Mimmit koodaa

Identiteetin hallinta

Jose Juhala

Identity, access, and resource management

Who

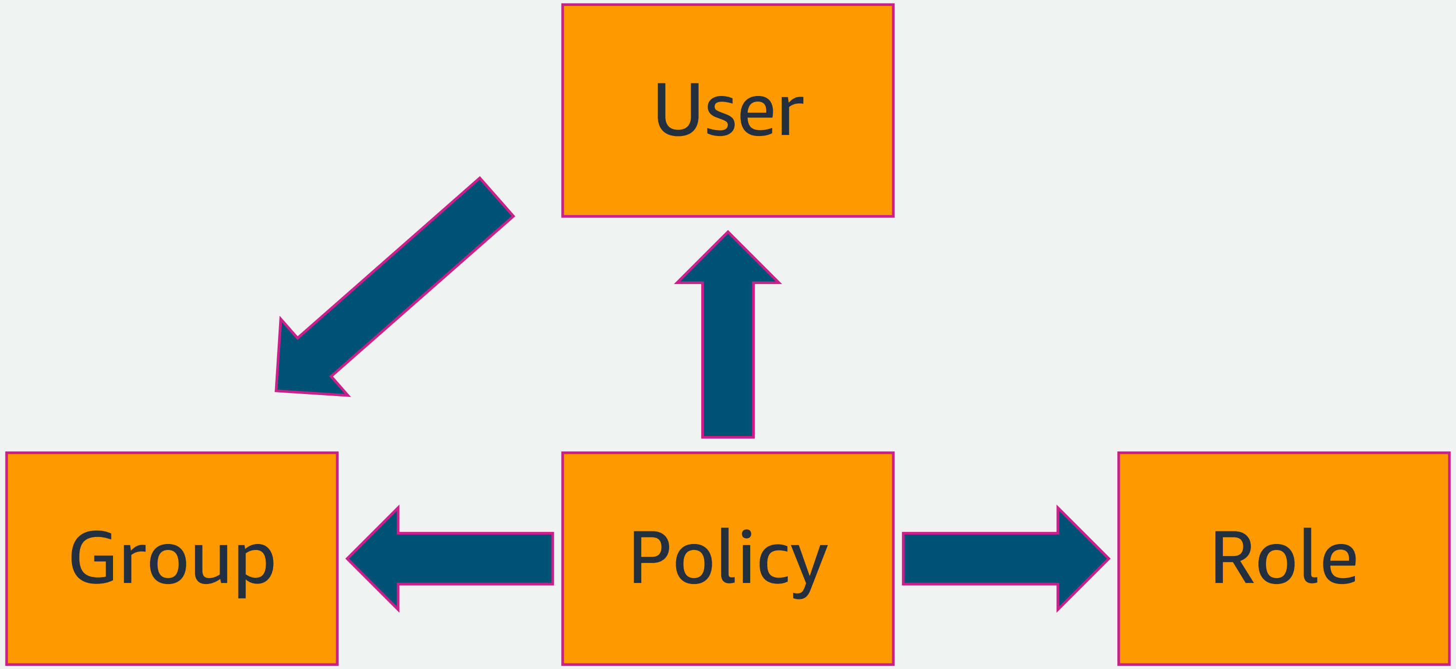
Developers and applications

Can access

Permissions

What

Resources



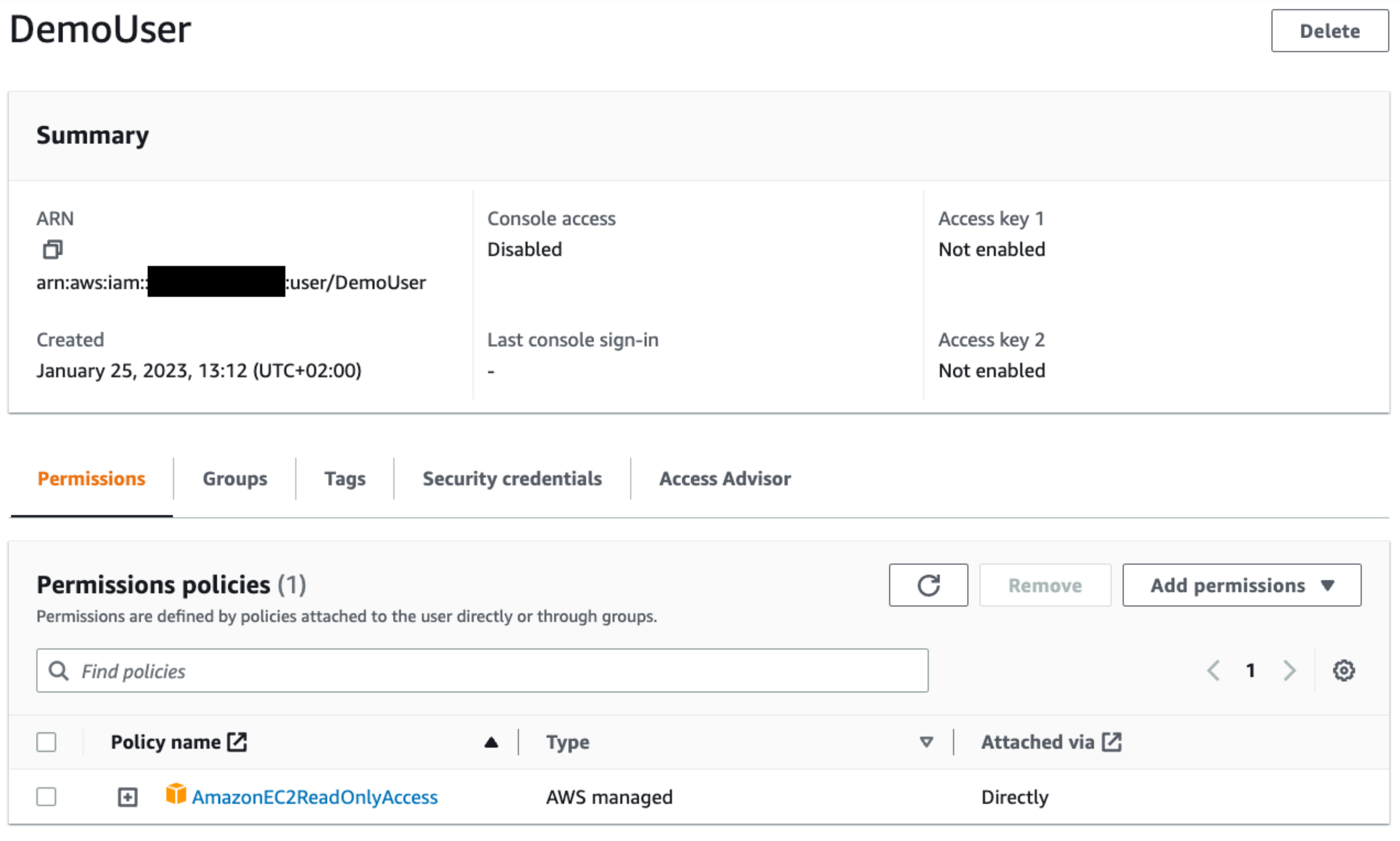
User
Group
Policy
Role

Methods for usage

- **IAM Users**
  - Human users
  - System user (Applications etc)
- **IAM Roles**
  - AWS Services
  - EC2 instances
- **IAM Groups**
  - Grouping of IAM Users for easier management

IAM user authentication

- **Password**
  - IAM User access to AWS Console
  - Controlled by password policies
  - Multi-Factor Authentication
- **Access Keys**
  - Programmatic access to AWS CLI and AWS APIs
  - Consists of an Access Key ID and a Secret Access Key
  - Example
    - Access Key ID: AKIA3R7HGUSSI4BOW
    - Secret Access Key: MxQ4QSzT0NsnEO5VNCo



DemoUser

Summary

ARN
arn:aws:iam::[redacted]:user/DemoUser

Created
January 25, 2023, 13:12 (UTC+02:00)

