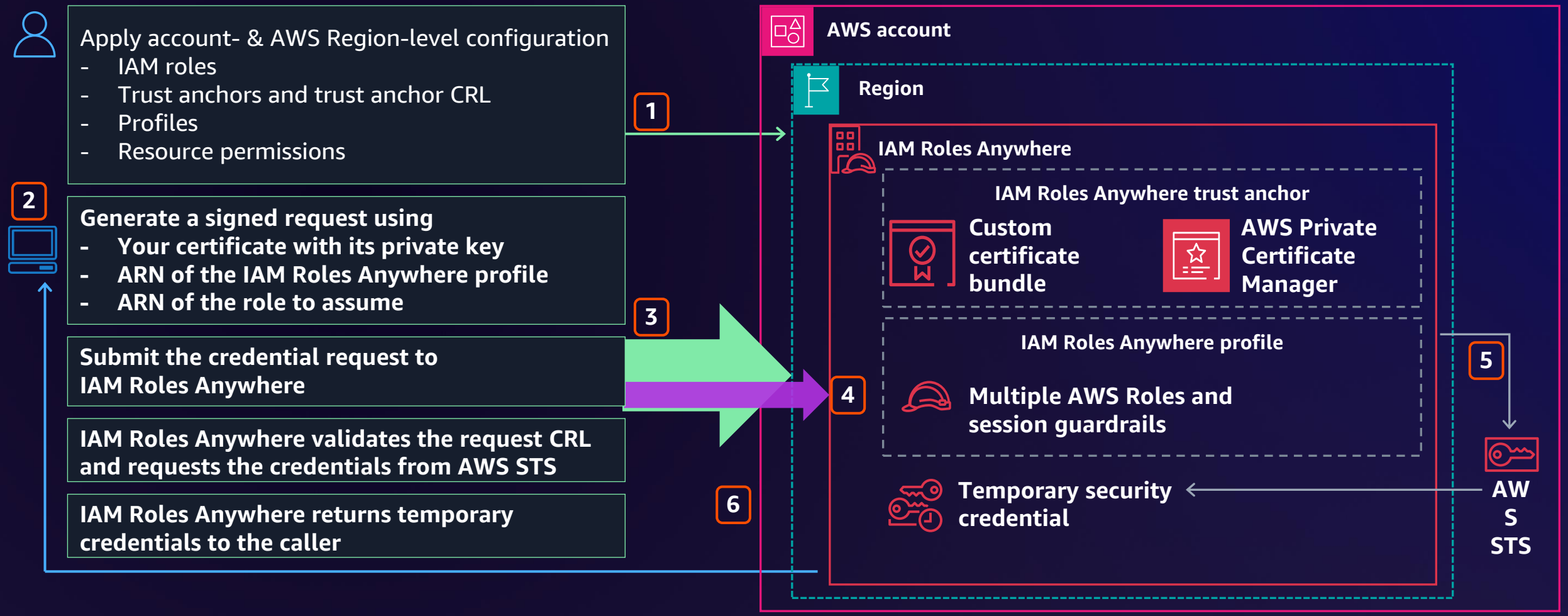
EXTEND THE USE OF IAM ROLES TO WORKLOADS OUTSIDE OF AWS




- ✓ Temporary credentials
- ✓ Use existing roles
- ✓ Increased security
- ✓ Simplify cloud migration

# IAM Roles Anywhere

SIMPLIFIED FLOW DIAGRAM



# IAM policies with IAM Roles Anywhere

USING UNIQUE FEATURES OF IAM ROLES ANYWHERE

IAM Roles Anywhere maps your certificate's X.509 subject and subject alternative name into principal tags

You can use principal tags within your trust policies, grant, or guardrails policies

# IAM policies with IAM Roles Anywhere

USING UNIQUE FEATURES OF IAM ROLES ANYWHERE

IAM Roles Anywhere maps your certificate's X.509 subject and subject alternative name into principal tags

```
"Condition": {
        "StringEquals": {
       "aws:PrincipalTag/x509SAN/DNS": "example.com",
       "aws:PrincipalTag/x509SAN/Name/CN": "Alice"  }
        }
```

You can use principal tags within your trust policies, grant, or guardrails policies

# IAM policies with IAM Roles Anywhere

USING UNIQUE FEATURES OF IAM ROLES ANYWHERE

IAM Roles Anywhere maps your certificate's X.509 subject and subject alternative name into principal tags

```
"Condition": {
        "StringEquals": {
       "aws:PrincipalTag/x509SAN/DNS": "example.com",
       "aws:PrincipalTag/x509SAN/Name/CN": "Alice”  }
        }
```

