
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

SEC313

Harness the power of IAM policies & rein in permissions with Access Analyzer

Brigid Johnson (she/her)

GM, AWS Identity and Access Management Access Analyzer
AWS

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The next hour

Harness the power of AWS permissions

- Permissions and access controls
- Policy evaluation and enforcement — Policy example — Pop quiz!
- Conditions — Policy example

Reining in permissions

- Setting permissions — Demo! — Policy example — Pop quiz!
- Verifying permissions — Demo!
  - IAM roles deep dive — Policy example — Pop quiz!
- Refining permissions — Demo!

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Harness the power of AWS permissions

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What we hear from customers

Business needs an accountable security posture

Teams need flexibility to explore and innovate

Need agility to move fast



Teams need to own security

Central teams must prevent powerful actions in the cloud

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Permissions in AWS

Who



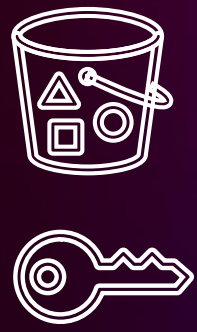
Developers and applications

Can access



Permissions

What



Resources

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Permissions and the shared security model

The job of AWS



Robust access controls



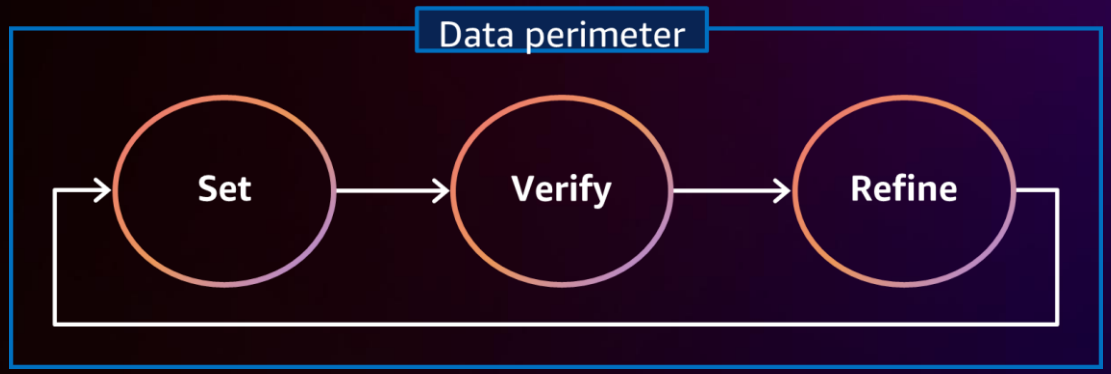
Enforce



Guidance

Your job

Establish access controls
based on your business needs



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Access controls: Overview

Type	Examples
Policies Grant or restrict access to identities, services, actions, and resources under specific conditions	Identity policies Service control policies (SCPs)
Restrictions Single, sharp access controls that restrict access, regardless of policies	Block Public Access (Amazon S3)
Service specific Access controls to service a specific need for a service	AMI sharing RDS sharing

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Access controls: Policies

```
{
 "Statement":[{
"Effect":"effect",
"Principal":"principal",
"Action":"action",
"Resource":"arn",
"Condition":
{  "condition":{
       "key":"value" }
}
  }
 ]
}
```

Principal – the entity that is allowed or denied access

```
"principal":"AWS":"arn:aws:iam::123456789012:user/username"
```

Action – type of access that is allowed or denied access

```
"action":"secretsmanager:getSecret"
```

Resource – the Amazon resources the action will act on

```
"Resource":"arn:aws:secretsmanager:us-east-2:111122223333:secret:pickles"
```

Condition – the condition under which the access defined is valid

```
"StringEquals": {"aws:CalledViaFirst": "lambda.amazonaws.com "}
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Access controls: Policy types and use cases

 **Service control policies (SCPs)**	Restrict access only Applies to organizations	Restrict powerful actions except for admins
 **Permissions boundaries**	Restrict access only Applies to IAM entities	Developers manage roles with limitations
 **IAM permissions policy**	Grant or deny access Applies to IAM entities	Grant only the actions your role uses
 **Scoped-down policies**	Restrict access only Applies to IAM role sessions	Restrict access for specific role sessions
 **Resource-based policies**	Grant or deny access Applies to resources	Grant direct, cross-account access
 **VPC endpoint policies**	Grant or deny access Applies to endpoints	Controls who can use the VPC endpoint to access the endpoint service

aws

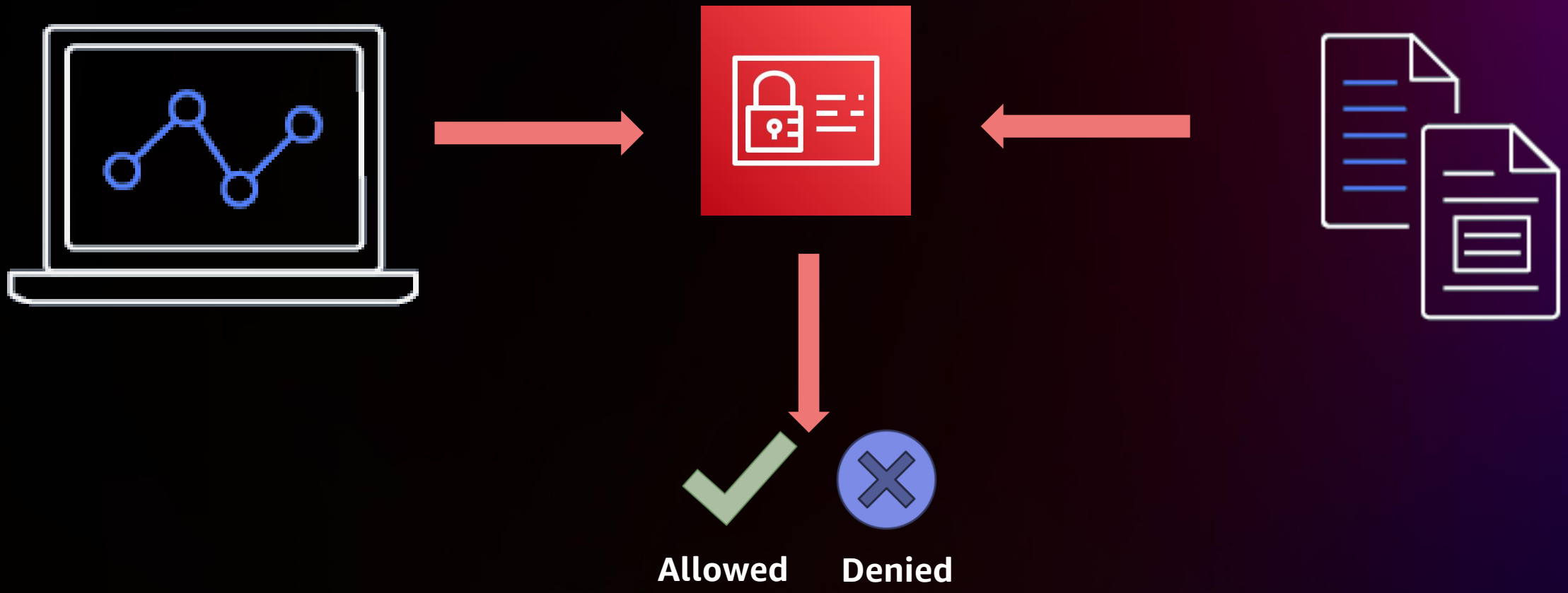
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Enforce: Context and policies to evaluate

Context of AWS request

IAM

Policies evaluated



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Enforce: Authorization context

Access is allowed

```
Principal:  AROADBQP57FF2AEXAMPLE
Action:     secretsmanager:getSecret
Resource:   arn:aws:secretsmanager:*:123:*
Context:
    aws:UserId=AROADBQP57FF2AEXAMPLE:BobsSession
    aws:PrincipalAccount=123456789012
    aws:PrincipalOrgId=o-example
    aws:PrincipalARN=arn:aws:iam::123456789012:role/Bob
    aws:MultiFactorAuthPresent=false
    aws:CurrentTime=...
    aws:EpochTime=...
    aws:SourceIp=...
    aws:PrincipalTag/dept=123
    aws:PrincipalTag/project=pickles
    aws:ResourceTag/project=pickles
```

```
“Effect”: ”Allow”,
“Action”: “secretsmanager:getSecret”,
“Resource”: ”*”,
“Condition”: {
   “StringEquals”: {
      “aws:ResourceTag/project”: “pickles”
   }
}
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Enforce: Pop quiz!

