SANS

BLUE TEAM

SUMMIT & TRAINING

Join us in Scottsdale, AZ or Live Online for **FREE**

SUMMIT: October 3–4 | TRAINING: October 5–10

SANS
BLUE TEAM
SUMMIT & TRAINING

Keynote - A Deep Dive into AWS IAM Privilege Escalation Attacks Defenders’ Edition 2022

Ashwin Patil, Microsoft Security Research

Roberto Rodriguez, Microsoft Security Research

https://aka.ms/SBTS22-Keynote-Slides

@Cyb3rWard0g

Roberto Rodriguez 🇵🇪

Principal Threat Researcher at the Microsoft Security Research Organization

Founder of the Open Threat Research community! @OTR_Community

I 🖤 open source and dogs!

Empowering others 🌎 https://github.com/OTRF

https://msrc-blog.microsoft.com/2022/09/07/curious-innovative-creative-community-driven-meet-cyb3rward0g-roberto-rodriquez/

@ashwinpatil

Ashwin Patil 🇮🇳

Senior Security Researcher at the Microsoft Security Research Organization

Empowering others 🌎
https://github.com/ashwin-patil/

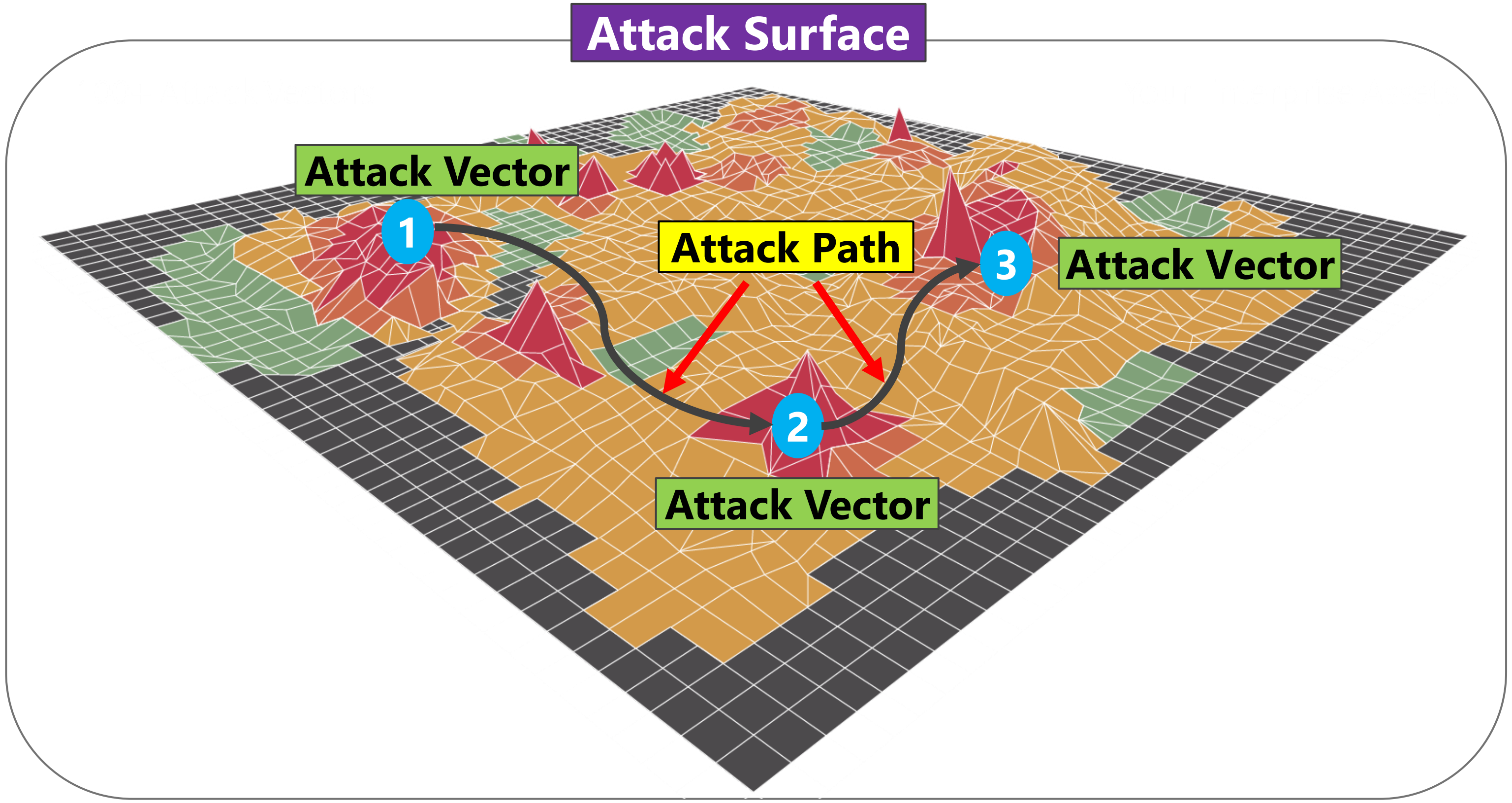
Agenda

- AWS Attack Surface
- AWS IAM 101
- Attack and Defend
- Free Resources

0

A Few Terms

- **Attack Surface**: All the angles of attack of a system, a system element, or an environment.
- **Attack Vector:** The means by which an adversary uses to compromise a system or an environment.
- **Attack Path:** Chain of exploitable attack vectors.



1

AWS Attack Surface

A Few AWS Services

Storage

- Amazon Elastic Block Store (Amazon EBS)
- AWS Snowball
- Amazon Simple Storage Service (Amazon S3)
- AWS Backup

Application Integration

- Amazon MQ
- Amazon AppFlow
- Amazon API Gateway
- Amazon Simple Queue Service (Amazon SQS)

Compute

- Amazon Elastic Compute Cloud (Amazon EC2)
- AWS Lambda
- Amazon Lightsail
- NICE DCV