You can use principal tags within your trust policies, grant, or guardrails policies

```
"Condition": {
     "StringEquals": {
        "aws:PrincipalTag/x509Subject/CN": "Alice" }
}
```

# Practical use notes

FOR USE OF AWS TEMPORARY CREDENTIAL

- Use PKI strengths
  - Attribute role assumptions to specific workloads – use certificate per workload
  - Expire your certificates per your business and security needs
  - Automate your CA's CRL publication to your IAM Roles Anywhere CRL store

- Limit the scope for role assumption
  - Use conditions in your IAM role trust policy to specify what entity can assume the role
  - Use conditions in your IAM Roles Anywhere session policy to set guardrails

# Phase 3: IAM efficient

1. Clean up what is not being used
   - Use IAM Access Advisor to remove unused access keys/permissions/roles
2. Clean up excessive permissions
   - Perform privilege reviews (work toward least privilege)
   - Use IAM Access Analyzer policy generation
3. Temporary credentials for external access
   - IAM Roles Anywhere
4. IAM as code
   - Use versioning and infrastructure as code for IAM policies

# Phase 4: Optimized

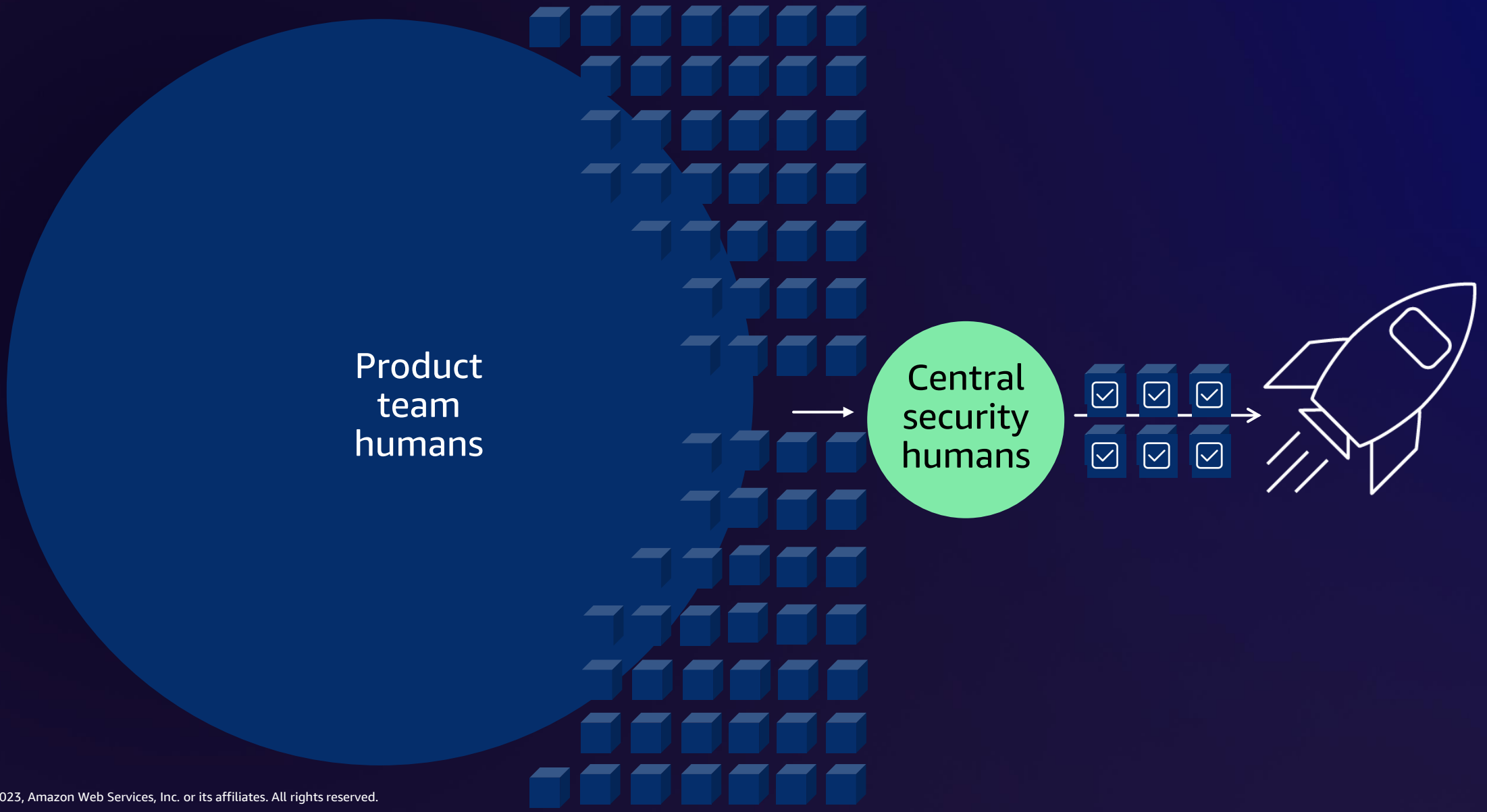
# Phase 4: IAM Optimized

1. Delegate permission management
   - Let developers safely create roles and get out of their way
   - IAM Pipeline: Establishing a process to create roles
2. Analyze CloudTrail for proper access
   - Denied actions are great signals of a misconfiguration or malicious activity
3. Establish a data perimeter
4. Restrict access using Zero Trust principles
   - Limited time authorization
     Check out Temporary Elevated Access Management (TEAM) solution
     https://aws-samples.github.io/iam-identity-center-team