Which one doesn’t belong?

1. Allow
2. Maybe allow?
3. Deny

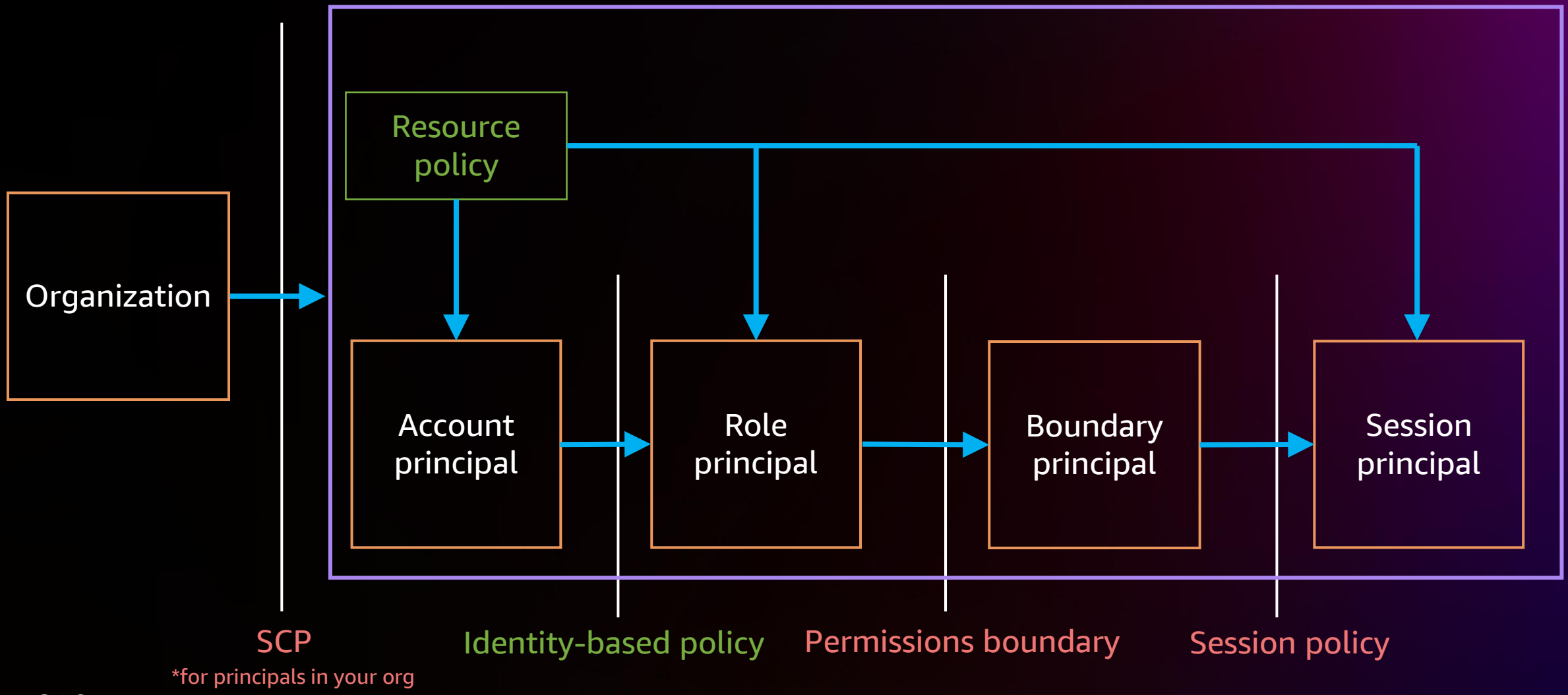


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Enforce: Policy evaluation

1. An explicit ALLOW must exist at all levels

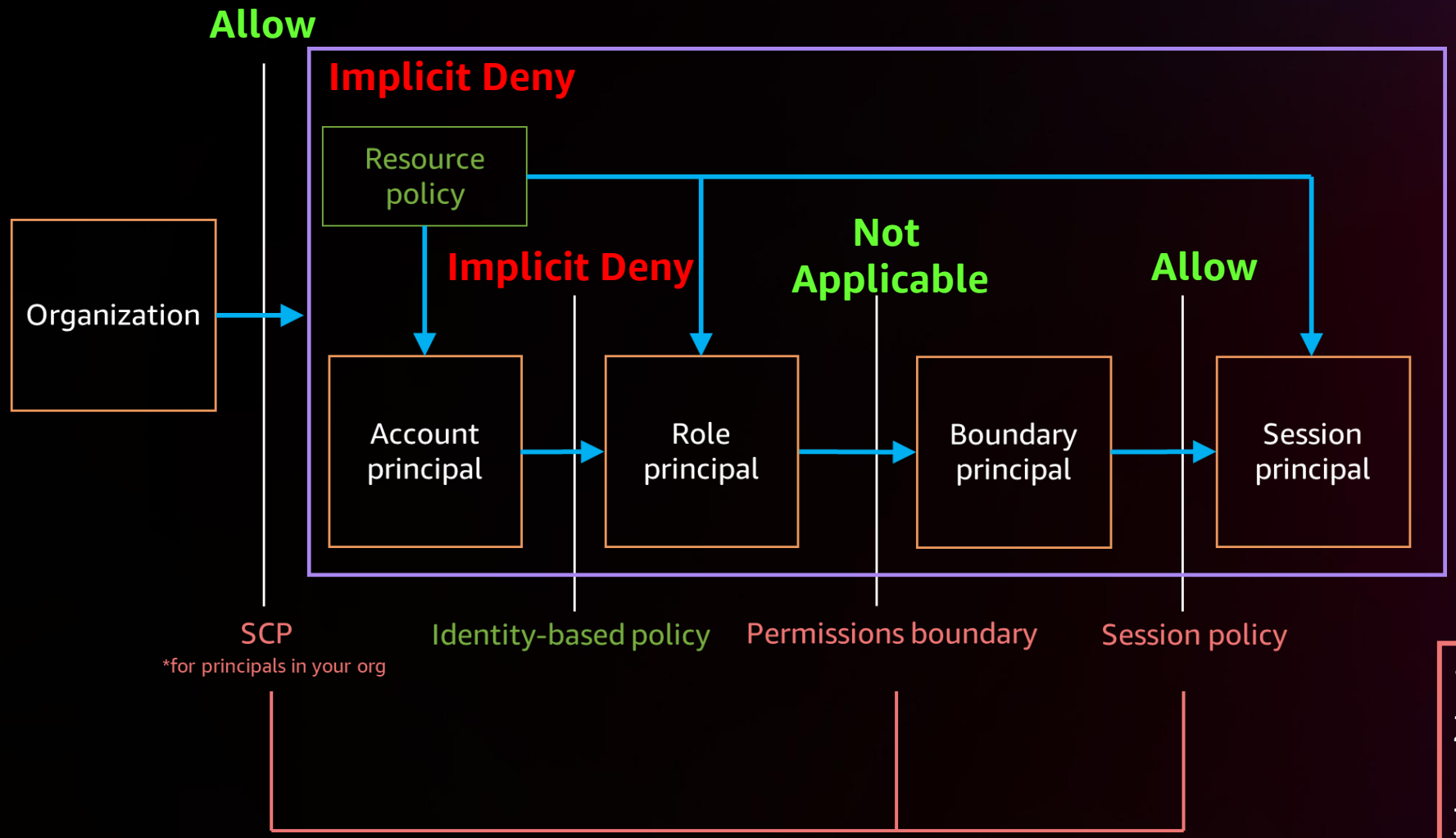
2. An explicit DENY always overrides an ALLOW