Console access
Disabled

Last console sign-in
-

Access key 1
Not enabled

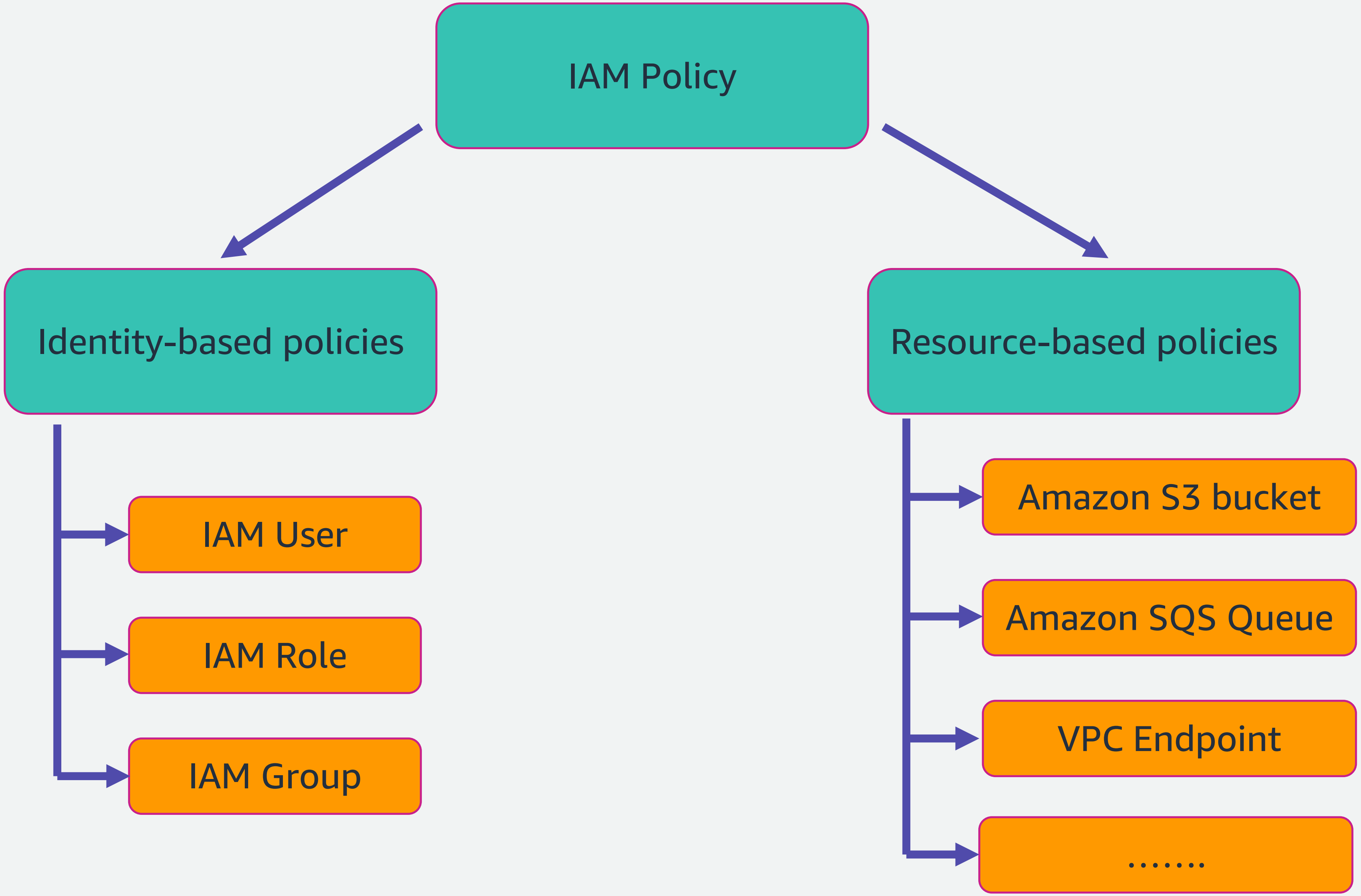
Access key 2
Not enabled

Permissions policies (1)

Permissions are defined by policies attached to the user directly or through groups.

□	Policy name	Type	Attached via
□	AmazonEC2ReadOnlyAccess	AWS managed	Directly

IAM policy



IAM Policy

Identity-based policies

IAM User

IAM Role

IAM Group

Resource-based policies

Amazon S3 bucket

Amazon SQS Queue

VPC Endpoint

....

What are IAM Policies?

Inline policies are policies that you create and manage, and that are embedded directly into a single user, group, or role.

Managed policies are standalone policies that you can manage separately from the IAM users, groups, or roles to which they are attached

- AWS managed policies
- Customer managed polices

Choosing Inline vs Managed Policies

Use *inline policies* when you need to:

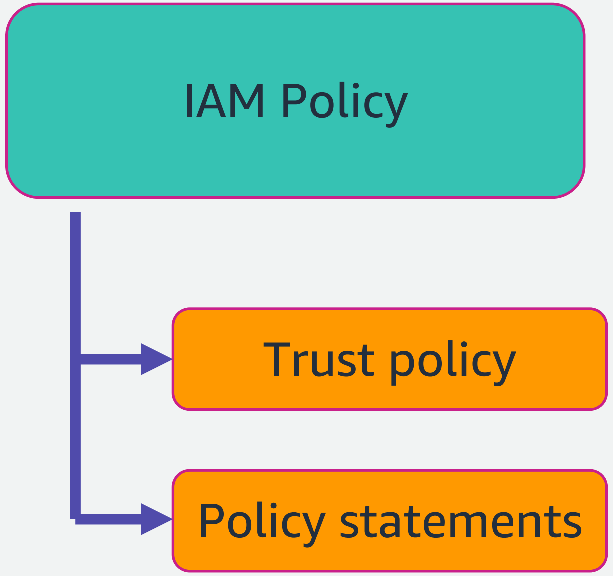
- Enforce a strict one-to-one relationship between policy and principal
- Avoid the wrong policy being attached to a principal
- Ensure the policy is deleted when deleting the principal

Controlling policies

- Permission boundaries
- Organisation SCP
- Access Control Lists
- Session policies

IAM policy structure

Policy structure



IAM Policy

Trust policy

Policy statements

IAM Policy principals

Possible principals

- AWS account and root user
- IAM roles
- Role sessions
- IAM users
- Federated user sessions
- AWS services
- All above

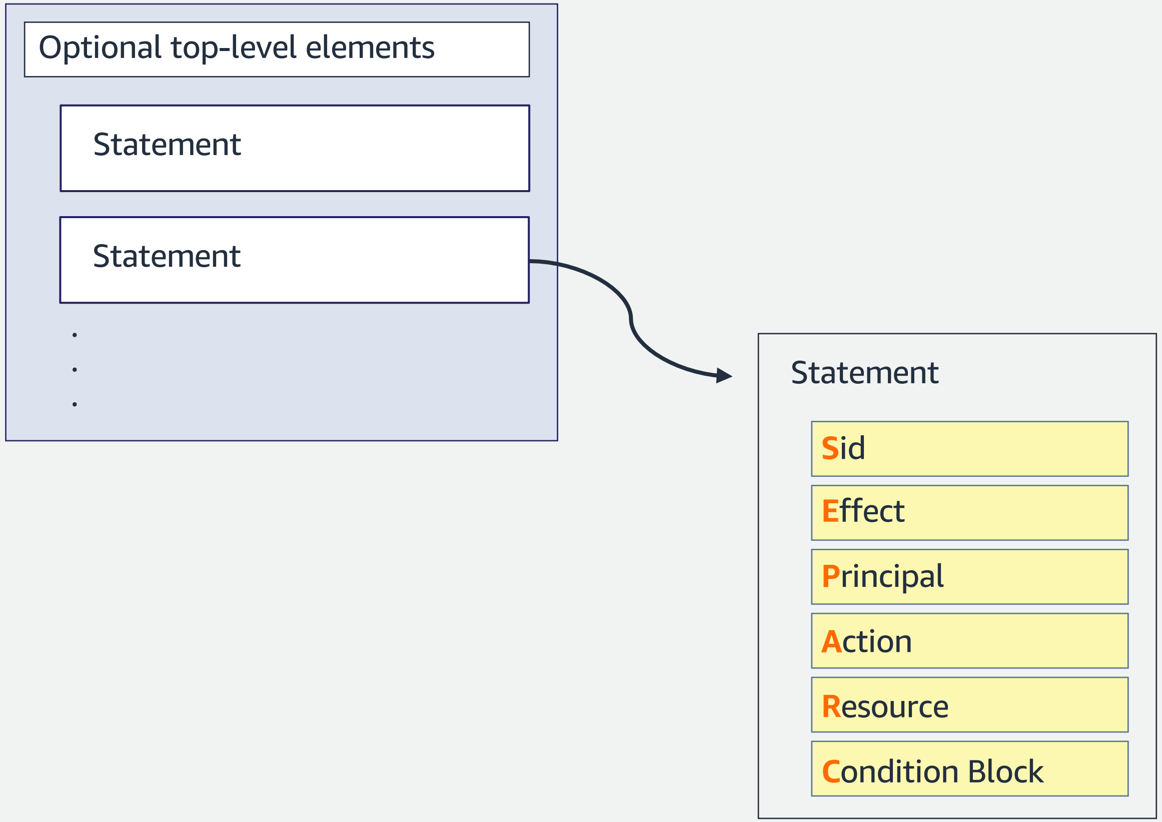
Principal with IAM user

```
 "Principal": {
    "AWS": [
       "arn:aws:iam::AWS-account-ID:user/user-name-1”,
        "arn:aws:iam::AWS-account-ID:user/user-name-2”
     ]
 }
```

Principal with service

```
"Principal": {
   "Service": [
      "ecs.amazonaws.com",
      "elasticloadbalancing.amazonaws.com”
   ]
}
```

AWS IAM policy structure



Optional top-level elements

Statement

Statement

.
.
.