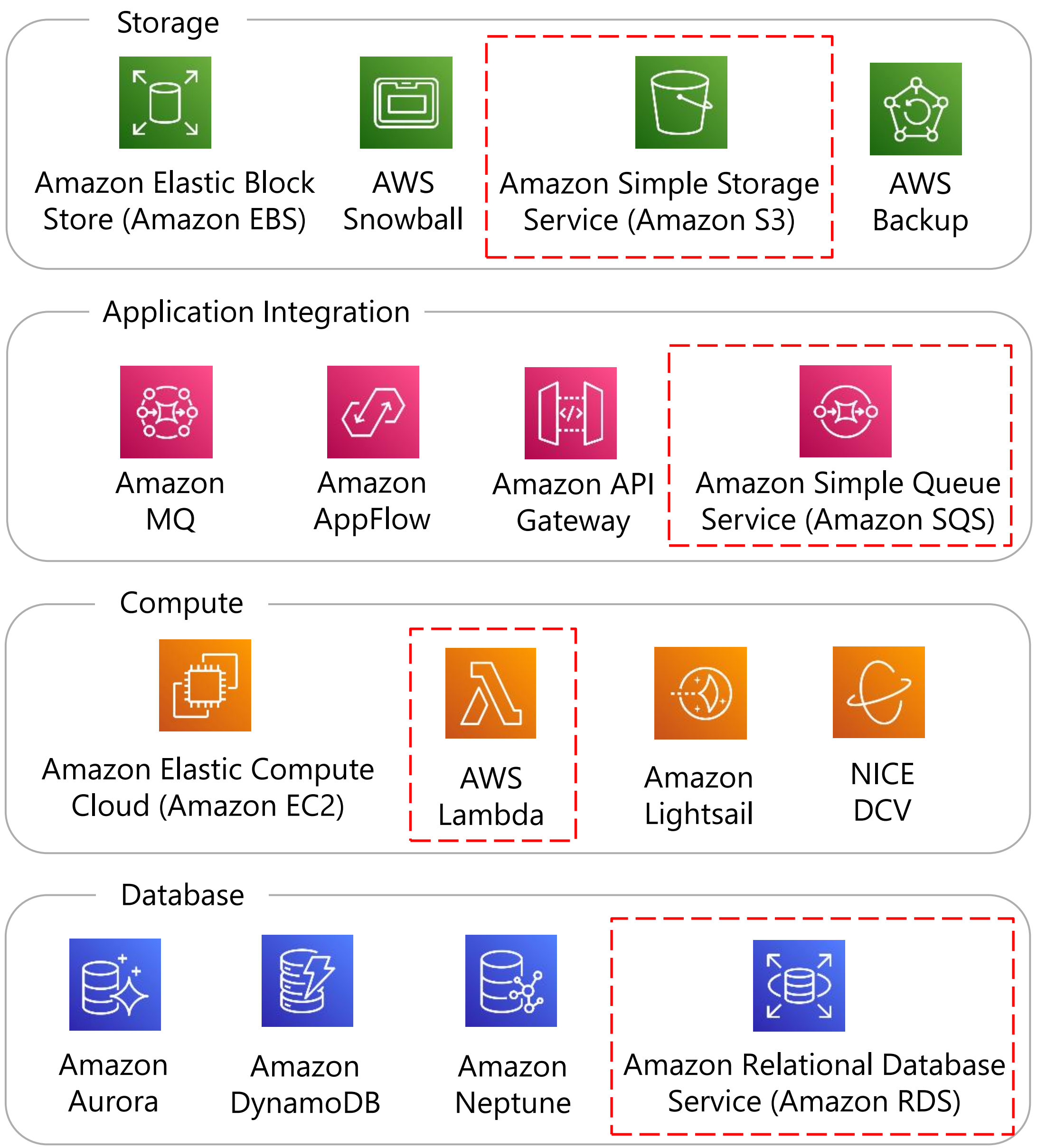
Database

- Amazon Aurora
- Amazon DynamoDB
- Amazon Neptune
- Amazon Relational Database Service (Amazon RDS)

1

AWS Attack Surface

A Few AWS Services



Storage

Amazon Elastic Block Store (Amazon EBS)

AWS Snowball

Amazon Simple Storage Service (Amazon S3)

AWS Backup

Application Integration

Amazon MQ

Amazon AppFlow

Amazon API Gateway

Amazon Simple Queue Service (Amazon SQS)

Compute

