AWS re:Inforce

JUNE 13 - 14, 2023 | ANAHEIM, CA

IAM336

Best practices for delegating access in IAM

Liam Wadman

Solutions Architect, AWS Identity
Amazon Web Services

Raymond Ma

Sr. Manager, Product Management, AWS Identity
Amazon Web Services

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What we'll cover

Delegation in AWS Identity and Access Management (IAM)

Service-linked roles (SLRs), service roles, and when to use them

Using iam:PassRole to give IAM roles to AWS resources

Managing access to roles through role trusts

IAM permissions boundaries for self-service access

aws

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

General best practices for IAM

- Use AWS IAM Identity Center (successor to AWS Single Sign-On) or federation for human access
- Avoid using IAM users
- Use IAM roles
- Use a multi-account strategy
- Avoid using root users
- Apply least privilege

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What is delegating access in AWS?

Delegating access: giving an identity or service the ability to assume a role or the permissions to perform an action

Examples

- An AWS service or SAML-federated user assuming a role
- A resource policy granting permissions to a principal
- Passing a role to a service through iam:PassRole
- Creating an SLR for a service to assume

aws

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How do I let my AWS workloads access my AWS resources securely?

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

You give them an IAM role

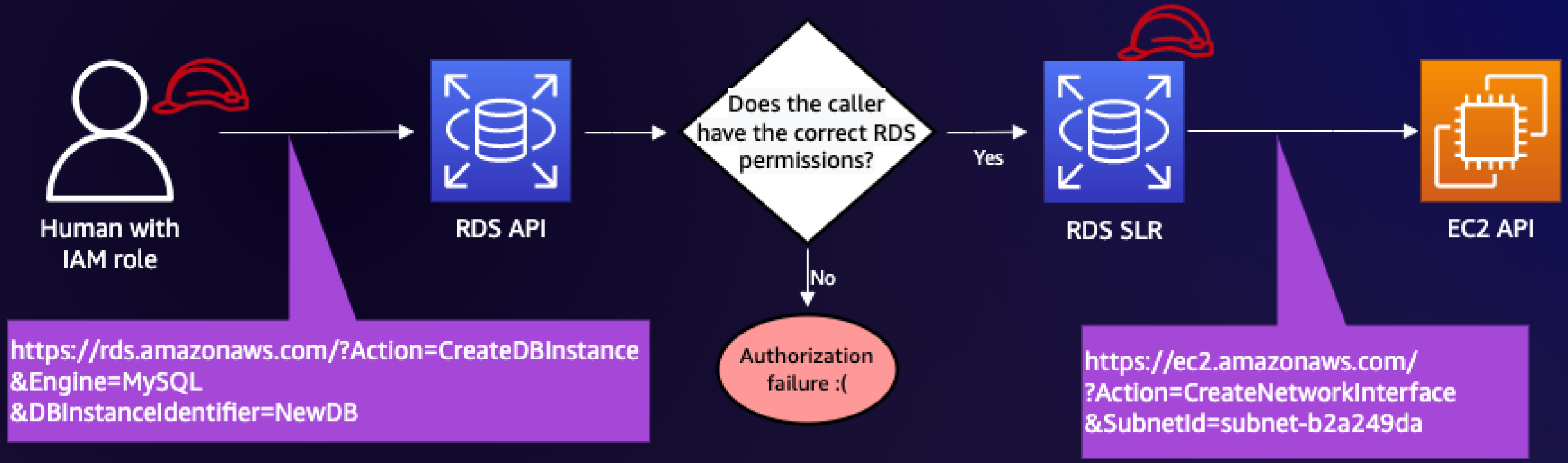
© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Service-linked roles (SLRs): What are they?