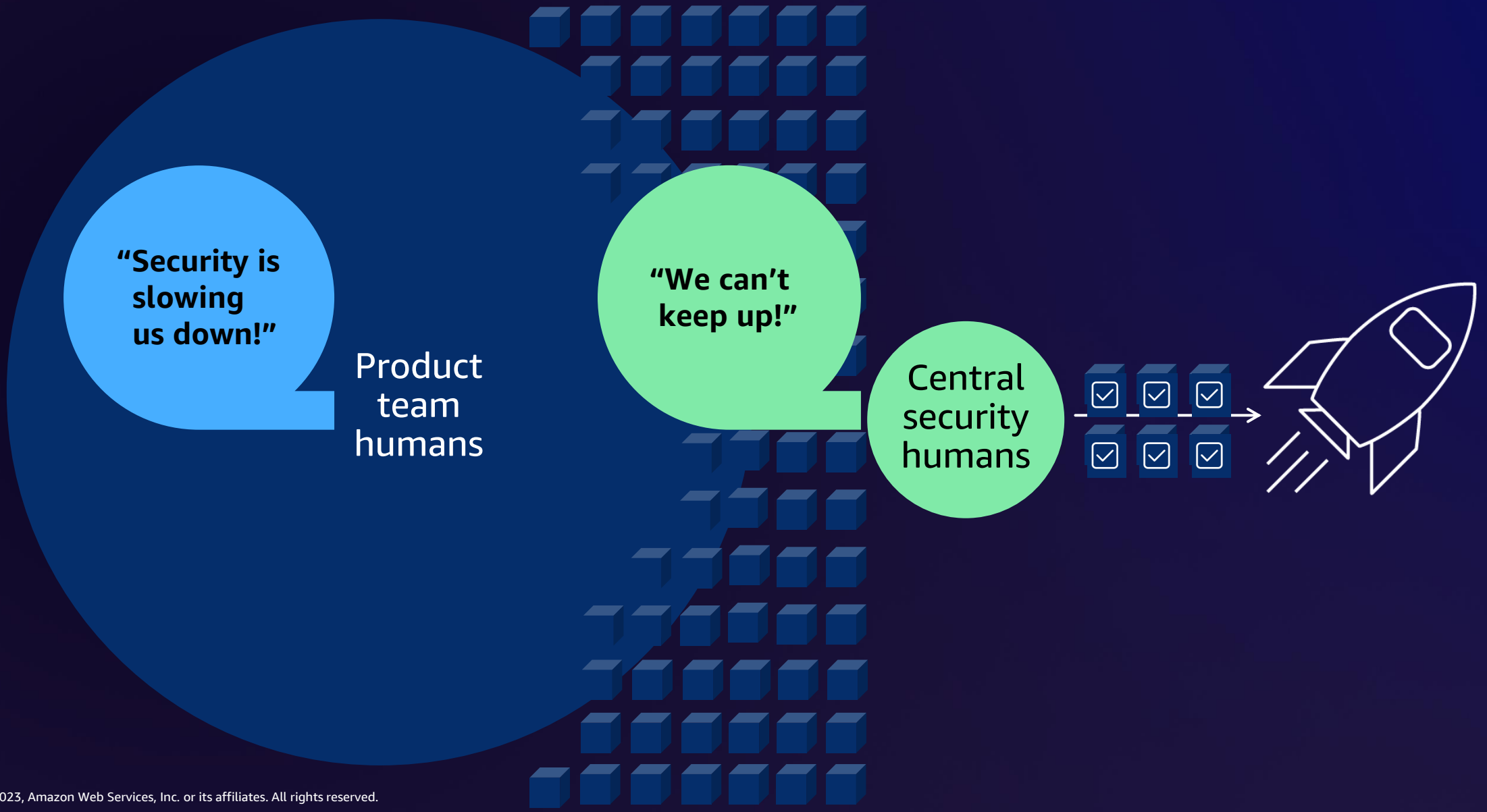
# Phase 4: IAM Optimized

1. Delegate permission management
   - Let developers safely create roles, and then get out of their way