Amazon Elastic Compute Cloud (Amazon EC2)

AWS Lambda

Amazon Lightsail

NICE DCV

Database

Amazon Aurora

Amazon DynamoDB

Amazon Neptune

Amazon Relational Database Service (Amazon RDS)

Attack Vectors

- Misconfigured Policy
- Misconfigured ACL

- No Authentication
- Misconfigured API Endpoints
- Misconfigured Policy

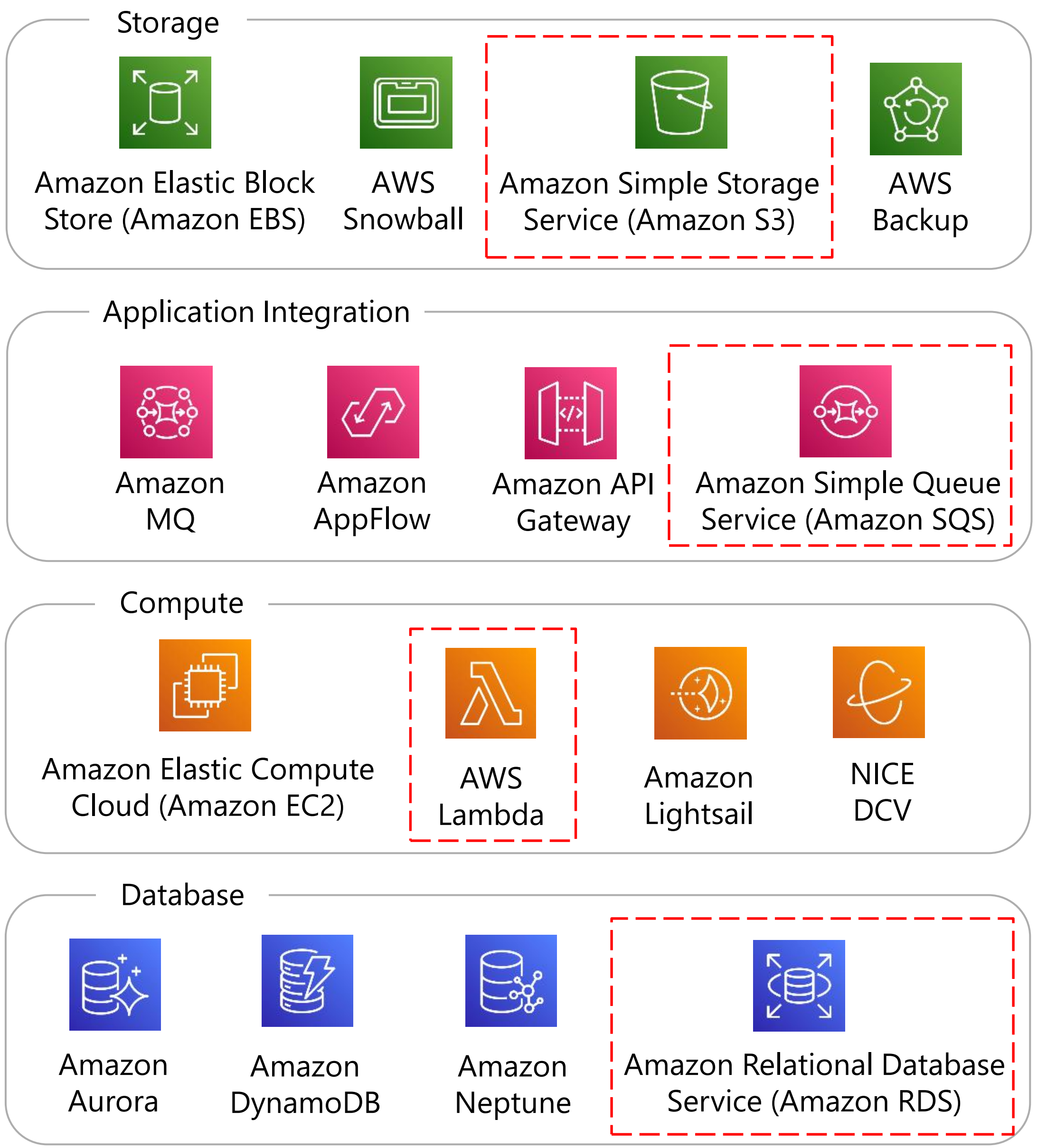
- Execution Input Validation
- Runtime Modification
- Misconfigured Policy