- An SLR is used by an AWS service to interact with resources in your account on your behalf
  - Example: Amazon RDS has an SLR for managing networking
- The principal calling Amazon RDS does not need network entitlements
- Entitlements are predefined and managed by AWS

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

SLRs: How they work



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

SLRs: Best practices

- Focus on entitlements to AWS services, not entitlements of SLRs
- iam:CreateServiceLinkedRole allows developer self-service

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Service roles: What are they?

- A service role is a role given to an AWS resource
  - Examples
    - Giving an AWS Lambda function a role to query your Amazon DynamoDB
    - Launching an Amazon EC2 instance with ec2:RunInstances
    - An Amazon S3 replication job uses a role to access your buckets
- Service roles have entitlements and trust policies managed by you
- Service roles are *passed* to AWS resources

aws

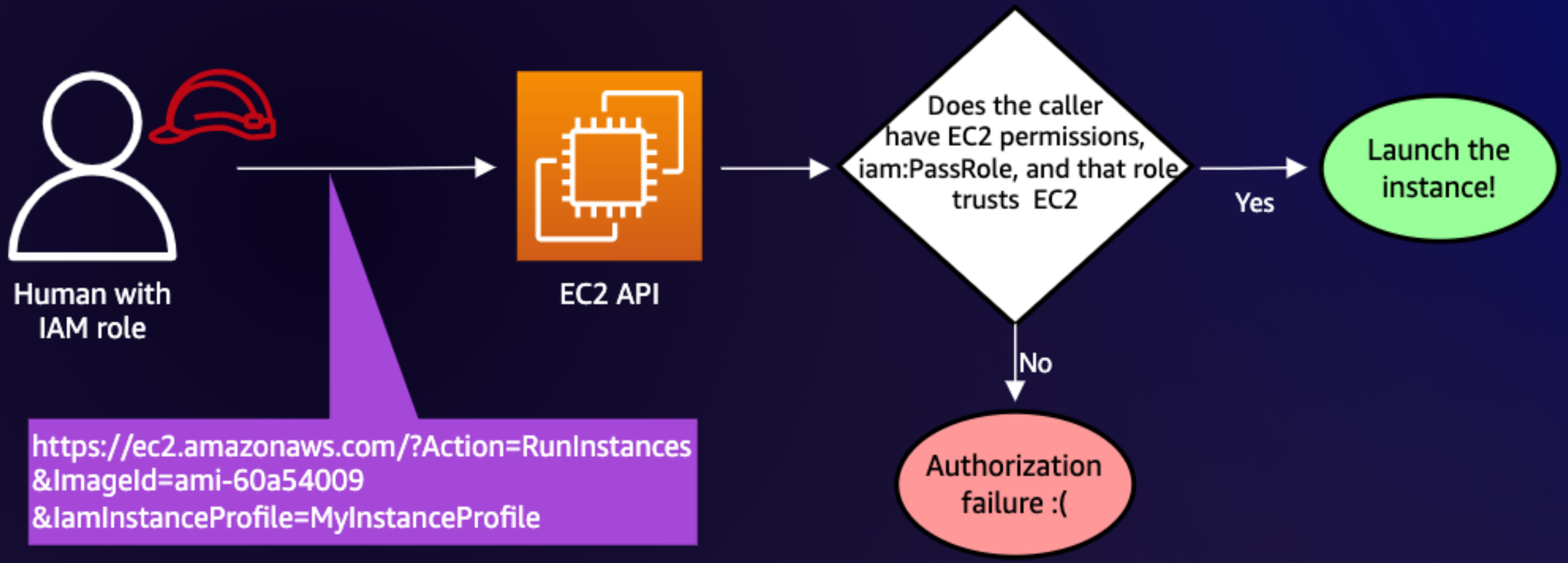
© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

iam:PassRole: What is it?