   - IAM pipeline – establishing a process to create roles

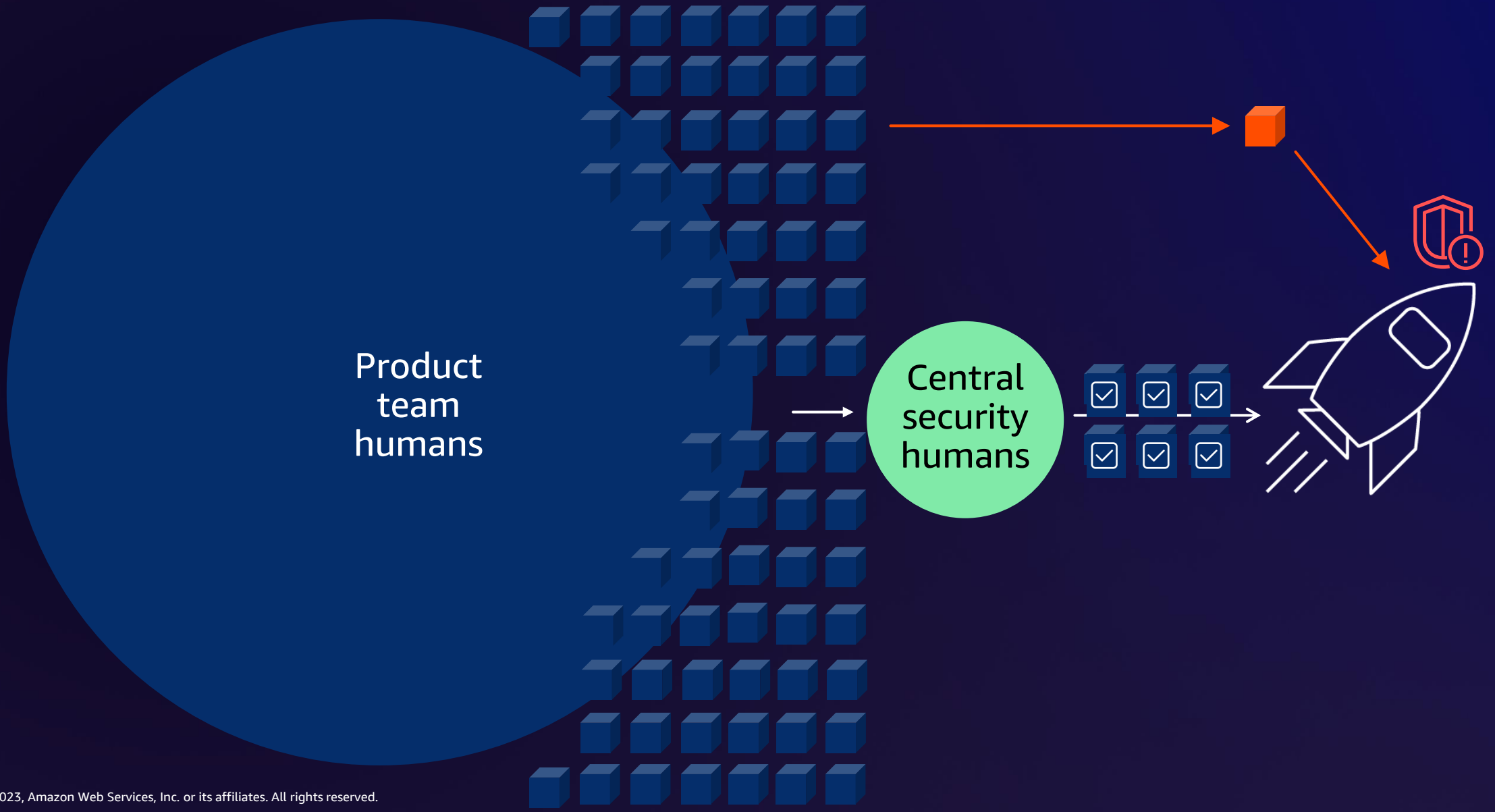
# A challenge with scaling security: Humans