- Web-based
- SQL Injections
- Misconfigured Policy

1

AWS Attack Surface

A Few AWS Services



Storage

Amazon Elastic Block Store (Amazon EBS)

AWS Snowball

Amazon Simple Storage Service (Amazon S3)

AWS Backup

Application Integration

Amazon MQ

Amazon AppFlow

Amazon API Gateway

Amazon Simple Queue Service (Amazon SQS)

Compute

Amazon Elastic Compute Cloud (Amazon EC2)

AWS Lambda

Amazon Lightsail

NICE DCV

Database

Amazon Aurora

Amazon DynamoDB

Amazon Neptune

Amazon Relational Database Service (Amazon RDS)

Attack Vectors

- Misconfigured Policy
- Misconfigured ACL

- No Authentication
- Misconfigured API Endpoints
- Misconfigured Policy

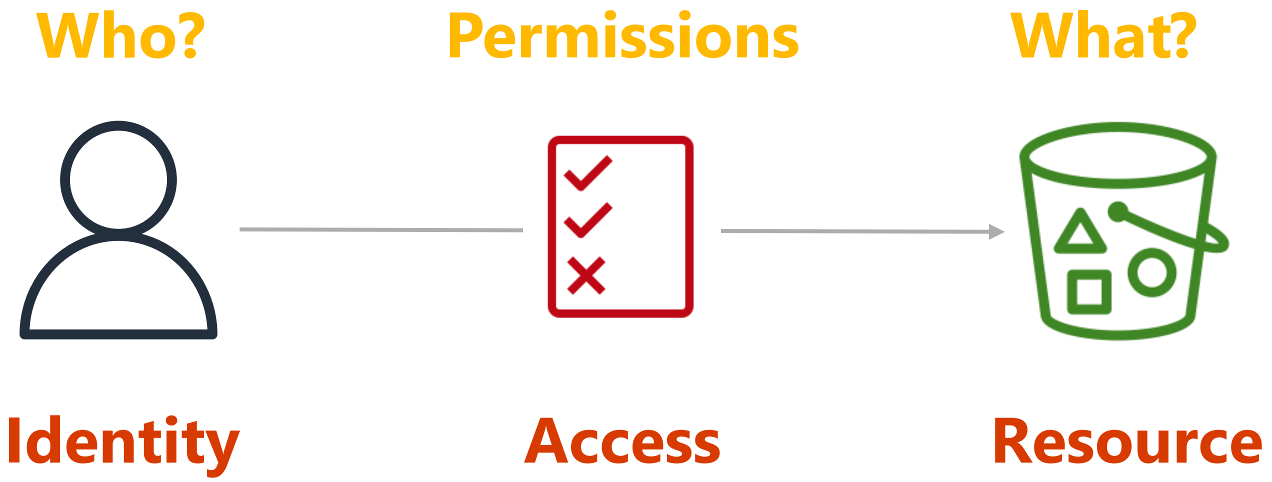
- Execution Input Validation
- Runtime Modification
- Misconfigured Policy