- An entitlement in AWS – not an API call!
- Allows creating resources with associated roles
- Logged in the call that passed the role

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

iam:PassRole



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

iam:PassRole: The three truths

- A principal must have the iam:PassRole entitlement to pass a role
- The role’s trust policy must allow the service to assume it
- The principal, role to be passed, and resource must be in the same account when passing

aws

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

iam:PassRole: Best practices

Use **IAM paths** to constrain what roles can be passed

```
{
   "Sid": "AllowPassingAppRoles",
   "Effect": "Allow",
   "Action": "iam:PassRole",
   "Resource":"arn:aws:iam::*:role/approles/*"
}
```



aws

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

iam:PassRole: Best practices

Use **wildcards** for account IDs in PassRole statements

```
{
   "Sid": "PrincipalTagInResourcePath",
   "Effect": "Allow",
   "Action": "iam:PassRole",
   "Resource":"arn:aws:iam::*:role/approles/*"
}
```



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

iam:PassRole: Best practices

Place PassRole entitlements in its own policy statement

```
{
  "Sid": "AllowPassingAppRoles",
  "Effect": "Allow",
  "Action": "iam:PassRole",
  "Resource":"arn:aws:iam::*:role/myrole"
}
```



aws

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How does a human/ workload/AWS get access to an AWS role’s credentials?

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

They assume the role

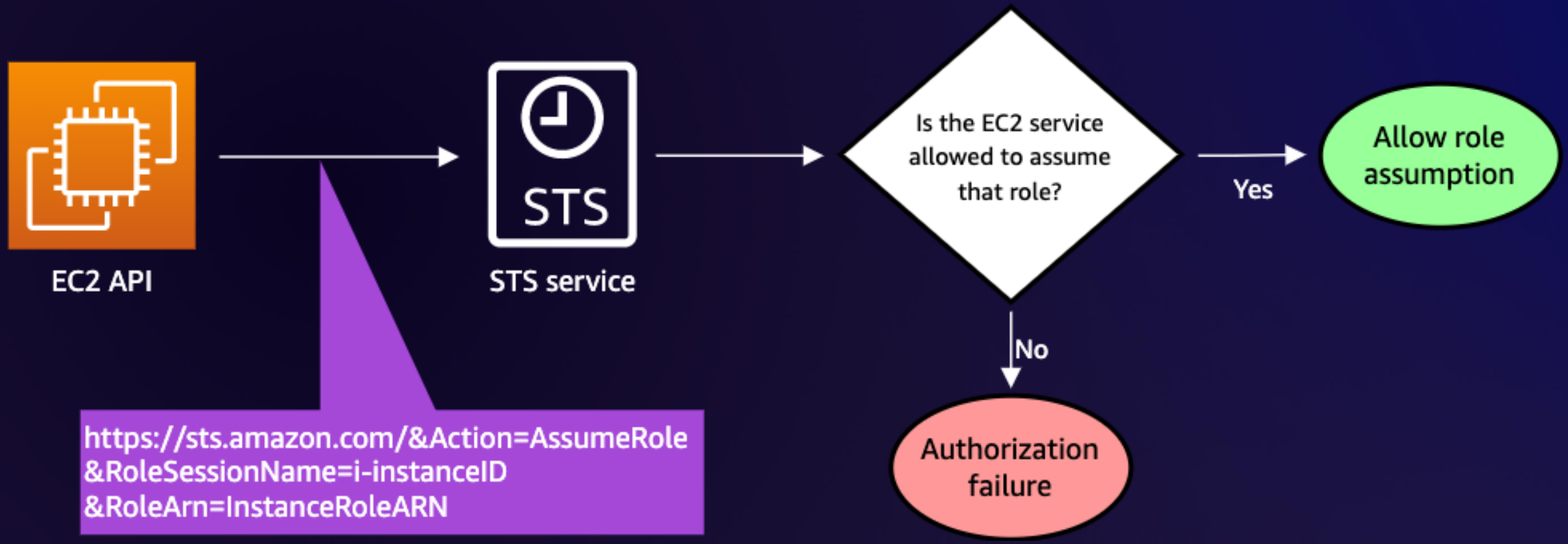
© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AssumeRole: What is it?