# A challenge with scaling security: Humans



# A challenge with scaling security: Humans



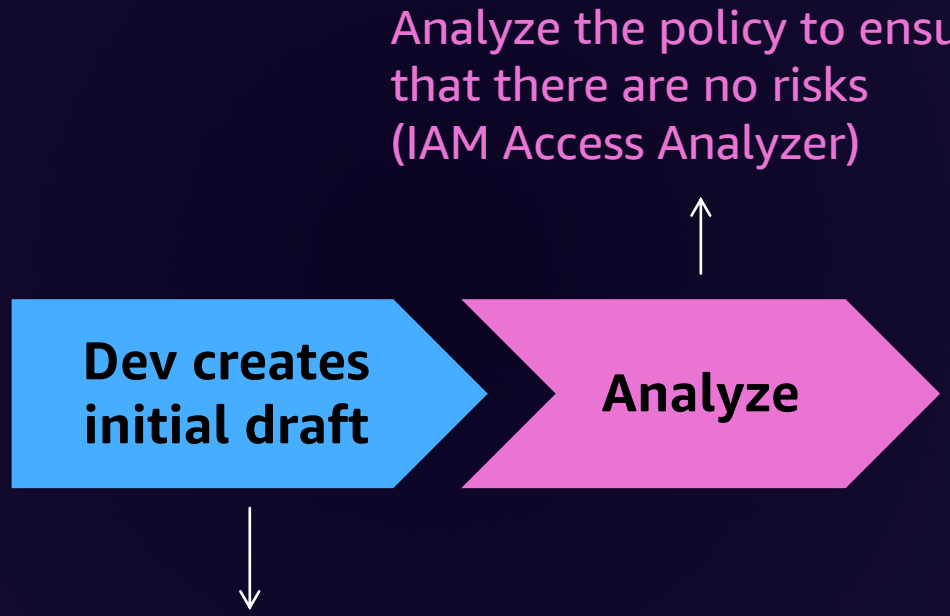
# Let developers safely create roles, and then get out of their way