*for principals in your org

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Enforce: What you start with



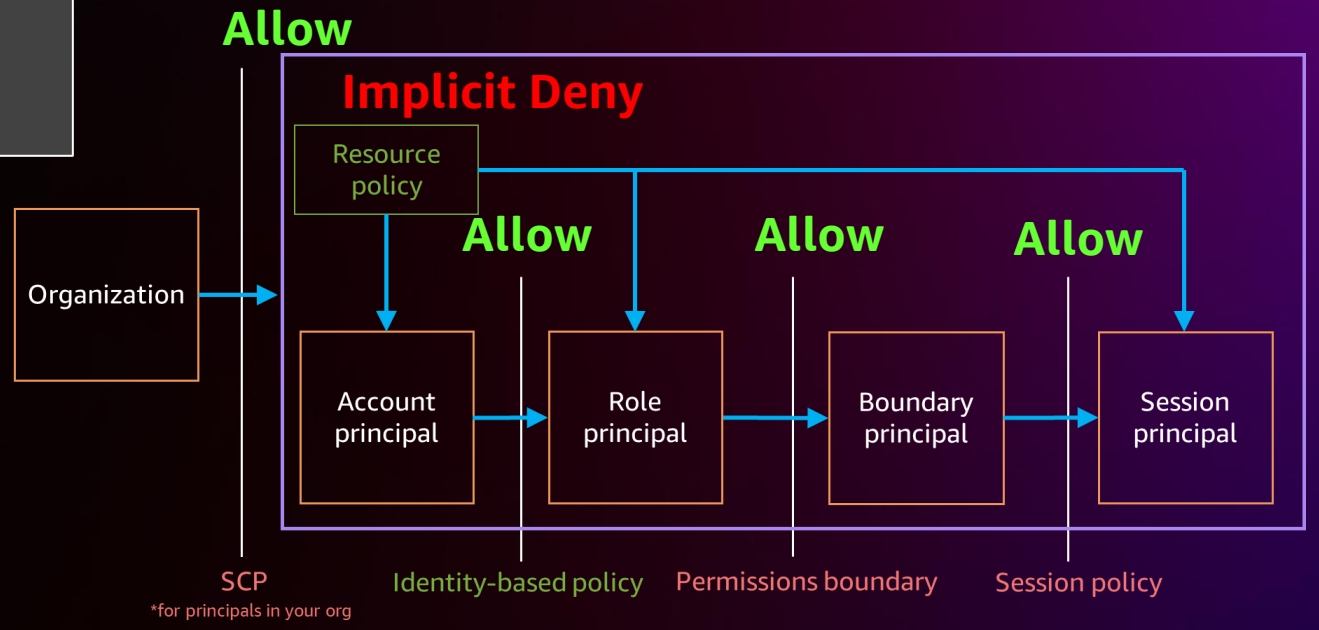


1. Access is denied by default
2. Grant access based on identity and resource only
3. If you don't apply SCP or permission boundary, then identity or resource policy describes access

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Enforce: Policy evaluation – Example 1

```
Identity Policy (Pickles' role, acct1-Barn)
        Allow
        secretsmanager:getSecret
        PicklesTreats
```



Final result?

Allow



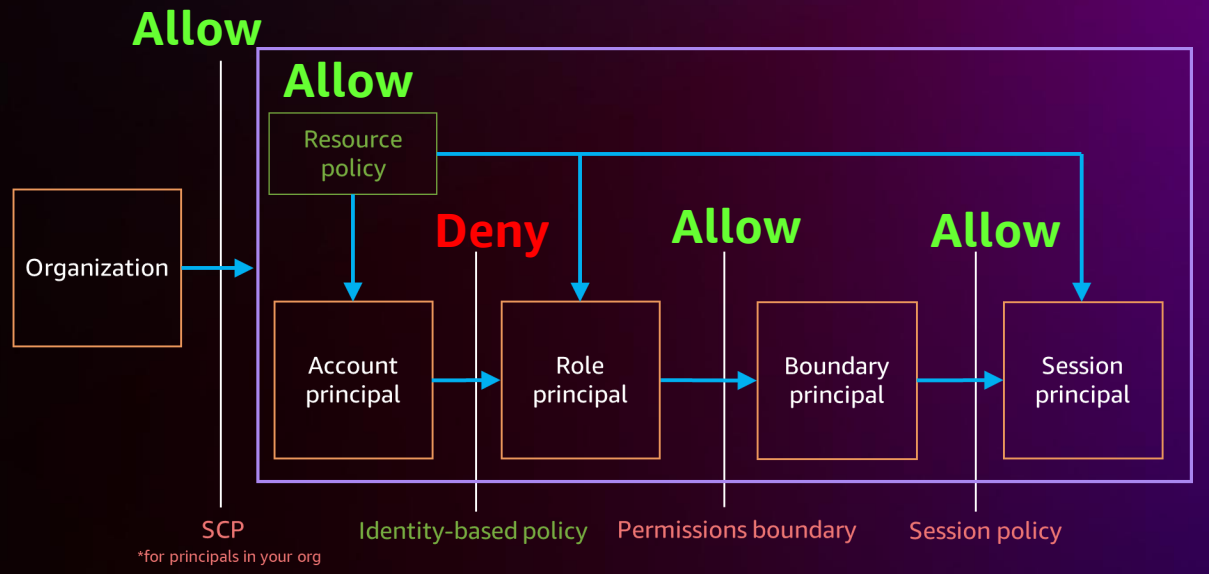
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Enforce: Policy evaluation – Example 2

```
Identity Policy (Pickles role, acct1-Barn)
        Deny
        secretsmanager:getSecret
        PicklesTreats
```

```
Identity Policy (Brigid role, acct1-Amzn)
        Allow
        secretsmanager:getSecret
        PicklesTreats
        Condition: OrgID is Barn
```

```
Resource Policy (PicklesTreats, acct1-Barn)
        Allow
        Roles[Pickles, Brigid]
        secretsmanager:getSecret
        PicklesTreats
```



Final result?

Deny



Allow



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

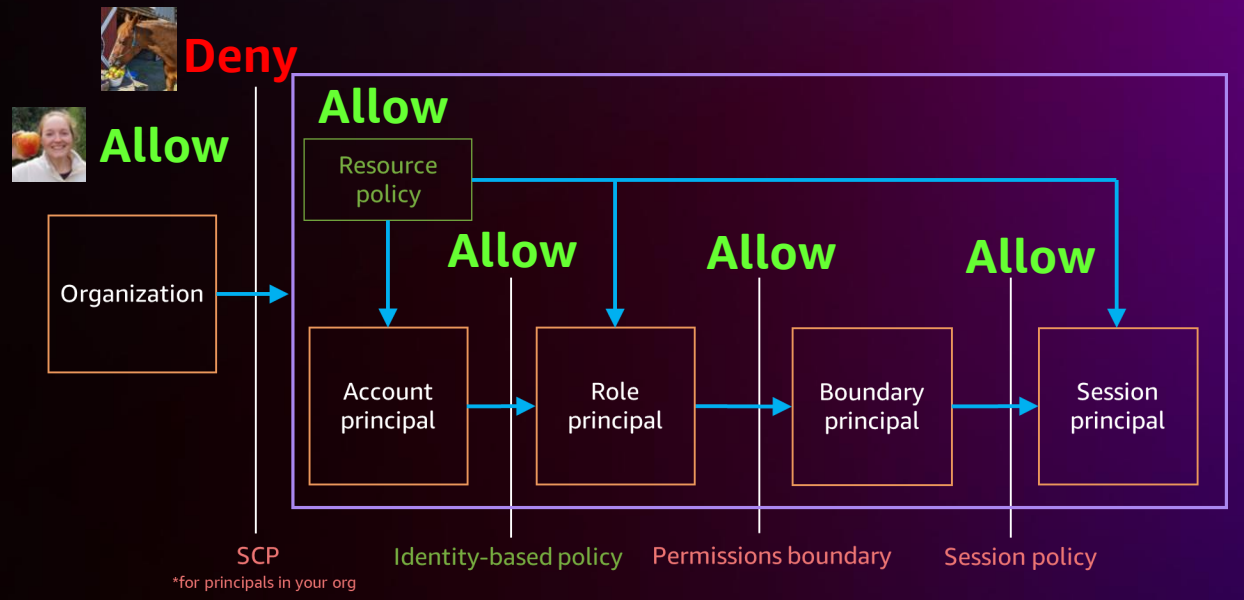
Enforce: Policy evaluation – Example 3

```
Identity Policy (Pickles role, acct1-Barn)
        Allow
        secretsmanager:getSecret
        PicklesTreats
```

```
Identity Policy (Brigid role, acct1-Amzn)
        Allow
        secretsmanager:getSecret
        PicklesTreats
        Condition: OrgID is Barn
```

```
Resource Policy (PicklesTreats, acct1-Barn)
        Allow
        Roles[Pickles, Brigid]
        secretsmanager:getSecret
        PicklesTreats
```

```
SCP (Org Barn)
        Deny
        secretsmanager:getSecret
        *
```