- AssumeRole is an AWS API call that returns IAM role credentials
  - AWS services assume service linked and service roles that are *passed to them*
  - SAML assertions, OIDC tokens, and X509 certificates let identities outside AWS assume a role
  - Allow principals from other accounts to assume into your account
- Every role has a trust policy that says who can assume it

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AssumeRole



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AssumeRole: Three truths

- IAM roles can assume other roles (role chaining)
- Roles assumed through SAML and OpenID Connect can only be assumed by IdPs in the same account
- The role trust policy controls who can assume and under what conditions

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AssumeRole: Trust policy examples

Role trust policy allowing cross-account

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::444455556666:role/OtherRole"
  },
  "Action": "sts:AssumeRole"
}
```



aws

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AssumeRole: Trust policy examples

Enforce that a role cannot be assumed from outside your organization, without denying AWS services access to the role

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "*"
  },
  "Action": "sts:AssumeRole",
  "Condition": {
    "StringNotEquals": {
      "aws:PrincipalOrgId": "${aws:ResourceOrgId}"
    },
    "BoolIfExists": {
      "aws:PrincipalIsAWSService": "false"
} } }
```



aws

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

IAM role best practices

- Avoid role chaining within the same account
- Use IAM Access Analyzer to detect cross-account role trusts
- Avoid using :<account_id>:root in trust policies
- Use dedicated roles for the different components of your workloads

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Do you feel comfortable letting your developers create IAM roles and policies in self-service?

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How can I safely let my developers create IAM roles and policies in self-service?

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

You use permissions boundaries

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Permissions boundaries: What are they?

- Permissions boundaries are policies used to limit IAM actions
  - They can explicitly deny actions with a Deny statement
  - They can implicitly deny actions with the lack of an Allow statement
- Use `iam:PermissionsBoundary` condition key to enforce usage