Statement

Sid

Effect

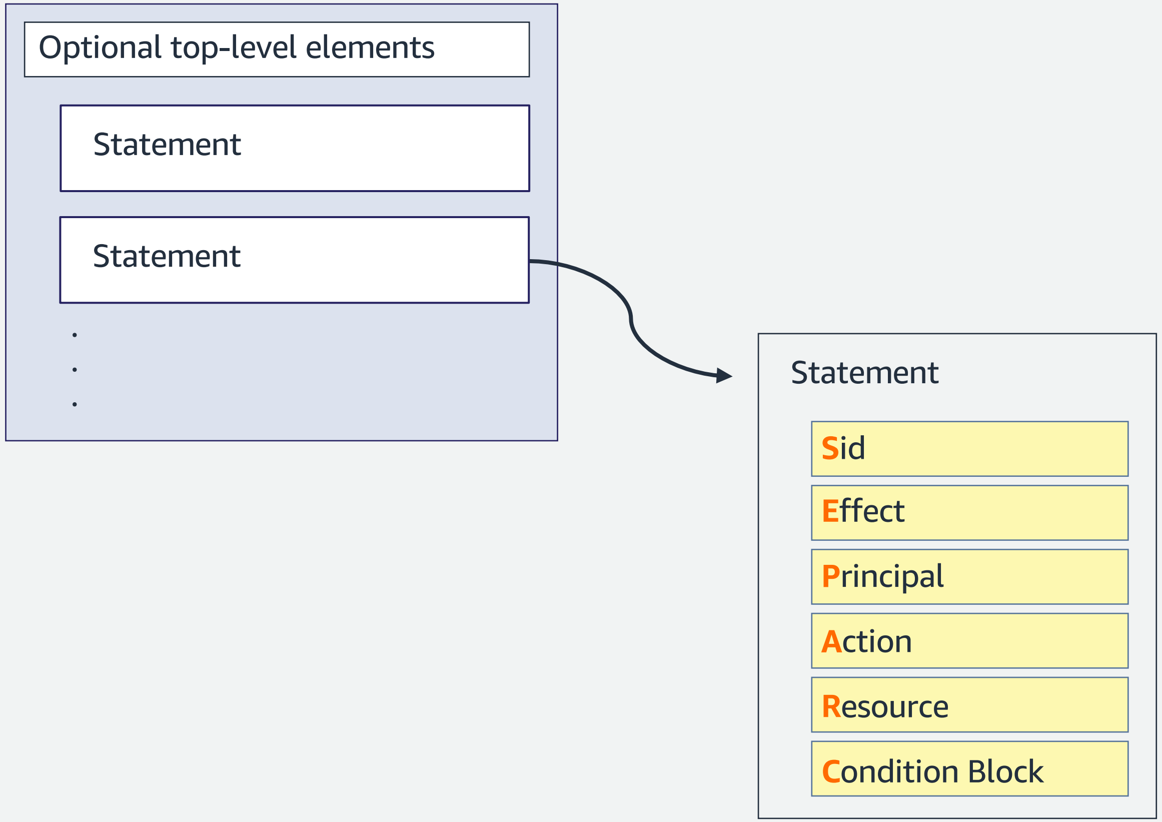
Principal

Action

Resource

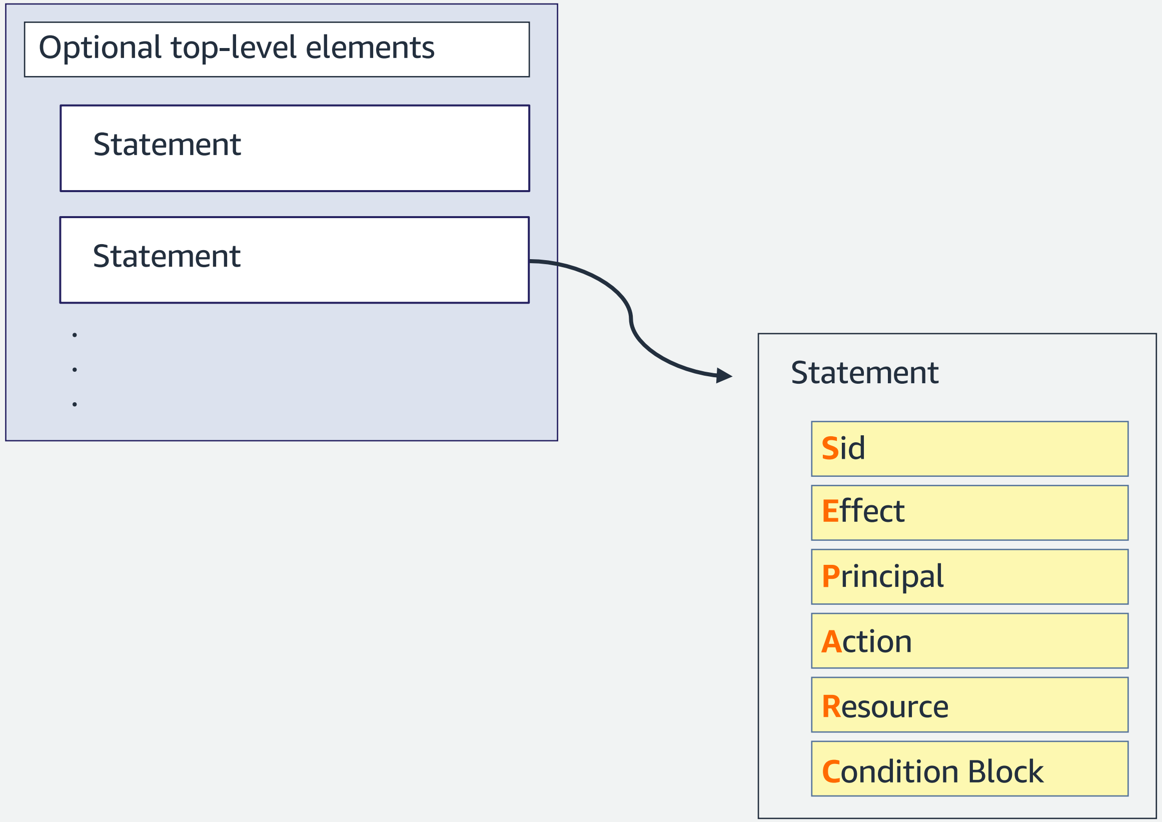
Condition Block

AWS IAM policy structure



Sid (Optional) – Include an optional statement ID to differentiate between your statements

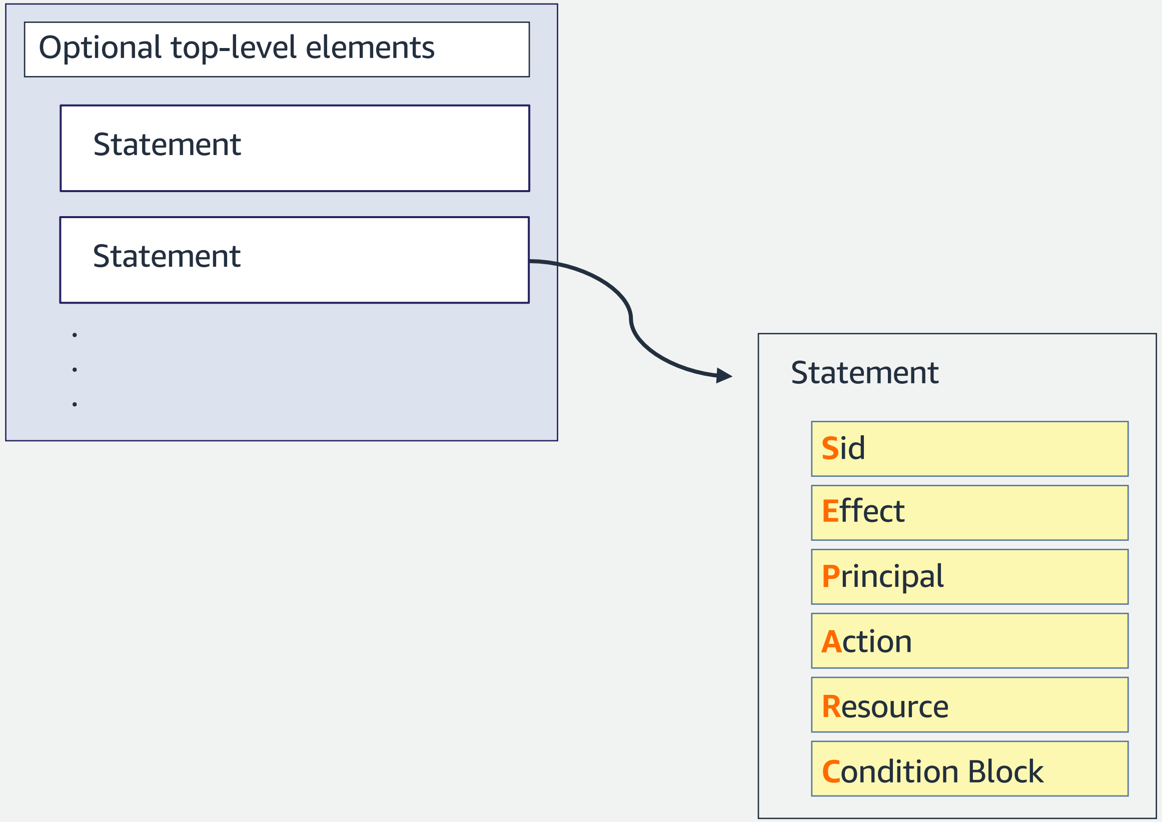
AWS IAM policy structure



Sid (Optional) – Include an optional statement ID to differentiate between your statements

Effect – Use Allow or Deny to indicate whether the policy allows or denies access.