Final result?

Deny



Allow



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

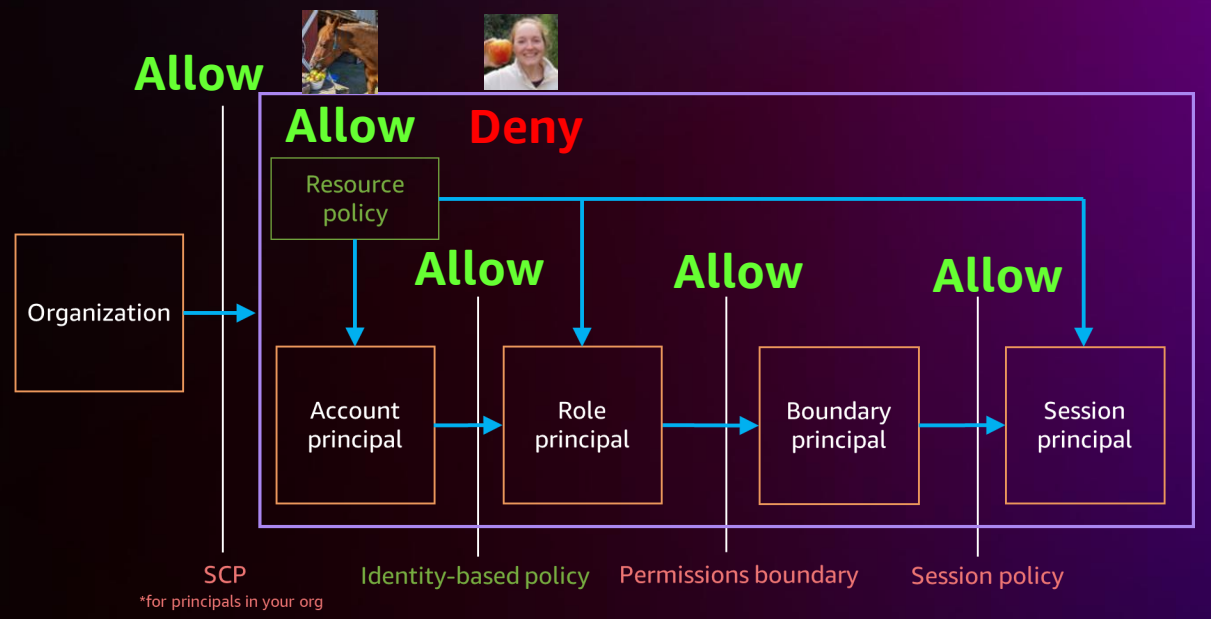
Enforce: Policy evaluation – Example

```
Identity Policy (Pickles role, acct1-Barn)
        Allow
        secretsmanager:getSecret
        PicklesTreats
```

```
Identity Policy (Brigid role, acct1-Amzn)
        Allow
        secretsmanager:getSecret
        PicklesTreats
        Condition: OrgID is Barn
```

```
Resource Policy (PicklesTreats, acct1-Barn)
        Allow
        Roles[Pickles, Brigid]
        secretsmanager:getSecret
        PicklesTreats
        Condition: principals in Org Barn
```

```
SCP (Org Barn)
        Allow
        secretsmanager:*
        *
```



Final result? Allow Deny





aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Enforce: Pop quiz!

Use SCPs to restrict access to which principals?

1. All root users
2. All users, roles, and root users in my organization
3. All roles in any organization in AWS
4. Roles named Pickles



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Conditions: Deep dive

```
"Condition" : { "{condition-operator}" : { "{condition-key}" : "{condition-value}" }}
```

Specify the type of operator used to compare the value from the context to the value in the policy

`StringEquals`
`StringLike`



If you don't have any wildcard characters, use stringEquals

Use predefined keys to look up the value in the context for policy comparison; you can use global condition keys or service-specific keys

`PrincipalIsAWSService`
`ResourceTag/project`



You cannot put a wildcard character in the condition key

The value you define that is based on your permissions requirements

`true`
`pickles`
`${PrincipalTag/project}`



A single-valued condition key can be used as a variable for condition values

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Conditions: Using multiple conditions

Every policy statement stands on its own; if there is a match, the effect applies

AND

All conditions met together

Same statement

OR

Any part of conditions met

Separate statements

OR

AND

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":“CreateSLRs",
         "Effect":"Allow",
         "Action":"iam:CreateServiceLinkedRole",
         "Resource":"*",
         "Condition":{
            "StringEquals":{
               "aws:PrincipalServiceName":"access-analyzer.amazonaws.com"
            }
         }
      },
      {
         "Sid":“RoleCreate-CFN",
         "Effect":"Allow",
         "Action":"iam:CreateRole",
         "Resource":"*",
         "Condition":{
            "StringEquals":{
             "iam:PermissionsBoundary":"arn:aws:iam::524290187879:
                                        policy/pasture-audit-broad",
               "aws:CalledViaFirst":"cloudformation.amazonaws.com",
               "aws:RequestTag/project":["pickles", "bubble"]
            }
         }
      }
   ]
}
```

OR

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Conditions: Noteworthy condition keys

ResourceTag	Allow actions, but only if these tags exist	Attribute-based access control
RequestTag	Create resources, but only with these tags	Attribute-based access control
NEW ResourceOrgId	AWS organization ID of the resource being accessed	Only access resources in my org
NEW ResourceAccount	AWS account ID of the resource being accessed	Only access resources in trusted account
PrincipalOrgID	Allow actions, but only if the principal is from my organization	A condition for most resource policies
PrincipalArn	Deny actions for all principals except this ARN	Use in SCPs to deny with exceptions
PrincipalIsAWSService	Restrict access to corporate IP and AWS services	Use in SCPs when restricting SourceIP

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Conditions: Resource conditions example

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyS3AccessOutsideMyBoundary",
      "Effect": "Deny",
      "Action": [
        "s3:*"
      ],
      "Resource": "arn:aws:s3:::*/*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceOrgID": "${aws:PrincipalOrgID}"
        }
      }
    }
  ]
}
```



Use ResourceOrgID in policies that control access to principals such as identity-based or SCPs