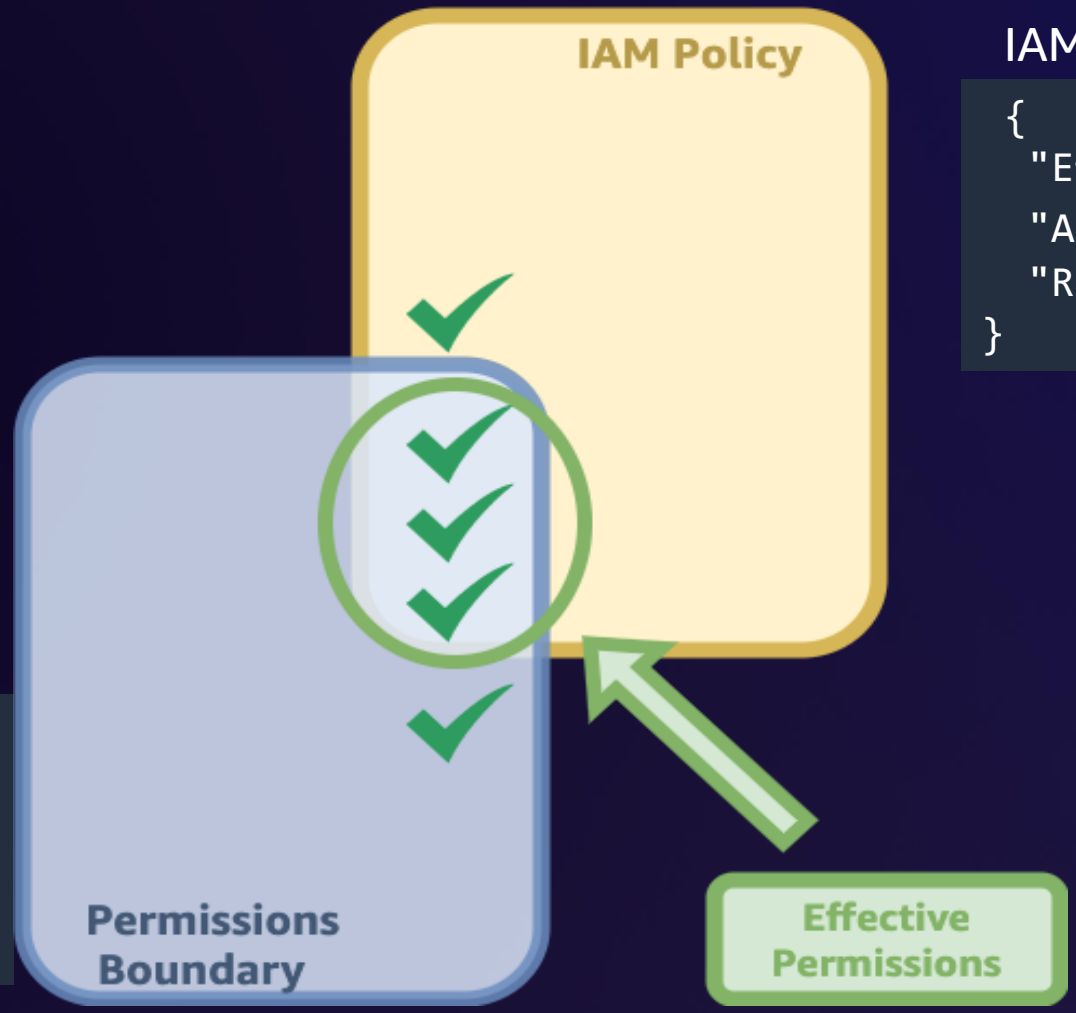
aws

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What is the effective entitlement?

IAM policy

```
{
  "Effect": "Allow",
  "Action": "*",
  "Resource": "*"
}
```



Permissions boundary

```
{
  "Effect": "Allow",
  "Action": "s3:GetObject",
  "Resource": "*"
}
```

aws
© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What is the effective entitlement?

```
{
  "Effect": "Allow",
  "Action": "s3:GetObject",
  "Resource": "*"
}
```



aws

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How can I use permissions boundaries to safely delegate IAM access to my developers?