- Web-based
- SQL Injections
- Misconfigured Policy

3

AWS Identity and Access Management (IAM)

- Manages access to AWS resources
- Allows granular permissions
- Enables Multi-factor authentication (MFA)
- Permits Identity federation
- Integrated with many AWS services

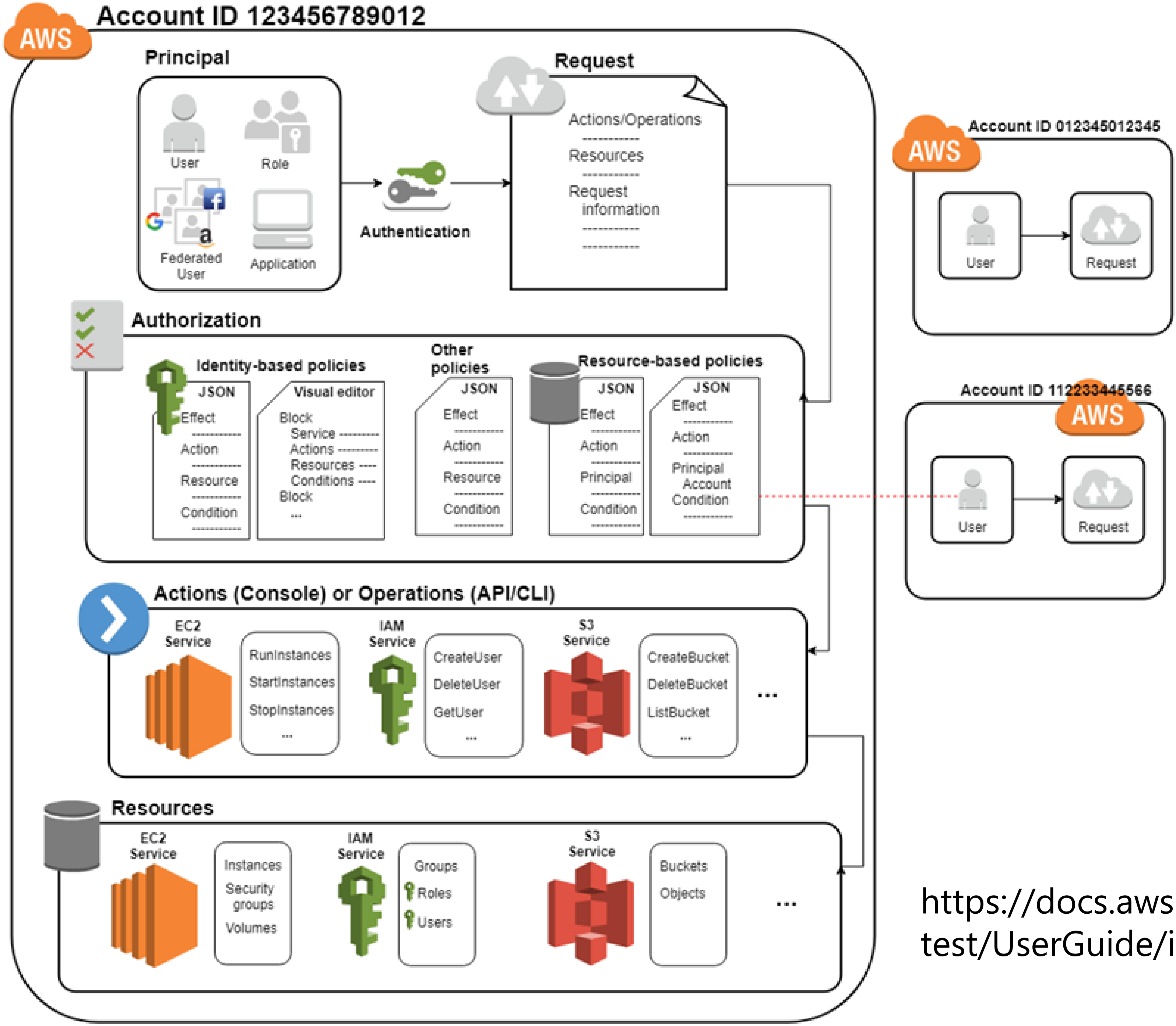


Who? Permissions What?