AWS IAM policy structure

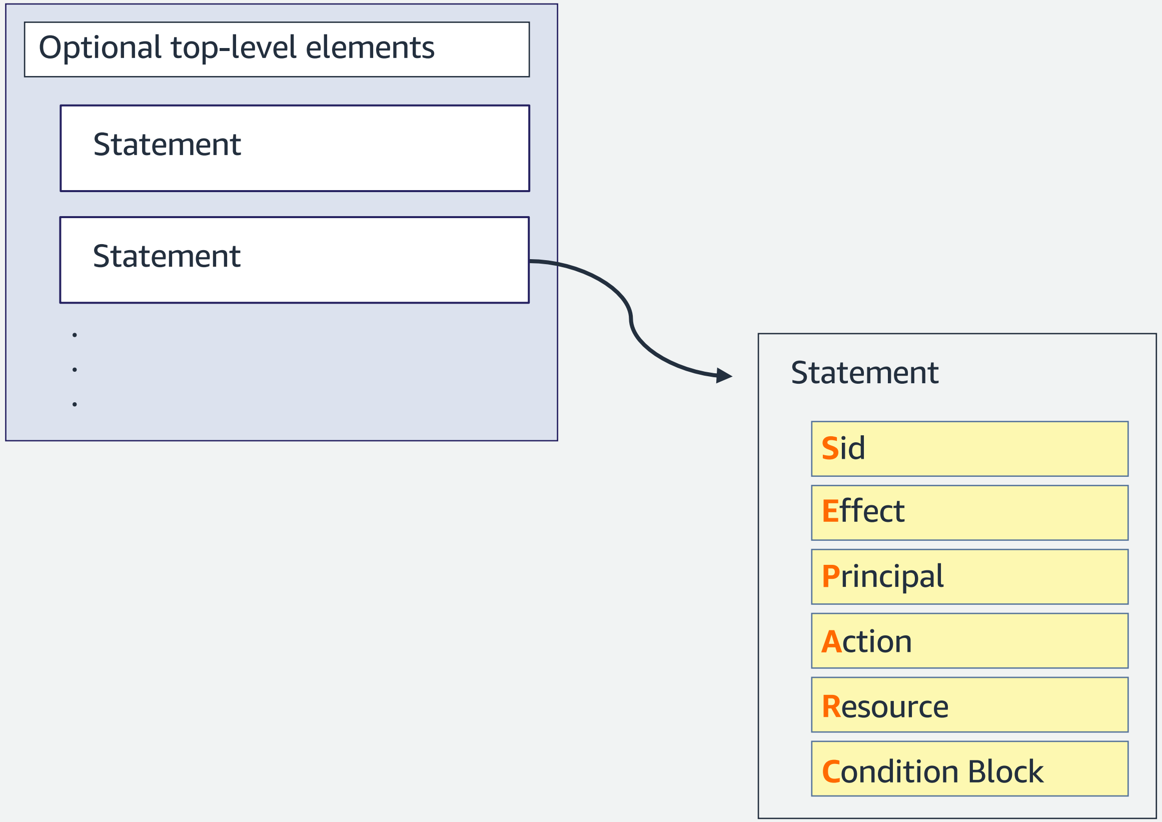


Sid (Optional) – Include an optional statement ID to differentiate between your statements

Effect – Use Allow or Deny to indicate whether the policy allows or denies access.

Principal (Required in only some circumstances) – If you create a resource-based policy, you must indicate the account, user, role, or federated user to which you would like to allow or deny access

AWS IAM policy structure



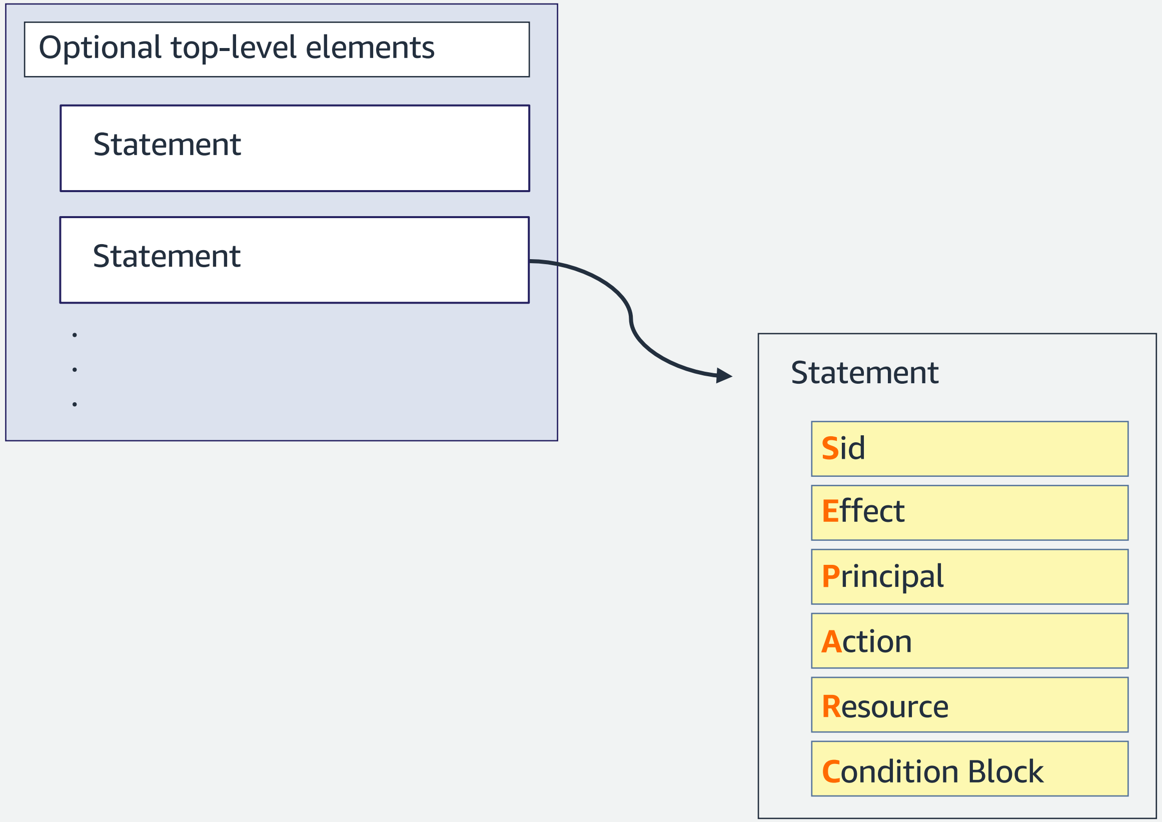
Sid (Optional) – Include an optional statement ID to differentiate between your statements

Effect – Use Allow or Deny to indicate whether the policy allows or denies access.

Principal (Required in only some circumstances) – If you create a resource-based policy, you must indicate the account, user, role, or federated user to which you would like to allow or deny access

Action – Include a list of actions that the policy allows or denies.

AWS IAM policy structure



Sid (Optional) – Include an optional statement ID to differentiate between your statements

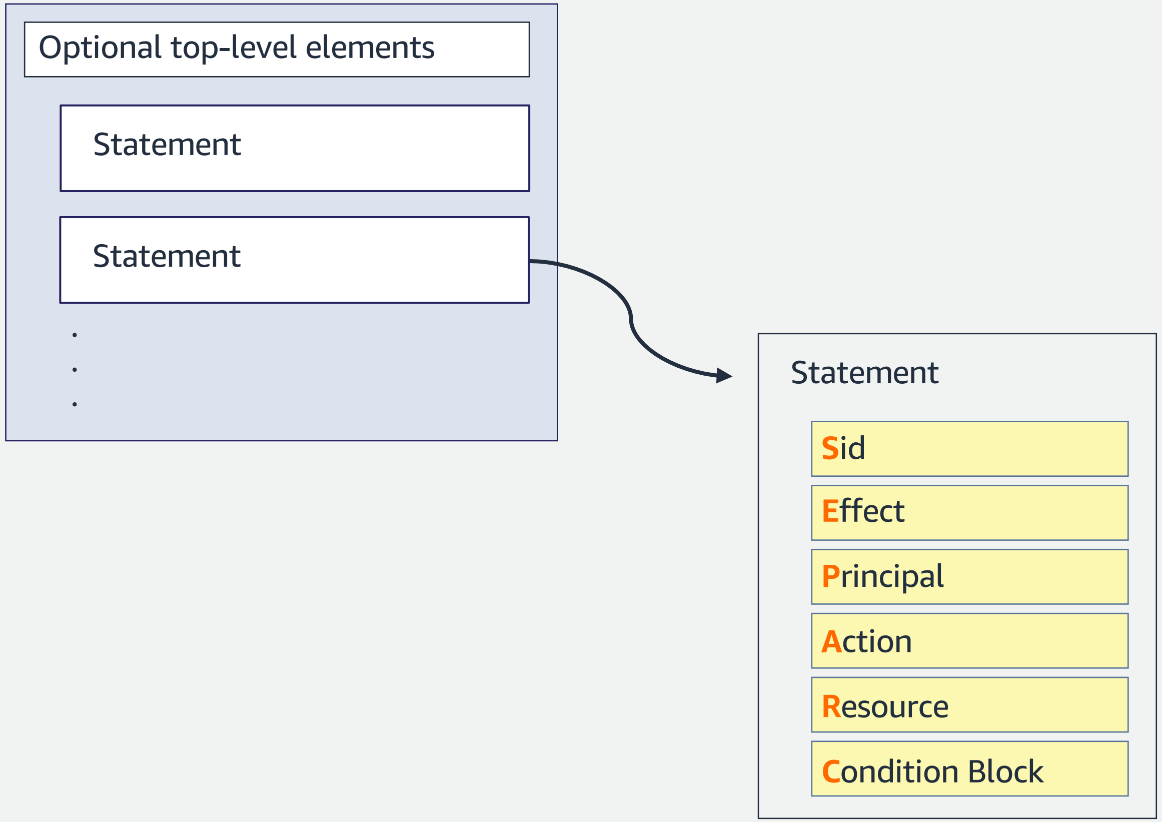
Effect – Use Allow or Deny to indicate whether the policy allows or denies access.

Principal (Required in only some circumstances) – If you create a resource-based policy, you must indicate the account, user, role, or federated user to which you would like to allow or deny access

Action – Include a list of actions that the policy allows or denies.

Resource (Required in only some circumstances) – If you create an IAM permissions policy, you must specify a list of resources to which the actions apply.