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

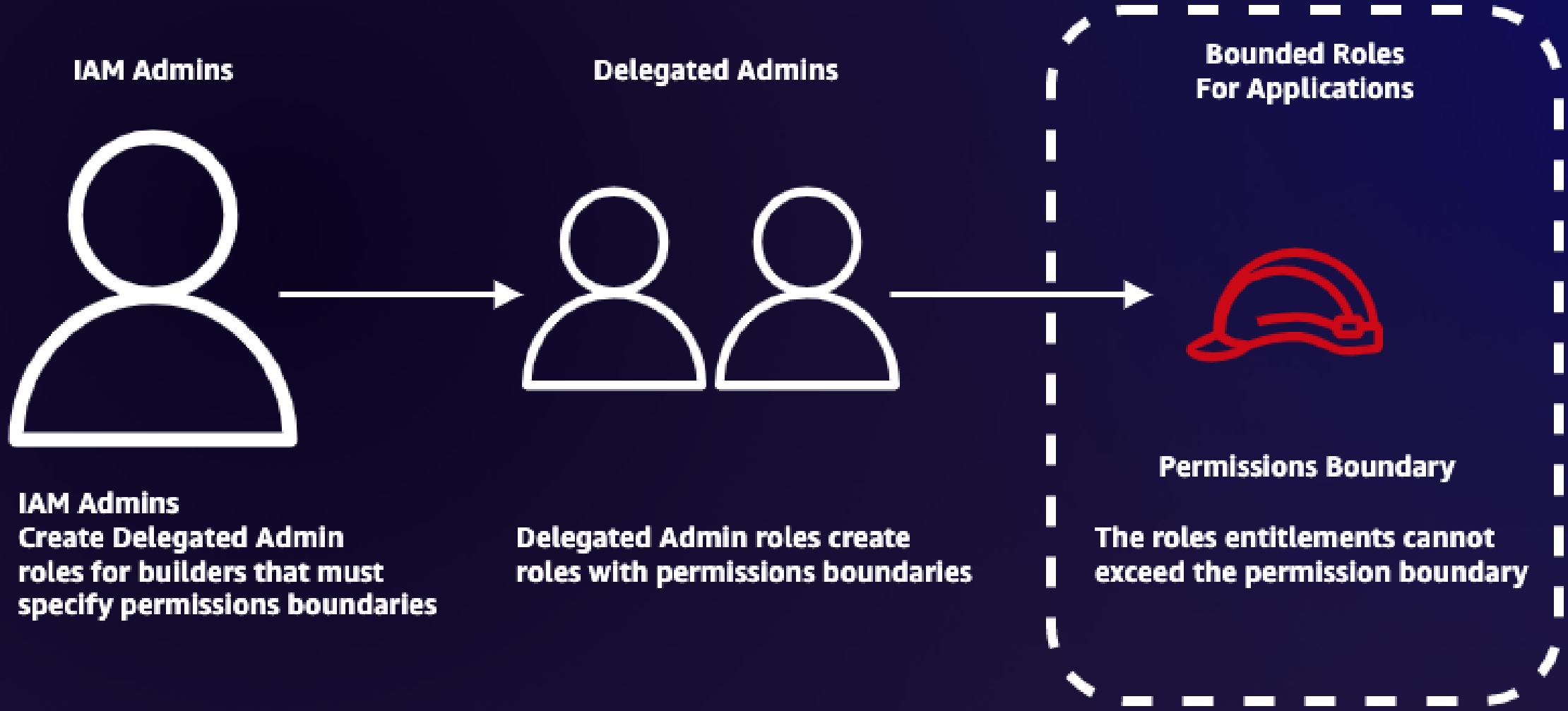
Create a permissions boundary for applications to do application-like actions



https://github.com/aws-samples/example-permissions-boundary

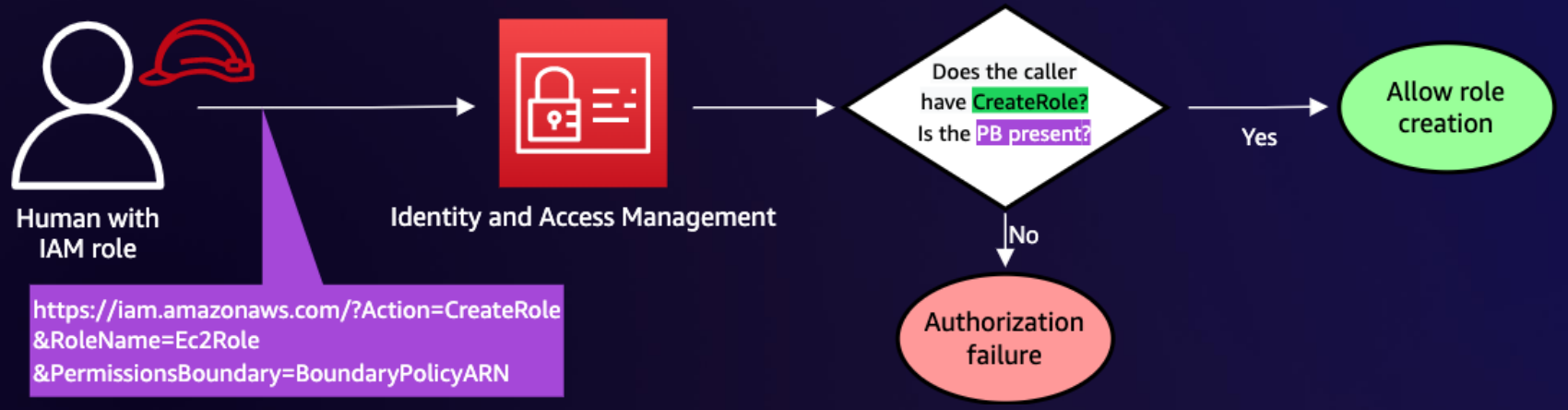
© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Identify your delegated admin roles