Identity Access Resource

AWS IAM

How Does It Work?



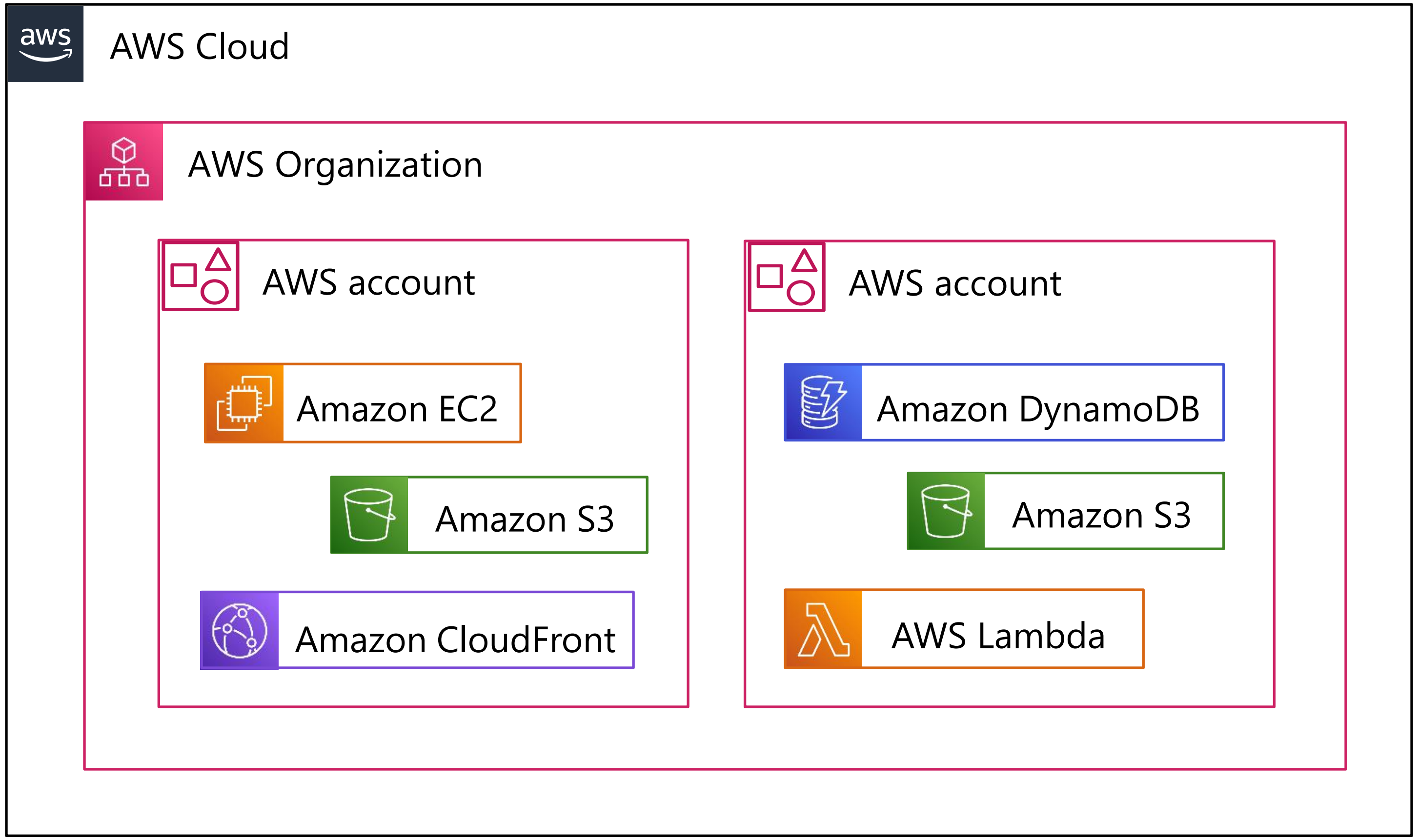
https://docs.aws.amazon.com/IAM/latest/UserGuide/intro-structure.html