AWS IAM policy structure



Sid (Optional) – Include an optional statement ID to differentiate between your statements

Effect – Use Allow or Deny to indicate whether the policy allows or denies access.

Principal (Required in only some circumstances) – If you create a resource-based policy, you must indicate the account, user, role, or federated user to which you would like to allow or deny access

Action – Include a list of actions that the policy allows or denies.

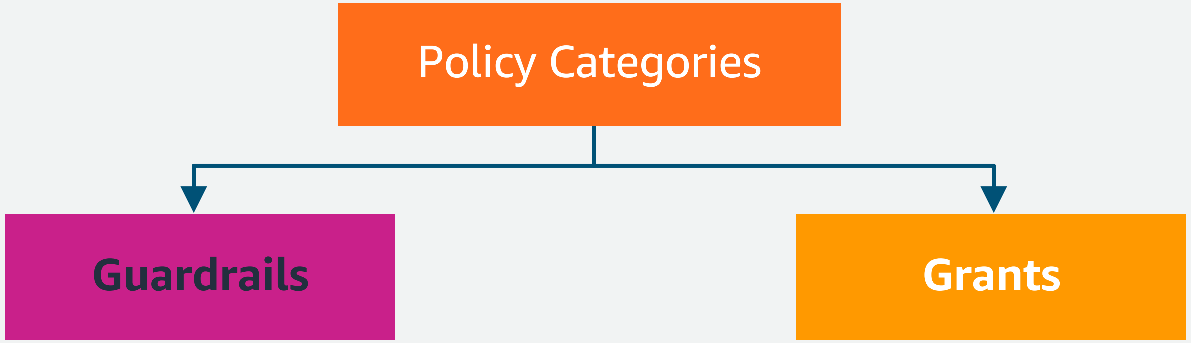
Resource (Required in only some circumstances) – If you create an IAM permissions policy, you must specify a list of resources to which the actions apply.

Condition (Optional) – Specify the circumstances under which the policy grants permission.