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Recap

The job of AWS



Robust access controls



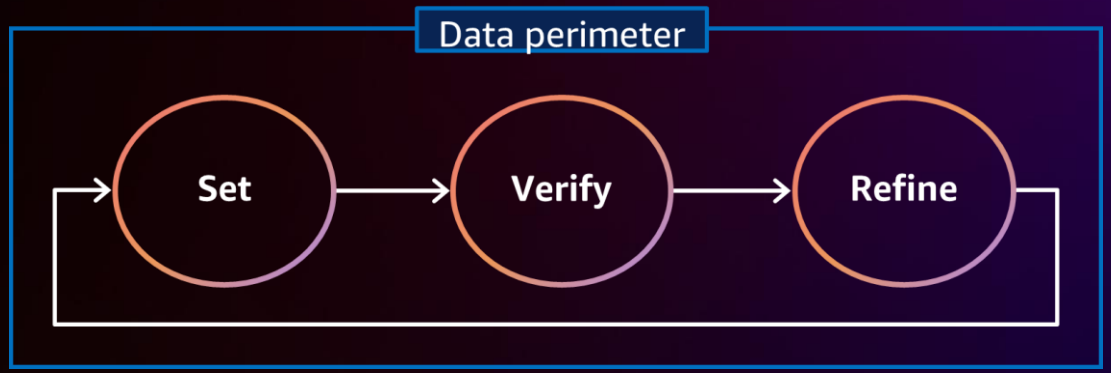
Evaluate and enforce



Guidance

Your job

Establish access controls based on your business needs



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Reining in permissions

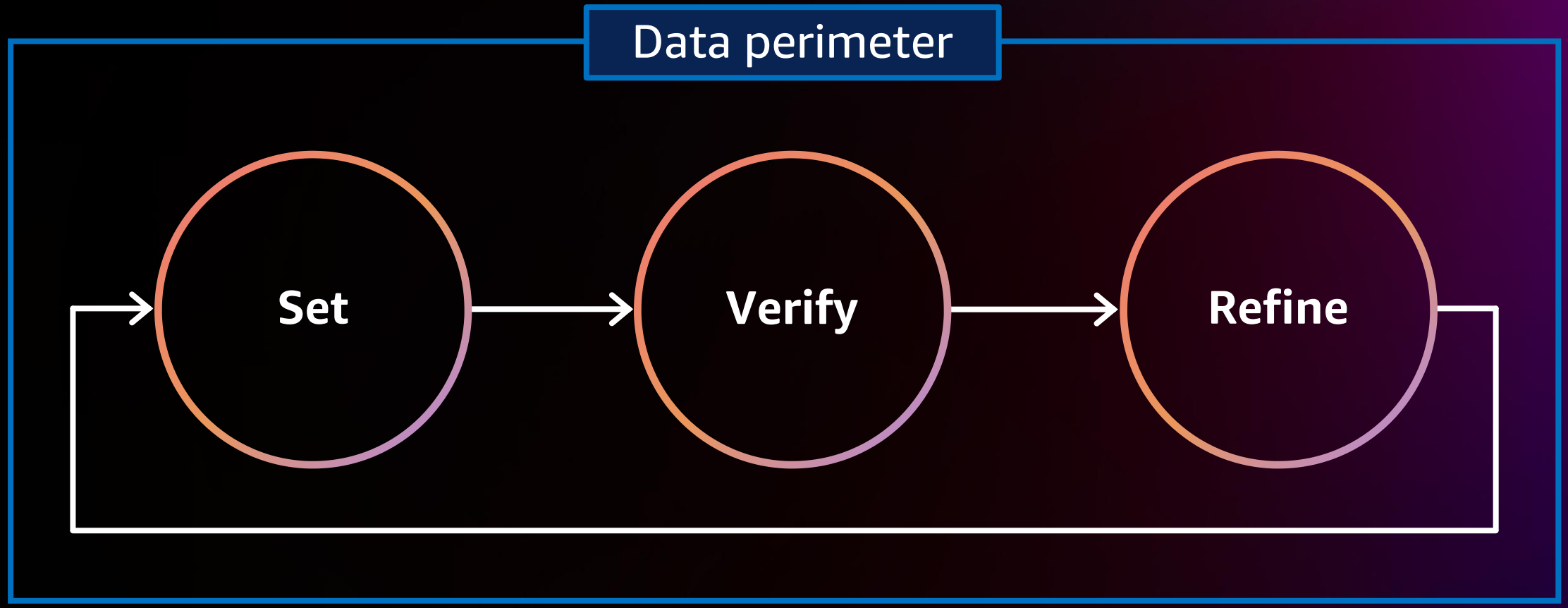
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Least privilege is a journey



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

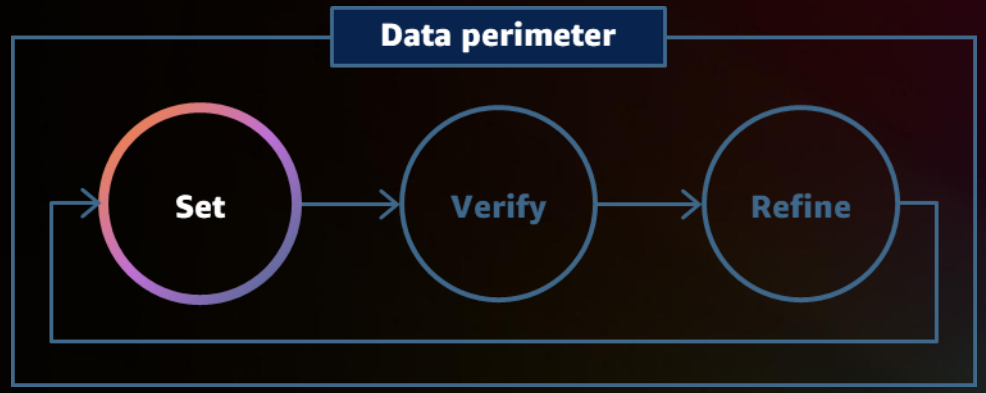
Getting to the right permissions



Permission lifecycle

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Setting permissions



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Who applies permissions?