# Let developers safely create roles, and then get out of their way

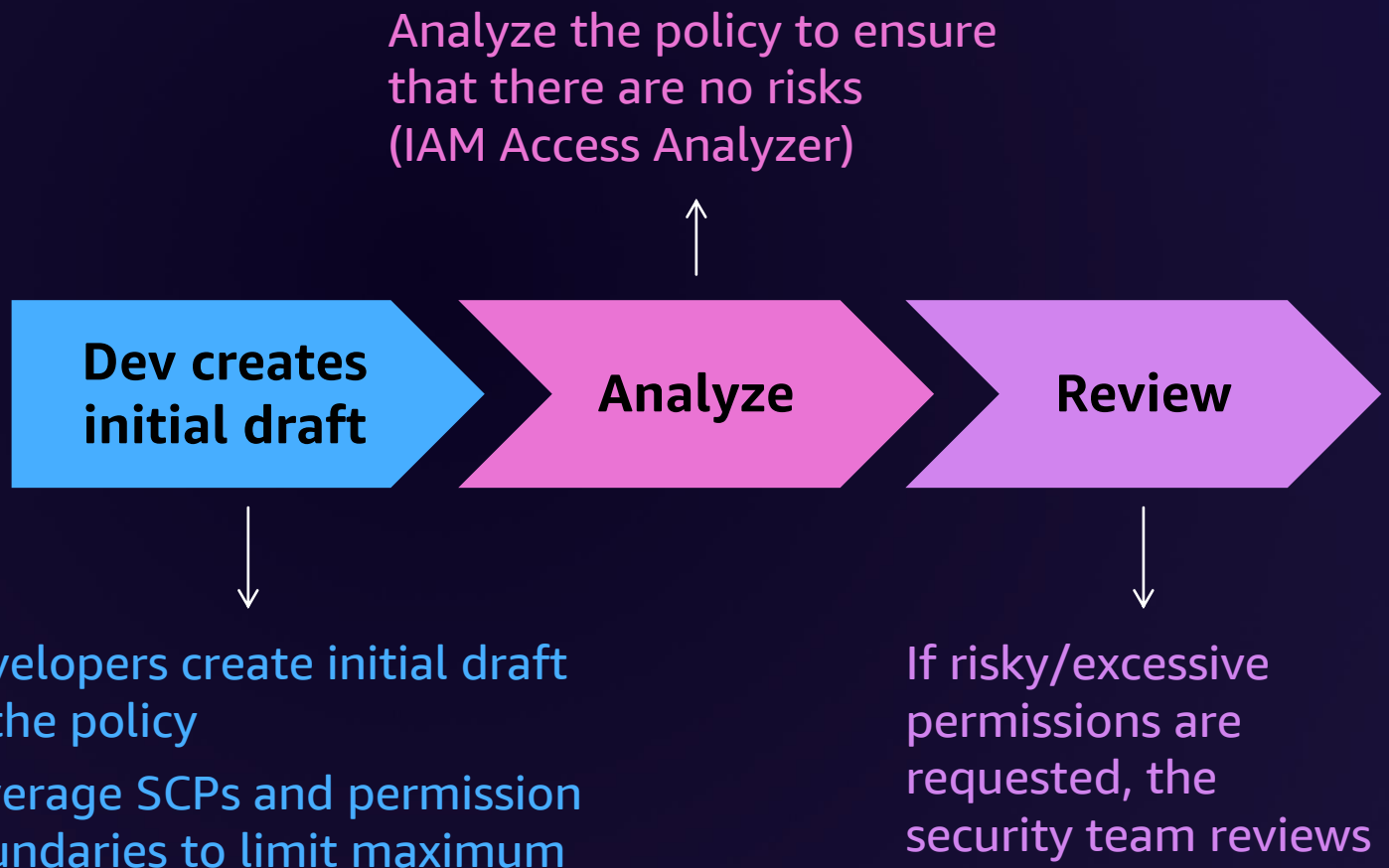

Developers create initial draft of the policy

Leverage SCPs and permission boundaries to limit maximum permissions

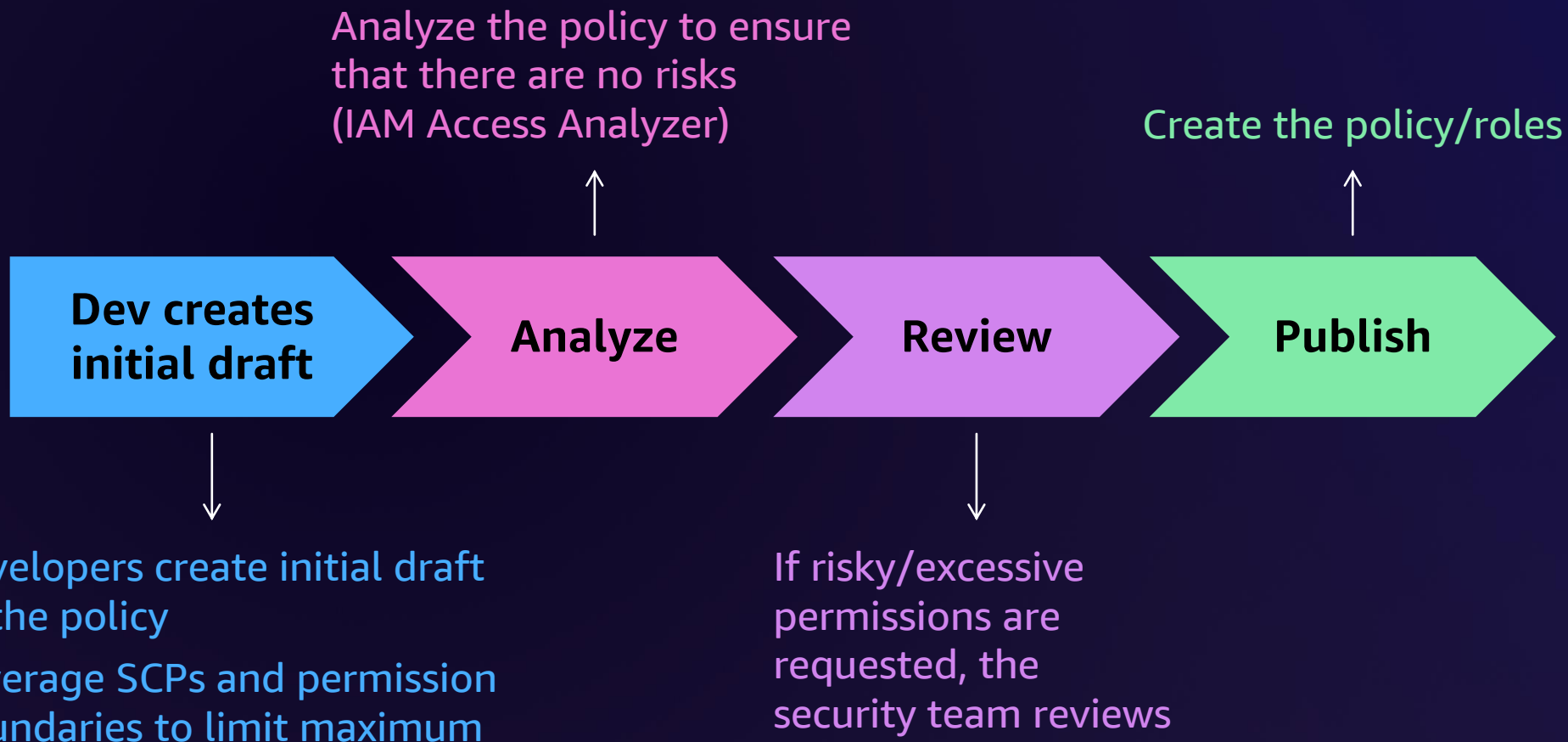
# Let developers safely create roles, and then get out of their way