AWS IAM policy structure example

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewAccountPasswordRequirements",
            "Effect": "Allow",
            "Action": "iam:GetAccountPasswordPolicy",
            "Resource": "*"
        },
        {
            "Sid": "ChangeOwnPassword",
            "Effect": "Allow",
            "Action": [
                "iam:GetUser",
                "iam:ChangePassword"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        }
    ]
}
```

AWS Access Management



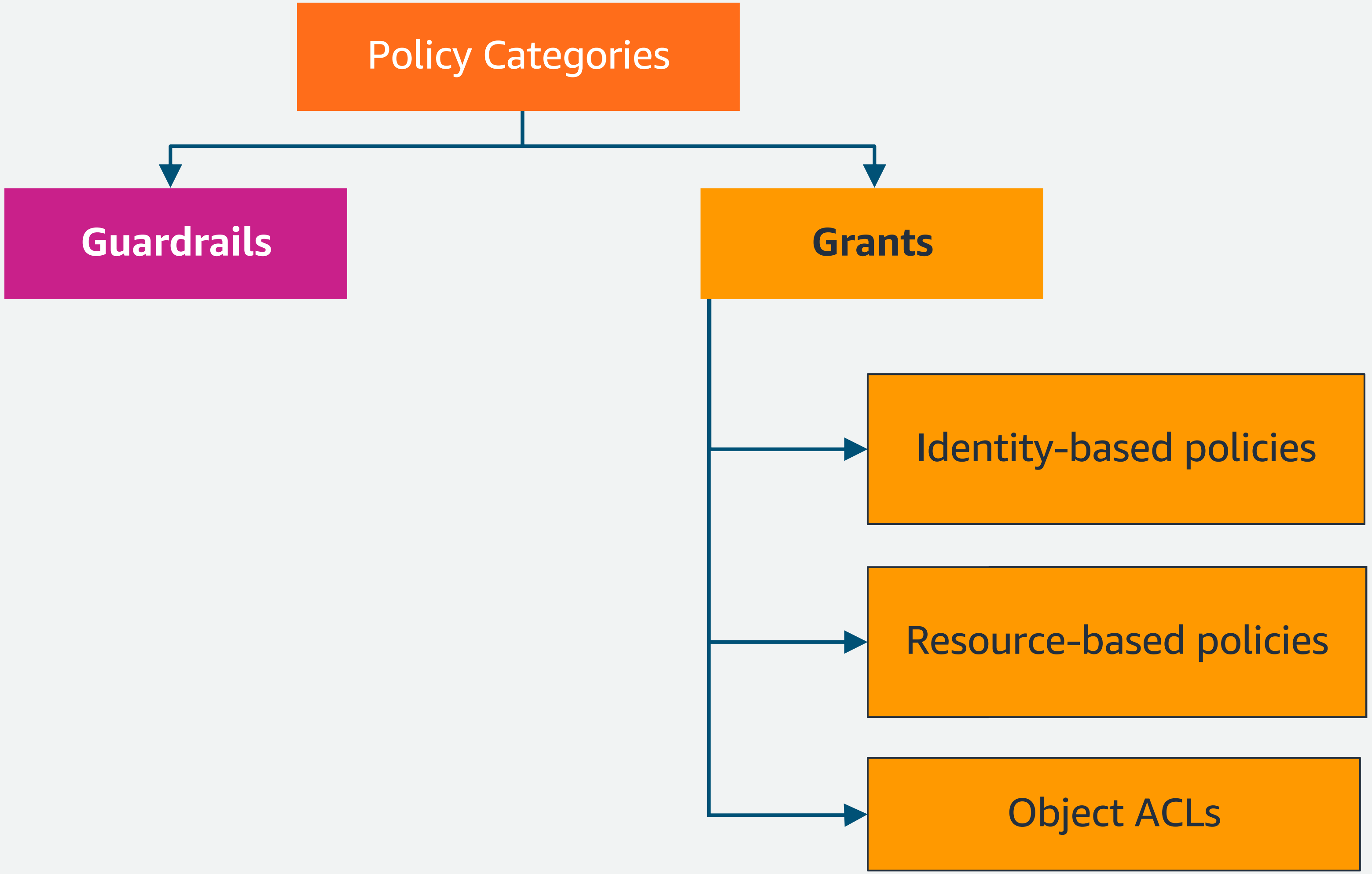
Policies that set the maximum permission

Policies that give permission

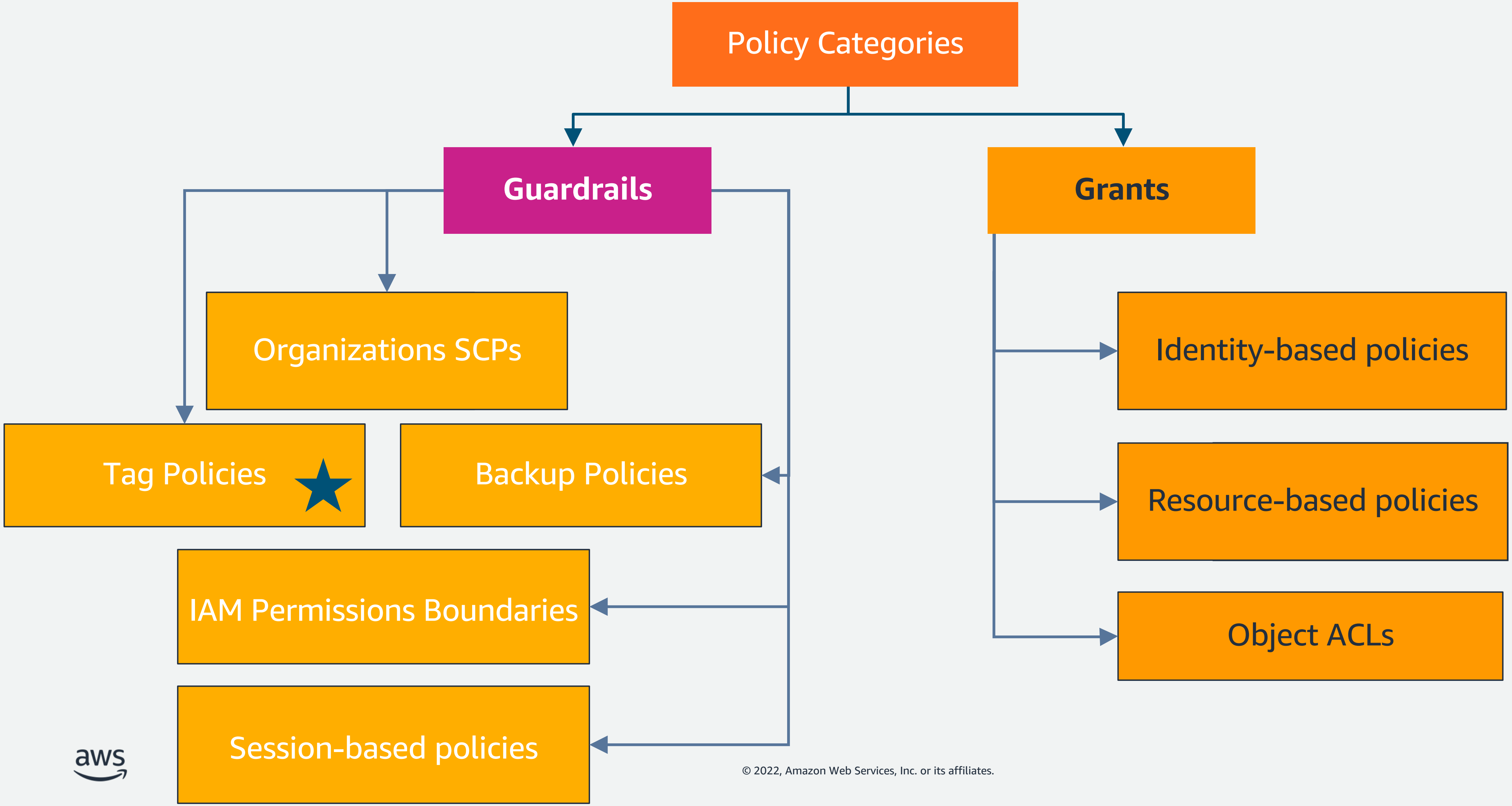
https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html

https://aws-samples.github.io/aws-iam-permissions-guardrails/guardrails/scp-guardrails.html

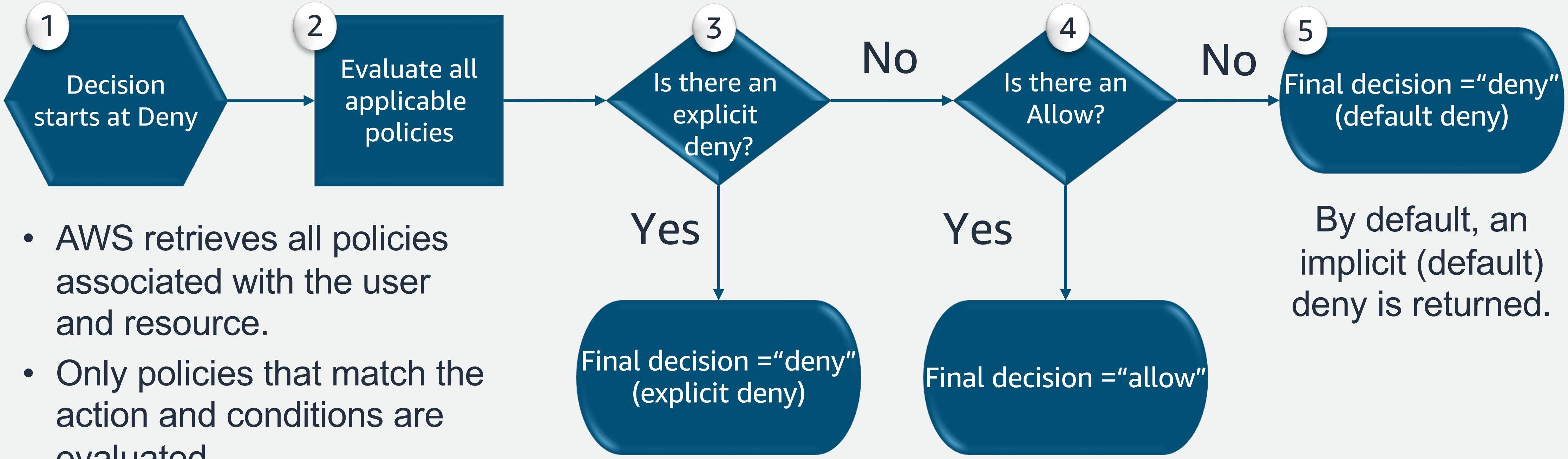
AWS Access Management



AWS Access Management



IAM Policy Evaluation Logic


1
Decision starts at Deny
2
Evaluate all applicable policies
3
Is there an explicit deny?
No
4
Is there an Allow?
No
5
Final decision =“deny” (default deny)
Yes
Final decision =“deny” (explicit deny)
Yes
Final decision =“allow”

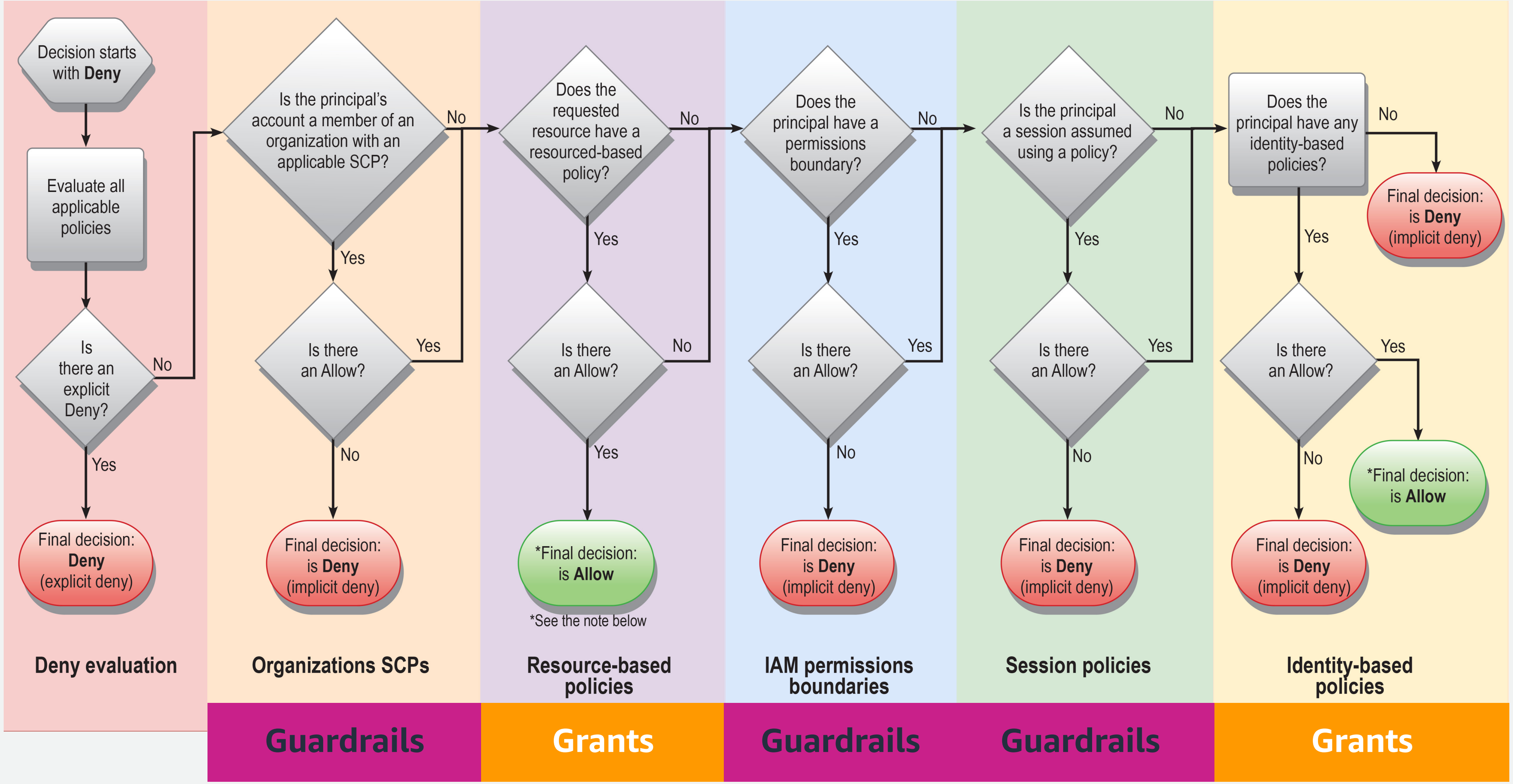
- AWS retrieves all policies associated with the user and resource.
- Only policies that match the action and conditions are evaluated.

If a policy statement has a deny, it trumps all other policy statements.

Access is granted if there is an explicit allow and no deny.

By default, an implicit (default) deny is returned.

AWS IAM policy evaluation logic. End-to-end


Decision starts with **Deny**
Evaluate all applicable policies
Is there an explicit Deny?
No
Yes
Final decision: **Deny** (explicit deny)
Is the principal's account a member of an organization with an applicable SCP?
No
Yes
Is there an Allow?
Yes
No
Final decision: is **Deny** (implicit deny)
Does the requested resource have a resourced-based policy?
No
Yes
Is there an Allow?
No
Yes
*Final decision: is **Allow**
*See the note below
Does the principal have a permissions boundary?
No
Yes
Is there an Allow?
Yes
No
Final decision: is **Deny** (implicit deny)
Is the principal a session assumed using a policy?
No
Yes
Is there an Allow?
Yes
No
Final decision: is **Deny** (implicit deny)
Does the principal have any identity-based policies?
No
Final decision: is **Deny** (implicit deny)
Yes
Is there an Allow?
Yes
*Final decision: is **Allow**
No
Final decision: is **Deny** (implicit deny)
Deny evaluation