Central security teams

Establish guardrails and perimeters

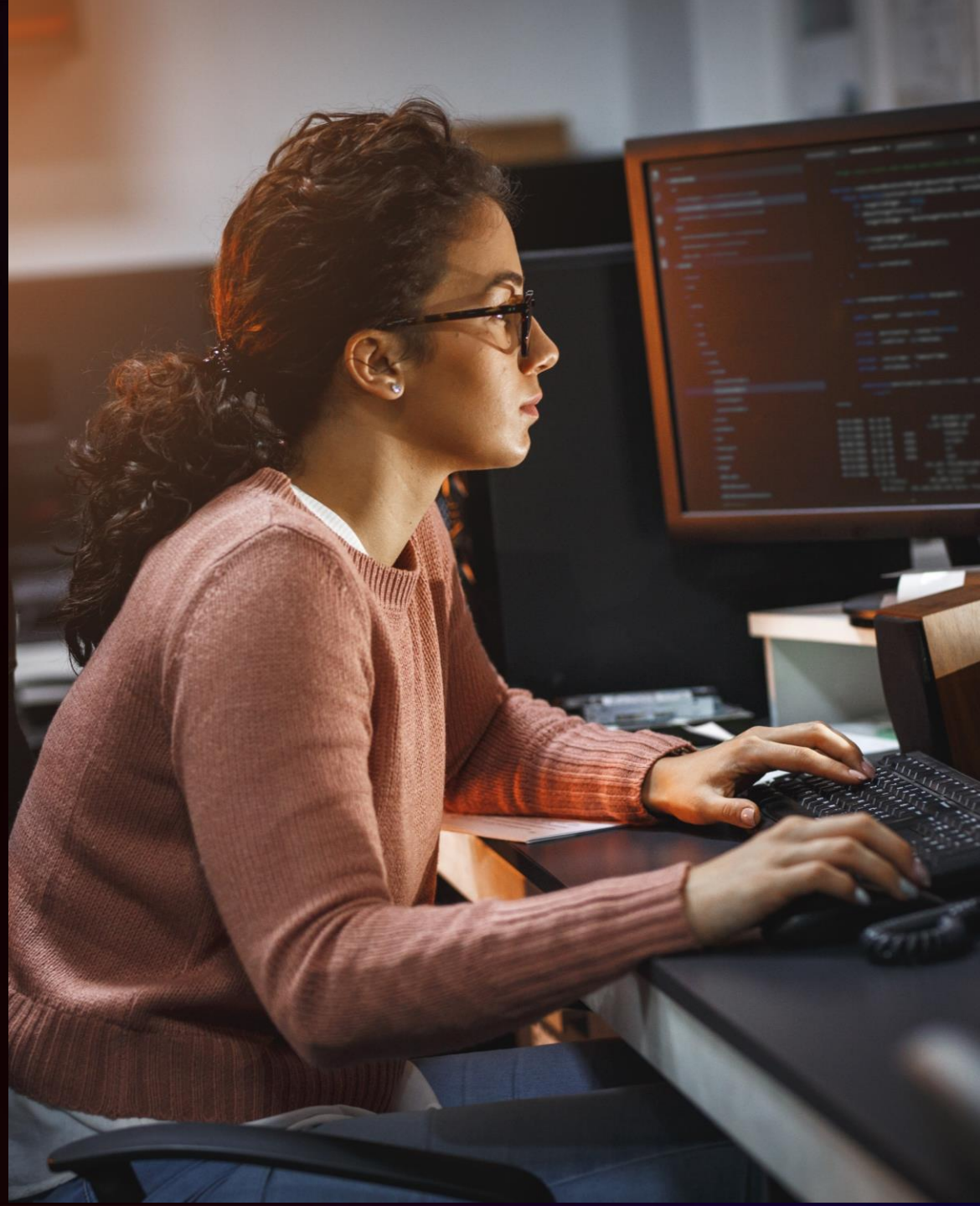
Verify and audit permission practices

Provide tools and templates

Engineering teams

Establish fine-grained permissions during development

Refine permissions up to production and respond to findings



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Set: Modes to establish permissions

AWS managed



Explore AWS, get started with defaults, and use templates

Tools to right-size



Generate and customize using policy generation with IAM Access Analyzer

DIY



Author custom policies and validate by using policy validation with IAM Access Analyzer

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Set: AWS managed mode

AWS managed policies

Get started with use case and service defaults using AWS managed policies; then move toward least-privilege permissions

AWS Cloud Development Kit (AWS CDK)

Define cloud infrastructure as code; preconfigure cloud resources with proven defaults

AWS SAM serverless connectors

Grant the appropriate level of access to the resources in their application's infrastructure

Composes purpose-built policies to facilitate data and event flowing between resources

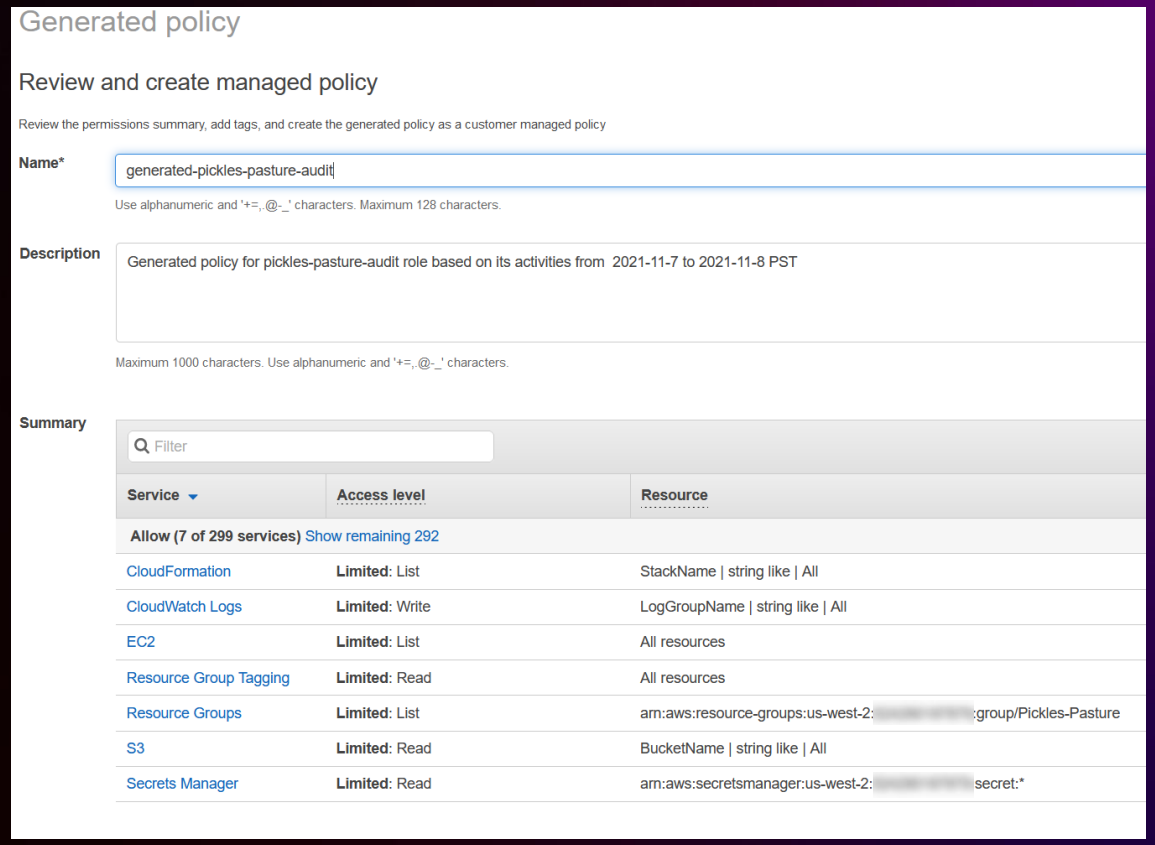
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Set: Policy generation with IAM Access Analyzer

GET TO THE RIGHT PERMISSIONS BY ANALYZING YOUR ACCESS ACTIVITY IN AWS CLOUDTRAIL

1. Run your application or task
2. Request a policy from IAM Access Analyzer
3. IAM Access Analyzer gets to work
4. Customize further and apply



NEW Action-level permissions for 140 services

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Set: Policy generation in action

Demo!

Goal

Right-size the permissions for a CloudFormation role used to set up Pickles's barn in a new region

Demo steps

1. Select the role used for CloudFormation
2. Request a policy from IAM Access Analyzer
3. Customize by specifying resources and additional actions
4. Apply the new right-sized policy



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Set: Policy validation with IAM Access Analyzer

AUTHOR SECURE AND FUNCTIONAL POLICIES WITH ACTIONABLE GUIDANCE

Security
A policy contains access that can be overly permissive

Errors
A policy contains errors that prevent the policy from functioning



IAM Access Analyzer

General warnings
A policy doesn't conform to best practices, but the issues are not security risks

Suggestions
A policy contains statements to be improved but doesn't impact the permissions of the policy

We are always adding new checks! Let us know if you have an idea

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

IAM Access Analyzer has your back

Over 100 checks

- Identity policies in IAM console
- Bucket policies in S3 console
- NEW Role trust policies in IAM console
- Terraform integration
- Unlimited possibilities with API

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Set: Policy validation (round 1)

Demo!

Goal

Update a custom policy to remediate security warnings and errors

Demo steps

1. Navigate to the policy editor
2. Assess findings from IAM Access Analyzer
3. Make edits to the policy
4. Repeat steps 2 and 3 until all findings are clear

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[ "cloudtrail:GetTrail",
                    …
            "lambda:ListFunctions" ],
         "Resource":"*"
      },      {
         "Effect":"Allow",
         "Action":[ "iam:GetRole",
                    …
            "iam:ListRoleTags" ],
         "Resource":"arn:aws:iam::524290187879:role/pickles-*"
      },
      {
         "Sid":"RoleCreateCFN",
         "Effect":"Allow",
         "Action":"iam:CreateRole",
         "Resource":"*",
         "Condition":{
            "StringEquals":{
"iam:PermissionsBoundary":"arn:aws:iam::524290187879:
policy/pasture-audit-broad",
                  "aws:RequestTag/project":["pickles", "bubble"]
}         }      }   ] }
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Guidance: Pop quiz

Which advice should you **not** follow?