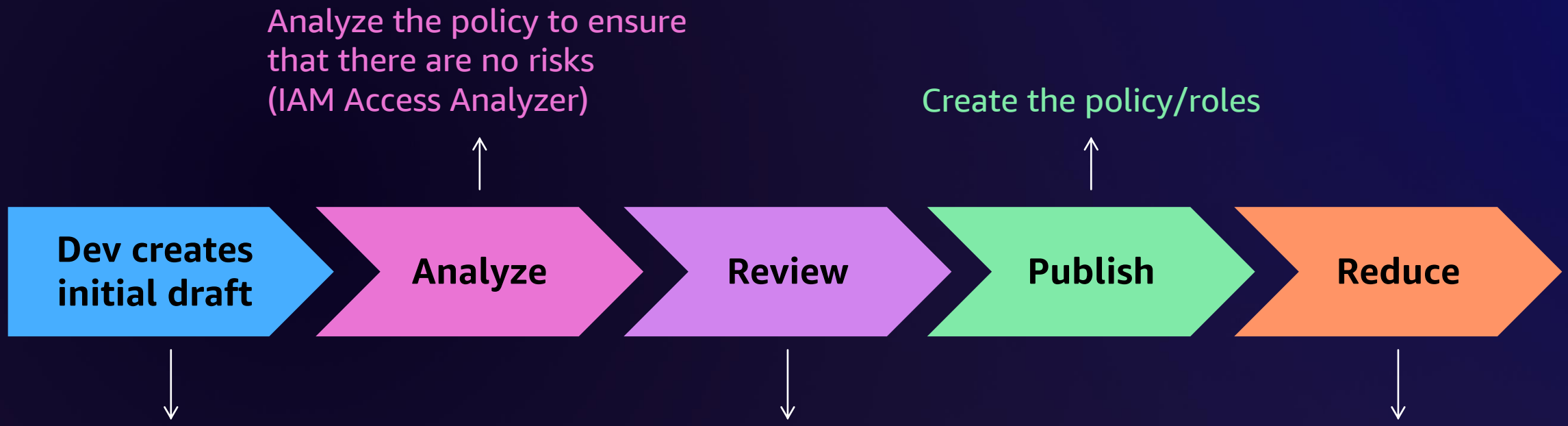
# Let developers safely create roles, and then get out of their way