Organizations SCPs
Resource-based policies
IAM permissions boundaries
Session policies
Identity-based policies
Guardrails
Grants
Guardrails
Guardrails
Grants

https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_evaluation-logic.html

Who can access what?

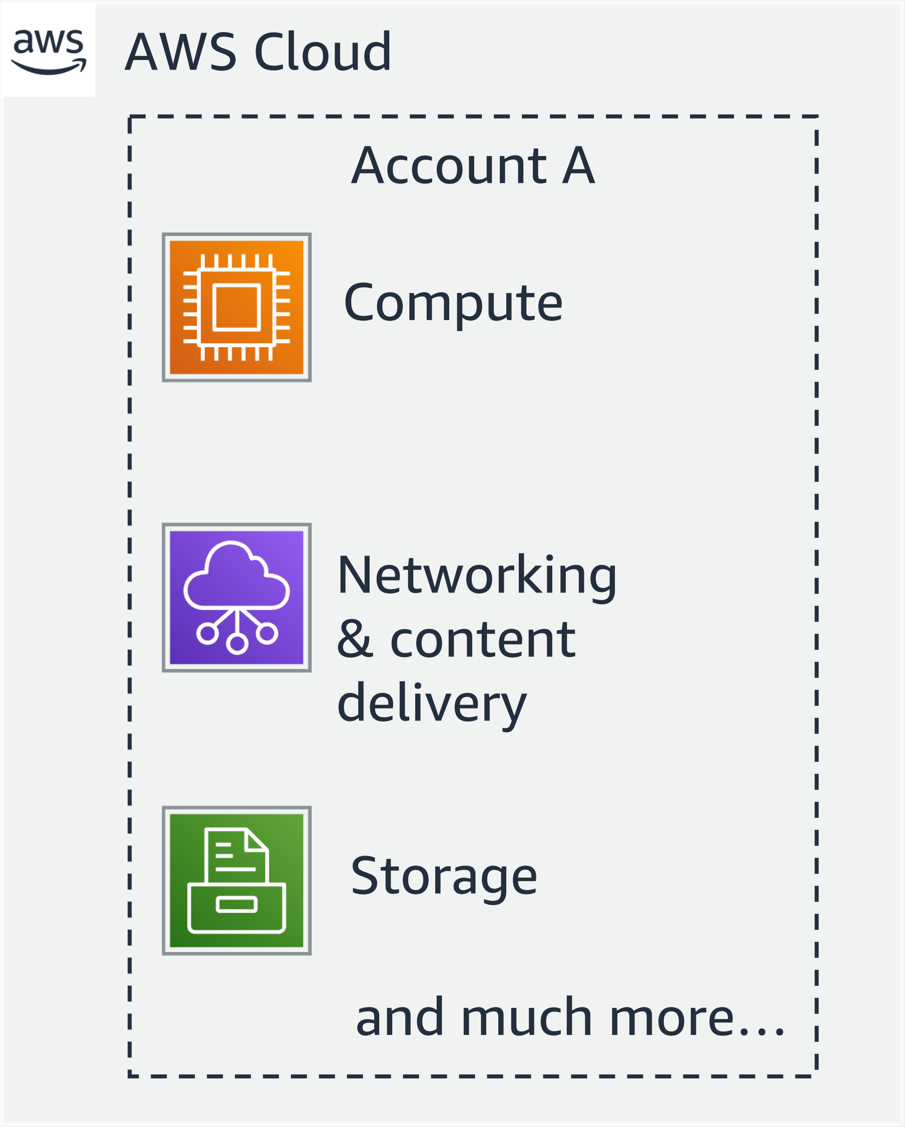
First understand an AWS account

Each AWS account is

- a resource container for AWS Cloud services
- an explicit security boundary
- a container for cost tracking and billing
- a mechanism to enforce limits and thresholds
    - e.g. Service Quotas and API thresholds

+ users, groups, roles, policies

Over time, customers will add more accounts to support more applications and services



AWS Organizations

- Central governance and management across AWS accounts for a comprehensive multi-account AWS environment

Manage and define your organisation and accounts

Control access and permission

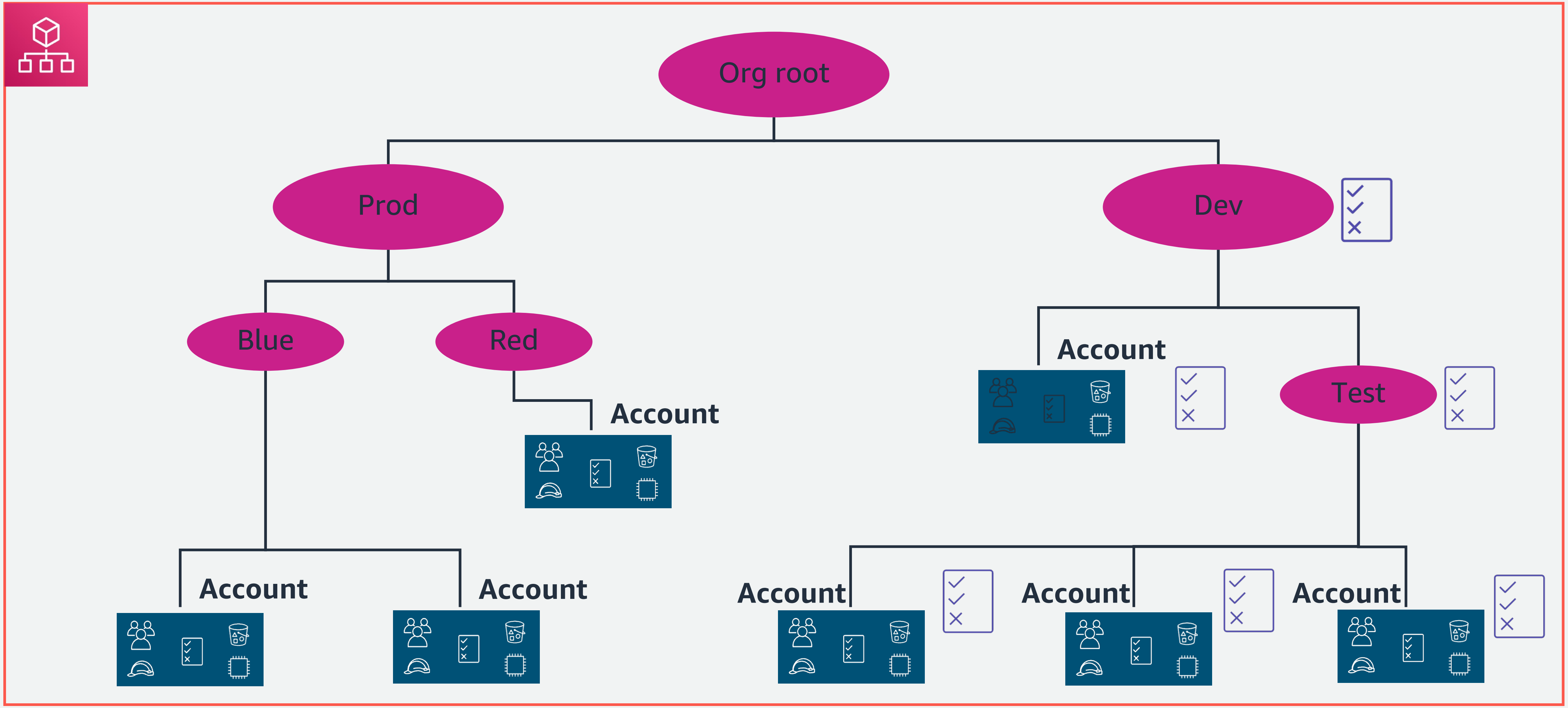
Audit, monitor, and secure your environment for compliance

Share resources across accounts

Centrally manage costs and billing

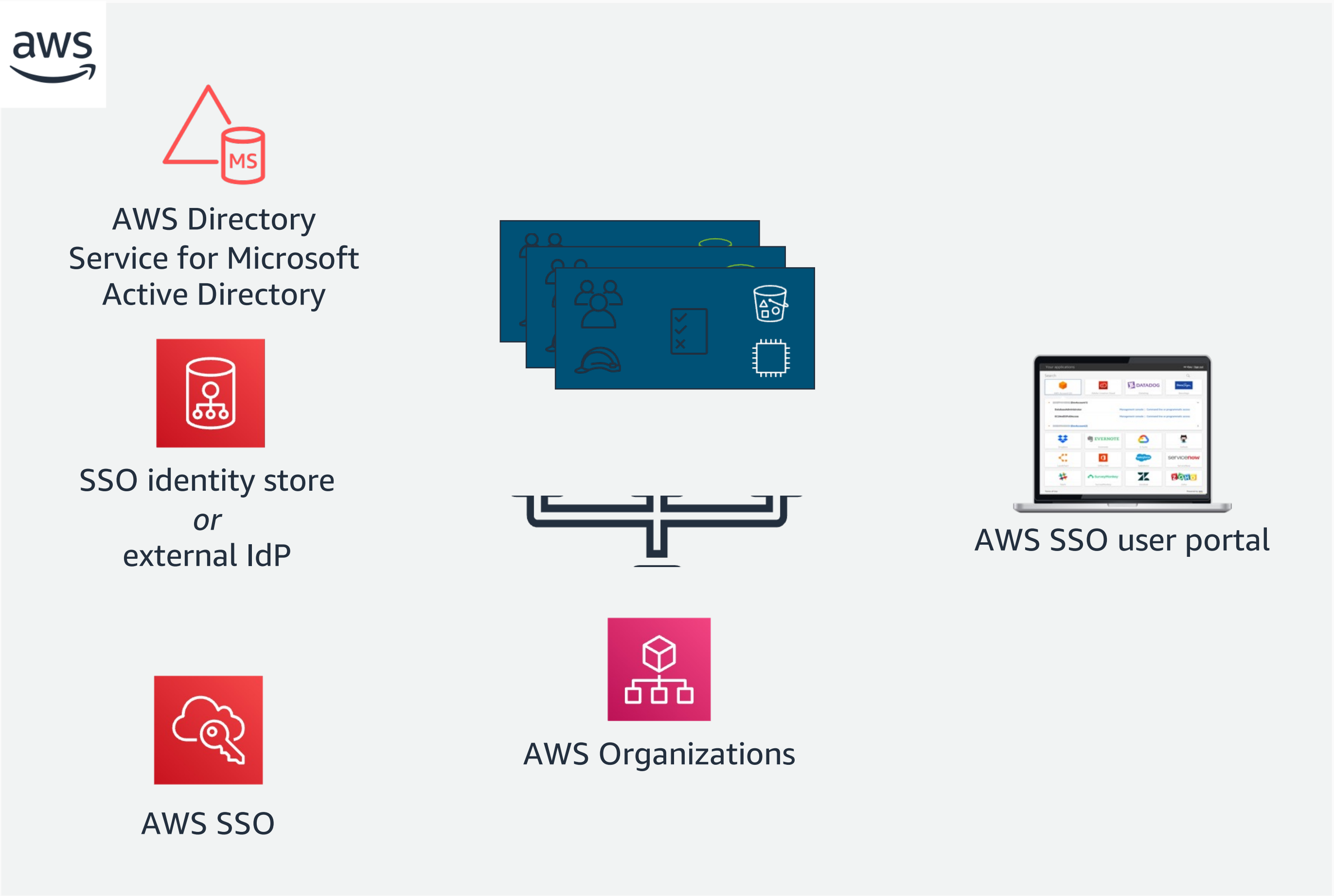
https://aws.amazon.com/organizations/getting-started/best-practices/

Ensure AWS accounts are governed



How to access AWS

Start with AWS Single Sign-On (SSO)



AWS Directory Service for Microsoft Active Directory

SSO identity store *or* external IdP

AWS SSO

AWS Organizations

AWS SSO user portal

This enables you to

- Manage users and groups where they want; connect to AWS once
- Centrally assign and manage access to AWS accounts; AWS SSO–integrated and cloud-based business applications
- Provide SSO user portal to assigned AWS accounts; AWS and business applications
- Increase developer productivity with AWS Command Line Interface (AWS CLI) v2

One AWS access control model
You choose your identity source

https://aws.amazon.com/blogs/aws/new-attributes-based-access-control-with-aws-single-sign-on/ ★

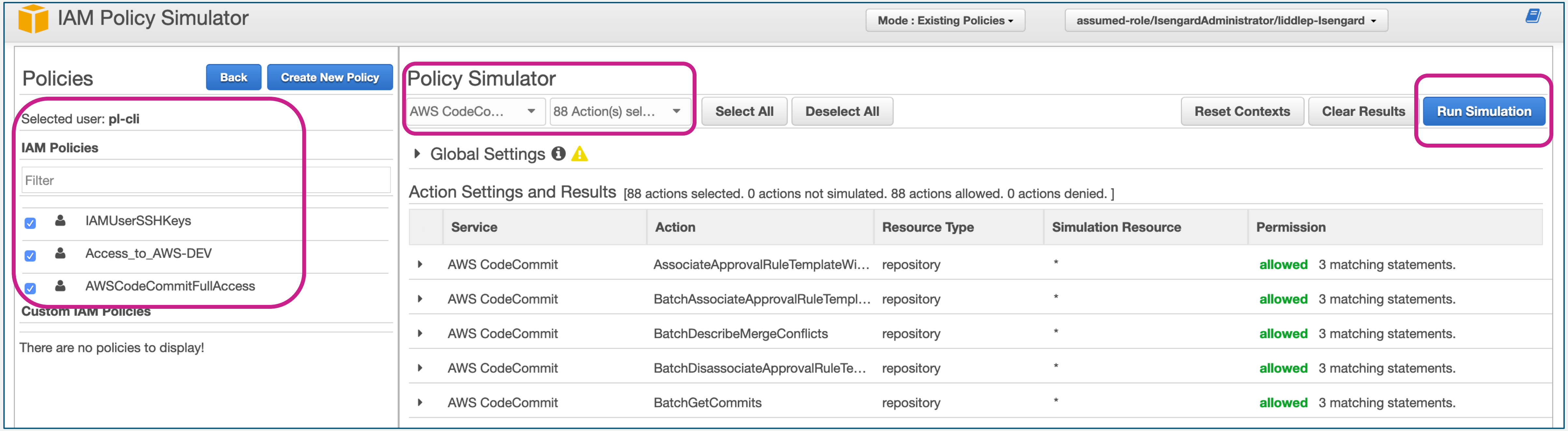
What is AWS Security Token Service (STS)?

The AWS Security Token Service (STS) is a web service that enables you to request temporary, limited-privilege credentials for AWS Identity and Access Management (IAM) users or for users that you authenticate (federated users)

AWS IAM tools

AWS IAM Tools – Policy simulator

1 2 3



Policies

Selected user: **pl-cli**

IAM Policies

- ☑ IAMUserSSHKeys
- ☑ Access_to_AWS-DEV
- ☑ AWSCodeCommitFullAccess

Custom IAM Policies

There are no policies to display!

Policy Simulator