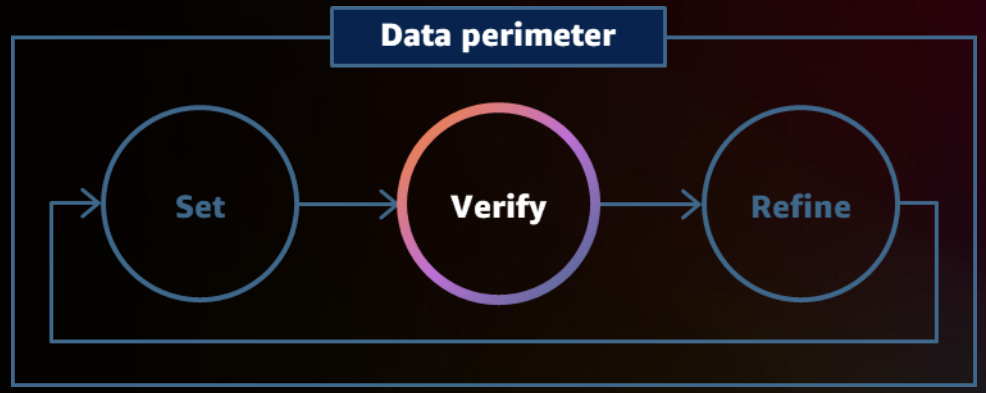
1. Use policy validation as your author policies
2. Integrate policy validation in CI/CD pipelines
3. Keep policy check ideas to yourself
4. Pay close attention to security warnings and errors



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Verifying permissions



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Inspect access to verify

What to look for

Public and cross-account access

```
Allow
Principal *
```

```
Allow
Personal Account 123
```

Access to add permissions

```
Allow
Iam:Passrole
*
```

```
Allow
S3:putBucketPolicy
*
```

```
Allow
iam:attachpolicy
*
```

```
Allow
Lambda:UpdateFunctionUrlConfig
Lambda:AddPermission
*
```



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deep dive: Understanding IAM roles

How roles work

1. **Create** roles with access to take actions on resources in an account

2. **Specify** entities to assume the role; includes identity providers, services, etc.

3. Entities call AWS Security Token Service (AWS STS) to **assume** the role

4. STS **verifies** the entity is allowed to assume the role based on the request and **returns temporary credentials**

5. Entity uses credentials to **make AWS requests**



Benefits

Temporary credentials

Fine-grained access to control entities that can assume roles

Fine-grained permissions once assumed

Auditable using AWS CloudTrail

AWS services use it to access as well!



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deep dive: Types of IAM roles

Service-linked roles (SLR)	Roles linked directly to an AWS service; permissions predefined to only those required by the service to call other AWS services on your behalf	IAM Access Analyzer Amazon EC2 Auto Scaling	Grant access to create SLRs for services you use
Service roles	Roles an AWS service assumes to perform actions on your behalf; you manage permissions	Amazon EC2 Lambda	Specify the service in the trust policy
Federation roles	Roles that trust an identity provider and users can assume from the identity provider	AWS IAM Identity Center	Set source identity
Application roles	Roles used by your application to access resources in or across accounts	Compute or third-party applications	Set ExternalID
IAM roles everywhere	Roles used by non-cloud entities to gain access to the cloud	On-premises entities	Review trust policy

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deep dive: Trust policies

Effect — Grant or restrict access

Pro-tip: Rely on allows

Principal — The entity who can assume the role

Options: Account, Federated, or a Service

Action — Actions allowed by assuming entity

- AssumeRole
- AssumeRoleWithSAML
- AssumeRoleWithWebIdentity
- SetSourceIdentity
- TagSession
- GetAccessKeyInfo
- GetCallerIdentity
- GetFederationToken
- GetSessionToken

Condition — Must be met for credential to be returned

- SAML:aud
- sts:SourceIdentity
- sts:ExternalId
- aws:PrincipalArn
- aws:RequestTag/<your tag>

```
{
 "Statement":[{
"Effect":"effect",
"Principal":"principal",
"Action":“sts:assumeRole",
"Condition":
{  "condition":{
      "key":"value" }  }
  }
 ]
}
```

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deep dive: Trust policy an example

```
{  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::012345678910:oidc-provider/token.actions.githubusercontent.com"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "token.actions.githubusercontent.com:aud": "sts.amazonaws.com",
          "token.actions.githubusercontent.com:sub": "repo:octo-org/pickles-repo:ref:refs/heads/pickles-branch"
        }      }    }  ] }
```

Principal	What entity do you trust?	Oidc-provider: github
Action	What do you trust them to do?	Assume role with web identity
Condition	What conditions must be met for trust?	Subject is from a specific repo Audience is sts.amazonaws.com

aws © 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Tools to inspect roles

- Public findings
- Cross-account findings
- Policy validation for role trust policies
- Role last used
- AWS CloudTrail
- Amazon Detective



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Roles deep dive: Pop quiz

To grant access to a third-party partner, you should

1. Allow full admin access
2. Rely on SourceIP restrictions
3. Use the same role for multiple third parties
4. Use ExternalID condition



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Verify: Findings with IAM Access Analyzer



IDENTIFY AND REMEDIATE PUBLIC AND CROSS-ACCOUNT ACCESS

1. **Enable** IAM Access Analyzer in your account or organization
2. Continuously **monitors and reviews** access controls across 14 resource types — NEW
3. Uses **automated reasoning** to determine public or cross-account access
4. **Generates findings** for you to review and determine whether they match your intent
5. **Verify** public and cross-account access by previewing access as you modify resource policies



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Verify: AWS uses automated reasoning

Reasoning is the process of **applying logic** to infer **new conclusions** from known facts about a system

Automated reasoning uses computers to unlock the ability for us to **reason about an infinite number of paths** in seconds

Proves the correctness of **universal statements** for complex systems

IAM Access Analyzer uses automated reasoning to provide you with a **comprehensive analysis of external access** to your resources

Review **public and cross-account findings** in IAM Access Analyzer to verify that existing access controls meet your intent

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Verify: Inspect all parts of IAM roles