# Let developers safely create roles, and then get out of their way



# Let developers safely create roles, and then get out of their way

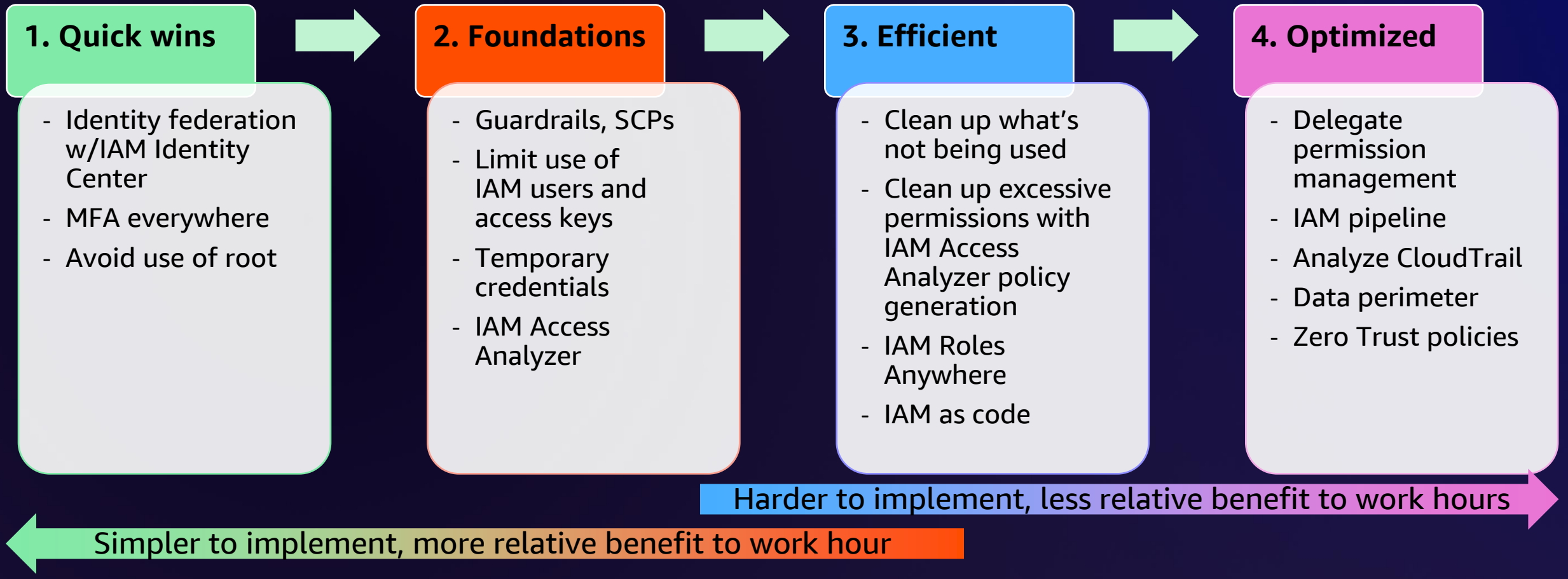


# Phase 4: IAM Optimized

1. Delegate permission management
   - Let developers safely create roles and get out of their way
   - IAM pipeline – establishing a process to create roles
2. Analyze CloudTrail for proper access
   - Denied actions are great signals of a misconfiguration or malicious activity
3. Establish a data perimeter
4. Restrict access using Zero Trust principles
   - Limited time authorization
     Check out Temporary Elevated Access Management (TEAM) solution
     https://aws-samples.github.io/iam-identity-center-team

# Conclusions

# AWS identity strategy: How to prioritize




Note: This is different for each customer's situation; this is our general advice based on our experience

# Related sessions

IDENTITY & ACCESS MANAGEMENT

IAM301: Establishing a data perimeter on AWS, featuring USAA

IAM302: Create enterprise-wide preventive guardrails, featuring Inter & Co

IAM305: Scaling access in IAM Identity Center, feat. Allegiant Airlines

IAM306: Managing hybrid workloads with IAM Roles Anywhere, featuring Hertz

IAM307: Steps towards a Zero Trust architecture on AWS

IAM331-R/-R1: Leaving static access keys behind: A modern path forward

# Additional resources

AWS Security Maturity Model: Security recommendations prioritized and categorized with concise explanations

Blog post: IAM Access Analyzer makes it easier to implement least privilege permissions by generating IAM policies based on access activity

Blog post: Use IAM Access Analyzer to generate IAM policies based on access activity found in your organization trail

Blog post: Establishing a data perimeter on AWS

Step-by-step instructions on how to establish a data perimeter using IAM conditions

# Thank you!

Please complete the session survey in the mobile app

**Dario Goldfarb**

linkedin.com/in/dariolucas/

**Arthur Mnev**

linkedin.com/in/arthurmnev/