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Enforce that permissions boundaries are used




```
{ "Effect": "Deny",
  "Action": [ "iam:CreateRole",
              "iam:PutRolePolicy"
              "iam:AttachRolePolicy" ],
  "Resource": "*",
  "Condition": {
    "StringNotLike": {
      "iam:PermissionsBoundary":
"arn:aws:iam::*:policy/secure/permissionsboundarypolicy"
} } }
```



aws

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Delegating access to developers

SCP mandating that all roles created by developers have a permissions boundary

```
{
  "Effect": "Deny",
  "Action": [ "iam:CreateRole",
              "iam:PutRolePolicy",
              "iam:AttachRolePolicy" ],
  "Resource": "*",
  "Condition": {
    "StringNotLike": {
      "iam:PermissionsBoundary":
        "arn:aws:iam::*:policy/secure/permissionsboundarypolicy"
    },
    "StringLike": {
      "aws:PrincipalArn": "arn:aws:iam::*:role/developer*"
} } }
```



aws

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Delegating access to developers

SCP denying modification of the permissions boundary policy

```
 {
  "Effect": "Deny",
  "Action": [
    "iam:DeletePolicy",
    "iam:CreatePolicyVersion",
    "iam:CreatePolicy",
    "iam:DeletePolicyVersion",
    "iam:SetDefaultPolicyVersion"
  ],
  "Resource": "arn:aws:iam::*:policy/secure/permissionsboundarypolicy",
  "Condition": {
    "ArnNotLike": {
      "aws:PrincipalArn": "arn:aws:iam::*:role/admin*"
} } }
```



aws

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Delegating access to developers

IAM policy requiring all roles be under a specific path

```
{
  "Effect": "Allow",
  "Action": [ "iam:CreateRole",
              "iam:PutRolePolicy",
              "iam:AttachRolePolicy" ],
  "Resource": "arn:aws:iam::*:role/applicationroles/*"
}
```



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Delegating access to developers

SCP denying modification of their own IAM roles

```
 {
  "Effect": "Deny",
  "Action": [
    "iam:PutRolePolicy",
    "iam:AttachRolePolicy",
    "iam:UpdateRole"],
  "Resource": "arn:aws:iam::*:role/developerroles/*",
  "Condition": {
    "StringLike": {
      "aws:PrincipalArn": "arn:aws:iam::*:role/developerroles/*"
} } }
```



aws

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Delegating access to developers

Constrain creation/modification of policies in a specific path

```
{
  "Effect": "Allow",
  "Action": [
    "iam:DeletePolicy",
    "iam:CreatePolicyVersion",
    "iam:CreatePolicy",
    "iam:DeletePolicyVersion",
    "iam:SetDefaultPolicyVersion"
  ],  "Resource": "arn:aws:iam::*:policy/applicationpolicies/*"
}
```



aws

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How do I start with permissions boundaries?

- Keep permissions boundaries wide and reusable
- Don’t include actions that applications shouldn’t need to do
  - PassRole, update IAM, modify virtual private clouds
- Reduce bounded IAM role permissions to what’s needed

aws

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Takeaways

- Manage who/what can pass/assume roles
- Use IAM paths to segregate roles and policies
- Enforce that permissions boundaries are used
- Protect your permissions boundaries policies
- Combine with SCP – see those sessions!

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!



Please complete the session survey in the mobile app

Liam Wadman
liwadman@amazon.com

Raymond Ma
raysma@amazon.com

Permissions boundary GitHub

aws

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.