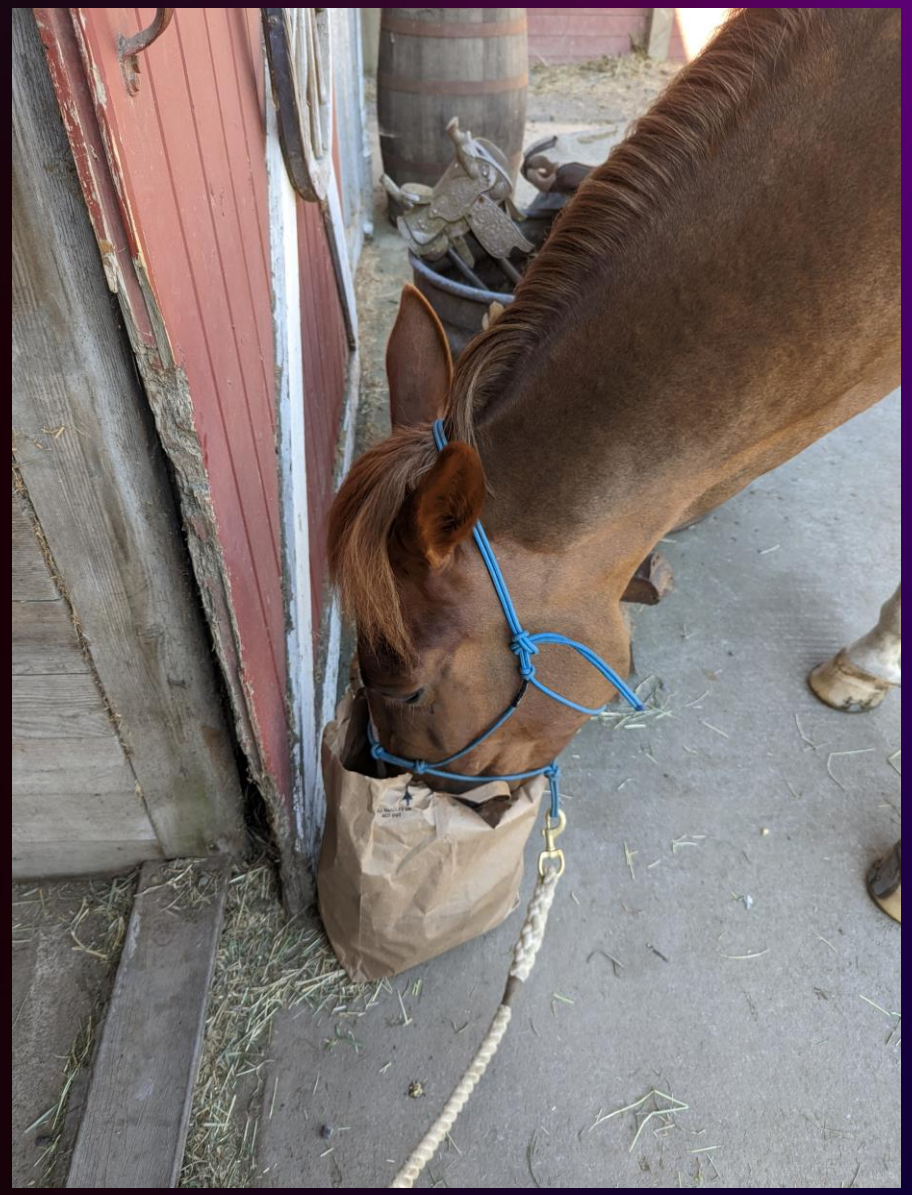
Demo!

Goal

Verify role to access Pickles' Barn

Demo steps

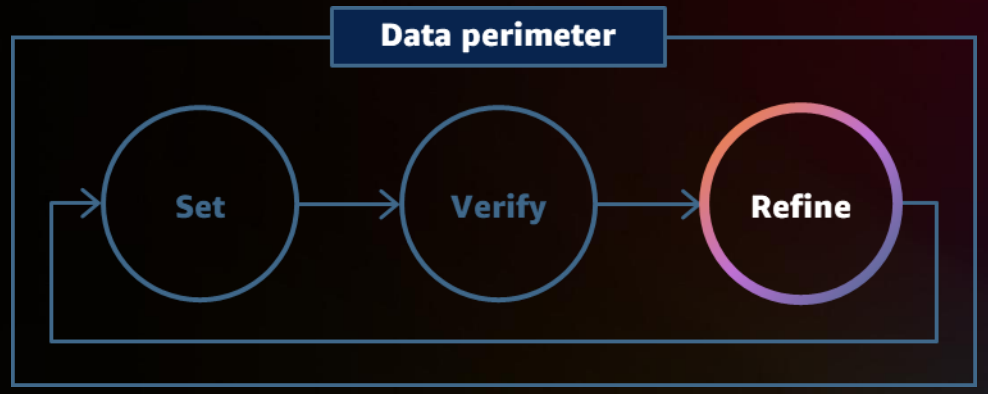
1. Select Pickles' role in the IAM console
2. Inspect trust policy to ensure it is functional and secure
3. Preview findings; add an account
4. Verify findings in IAM Access Analyzer



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Refining permissions



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Identify and safely remove unused permissions

- Role last used
- Access key last used
- Password last used
- Service last accessed
- Action last accessed



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Refine: Last accessed in action

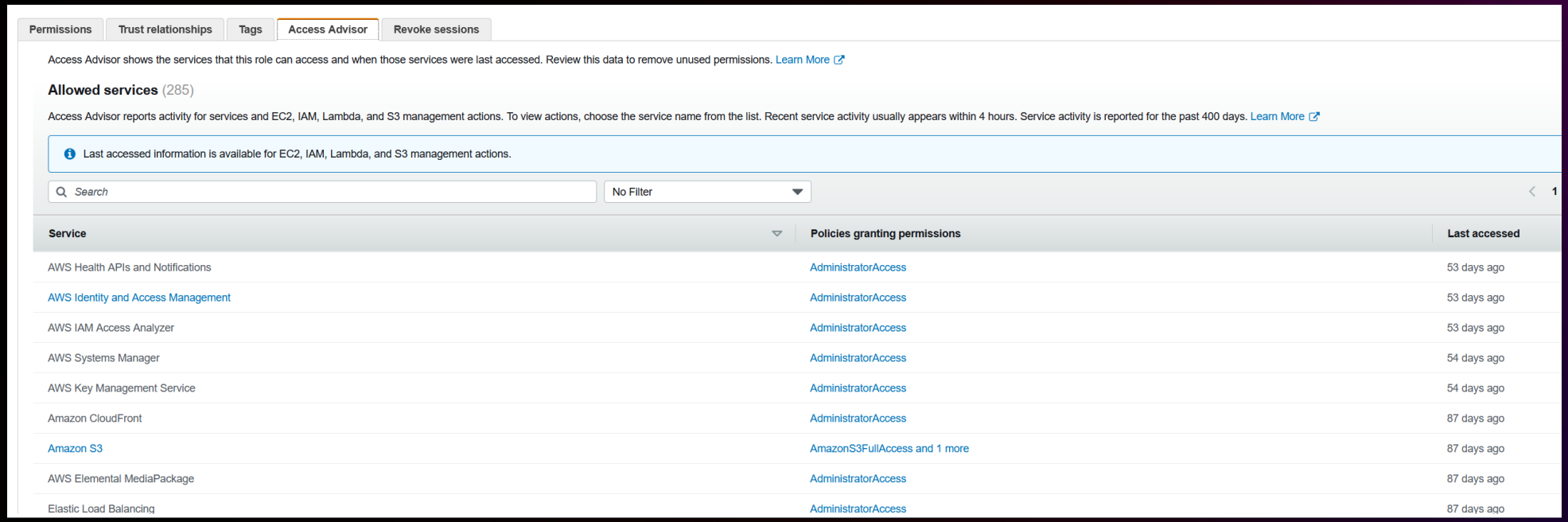
Demo!

Goal

Identify permissions the role did not use in the last 90 days and tighten permissions

Demo steps

1. Navigate to my application role
2. Analyze last accessed information
3. Update policy to remove unused access



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What to take back to your teams

- Use the right permission tool for the right job
- Harness the power of policies, especially conditions
- Restrict public and cross-account access
- Restrict PassRole permissions
- Enable IAM Access Analyzer
- Validate your policies
- Use policy generation with IAM Access Analyzer

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Brigid's favorite permission resources

Service Authorization Reference

Actions, resources, and condition keys for AWS services

PDF

Each AWS service can define actions, resources, and condition context keys for use in IAM policies. This topic describes ho
service are documented.

Each topic consists of tables that provide the list of available actions, resources, and condition keys.

Become an IAM Policy Master in 60 Minutes or Less

AWS re:Invent 2018: [REPEAT 1] Become an IAM Policy ...



Are you interested in becoming a IAM **policy master** and learning about powerful techniques for controlling access to AWS...

YouTube · Amazon Web Services · Nov 28, 2018

How to use trust policies with IAM roles

AWS Security Blog

How to use trust policies with IAM roles

by Jonathan Jenkyn and Liam Wadman | on 03 NOV 2022 | in Advanced (300), AWS Identity And Access Management (IAM), Security, Identity, & Compliance | Permalink | Comments | Share

A least privilege journey: AWS IAM policies and Access Analyzer

A least privilege journey: AWS IAM policies and Access Analyzer



Are you looking for tips and tools for applying least privilege permissions for your users and workloads? Love demonstration...

YouTube · AWS Events · Dec 16, 2021

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Brigid Johnson
@bjohnso5y



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.