AWS CodeCo... ▾ 88 Action(s) sel... ▾

▸ Global Settings ℹ ⚠

Action Settings and Results [88 actions selected. 0 actions not simulated. 88 actions allowed. 0 actions denied.]

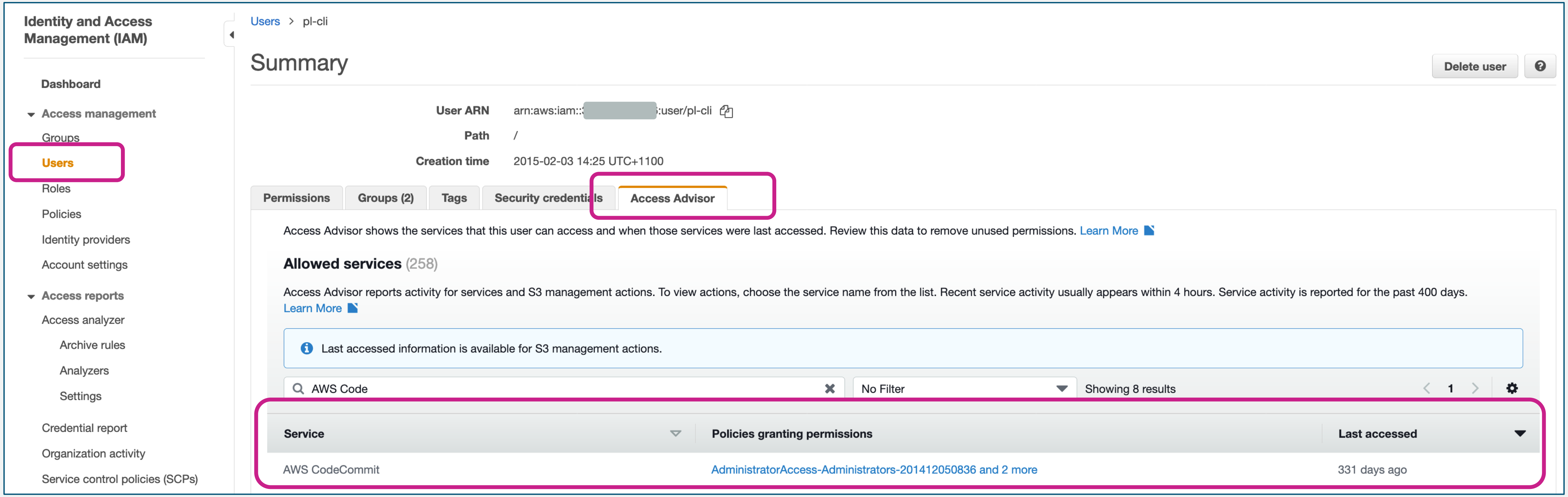
	Service	Action	Resource Type	Simulation Resource	Permission
▸	AWS CodeCommit	AssociateApprovalRuleTemplateWi...	repository	*	allowed 3 matching statements.
▸	AWS CodeCommit	BatchAssociateApprovalRuleTempl...	repository	*	allowed 3 matching statements.
▸	AWS CodeCommit	BatchDescribeMergeConflicts	repository	*	allowed 3 matching statements.
▸	AWS CodeCommit	BatchDisassociateApprovalRuleTe...	repository	*	allowed 3 matching statements.
▸	AWS CodeCommit	BatchGetCommits	repository	*	allowed 3 matching statements.

https://policysim.aws.amazon.com/

AWS IAM Tools – Access Advisor

1

2



Identity and Access Management (IAM)

- Dashboard
- Access management
  - Groups
  - Users
  - Roles
  - Policies
  - Identity providers
  - Account settings
- Access reports
  - Access analyzer
    - Archive rules
    - Analyzers
    - Settings
  - Credential report
  - Organization activity
  - Service control policies (SCPs)

Users > pl-cli

Summary

Delete user

User ARN arn:aws:iam::[redacted]:user/pl-cli

Path /

Creation time 2015-02-03 14:25 UTC+1100

Permissions | Groups (2) | Tags | 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 (258)

Access Advisor reports activity for services 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 the past 400 days. Learn More

Last accessed information is available for S3 management actions.

AWS Code | No Filter | Showing 8 results | 1

Service	Policies granting permissions	Last accessed
AWS CodeCommit	AdministratorAccess-Administrators-201412050836 and 2 more	331 days ago

3

https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_access-advisor-view-data.html

Summary

IAM Best Practices

- Take an iterative approach
- Automate NOW!
- Lock away your AWS account (root) access keys
- Create individual IAM users
- Use groups to assign permissions to IAM users
- Grant least privilege
- Configure a strong password policy for your users
- Enable MFA for privileged users

IAM Best Practices

- Use roles for applications that run on Amazon EC2 instances
- Delegate by using roles instead of by sharing credentials
- Rotate credentials regularly
- Remove unnecessary credentials
- Use policy conditions for extra security
- Monitor activity in your AWS account

http://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html

Thank you!