AWS Accounts &

AWS IAM Users, Groups and Roles

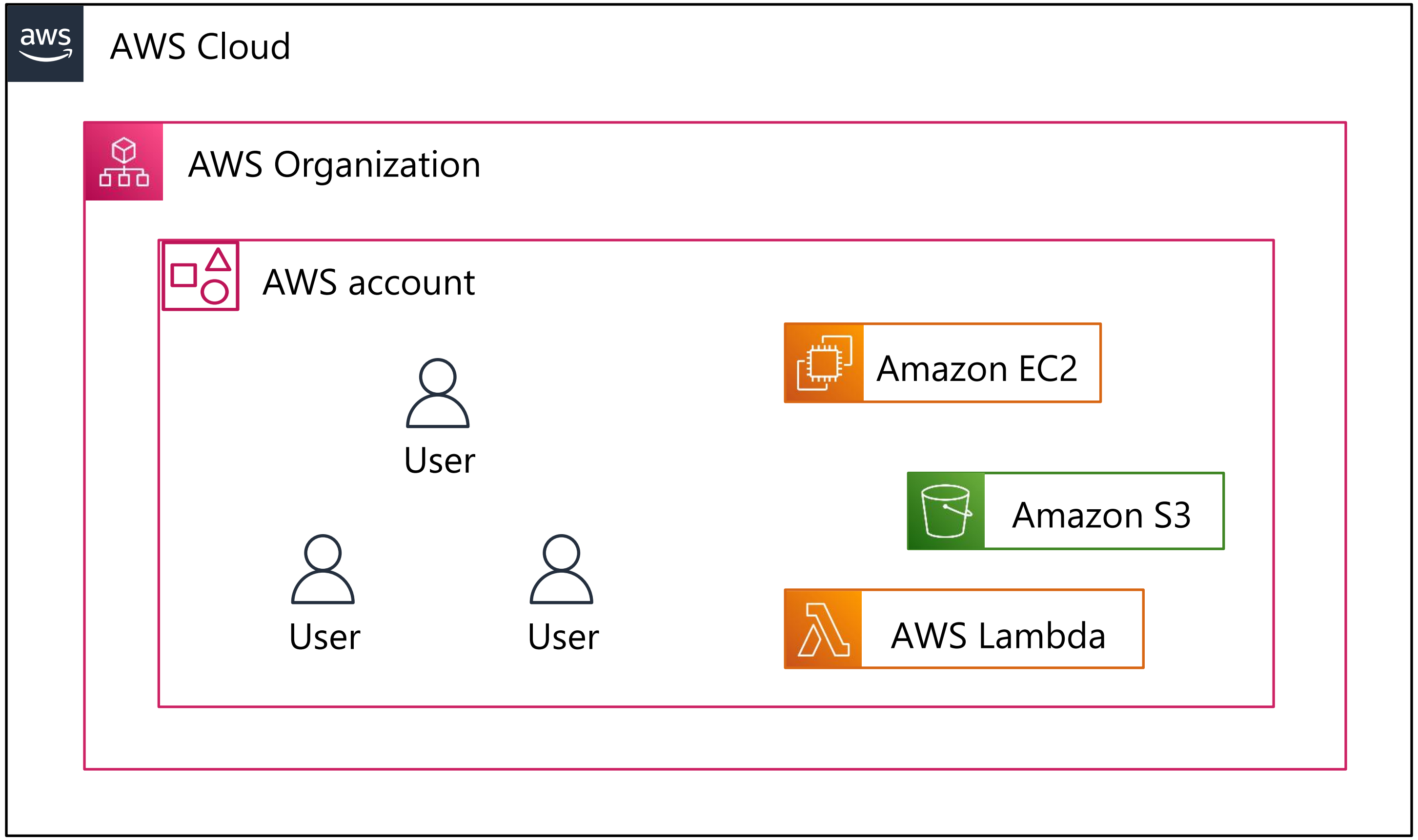
AWS Account

- A resource container for AWS cloud services
- Isolates resources (explicit security boundary)
- AWS organizations allow you to organize accounts



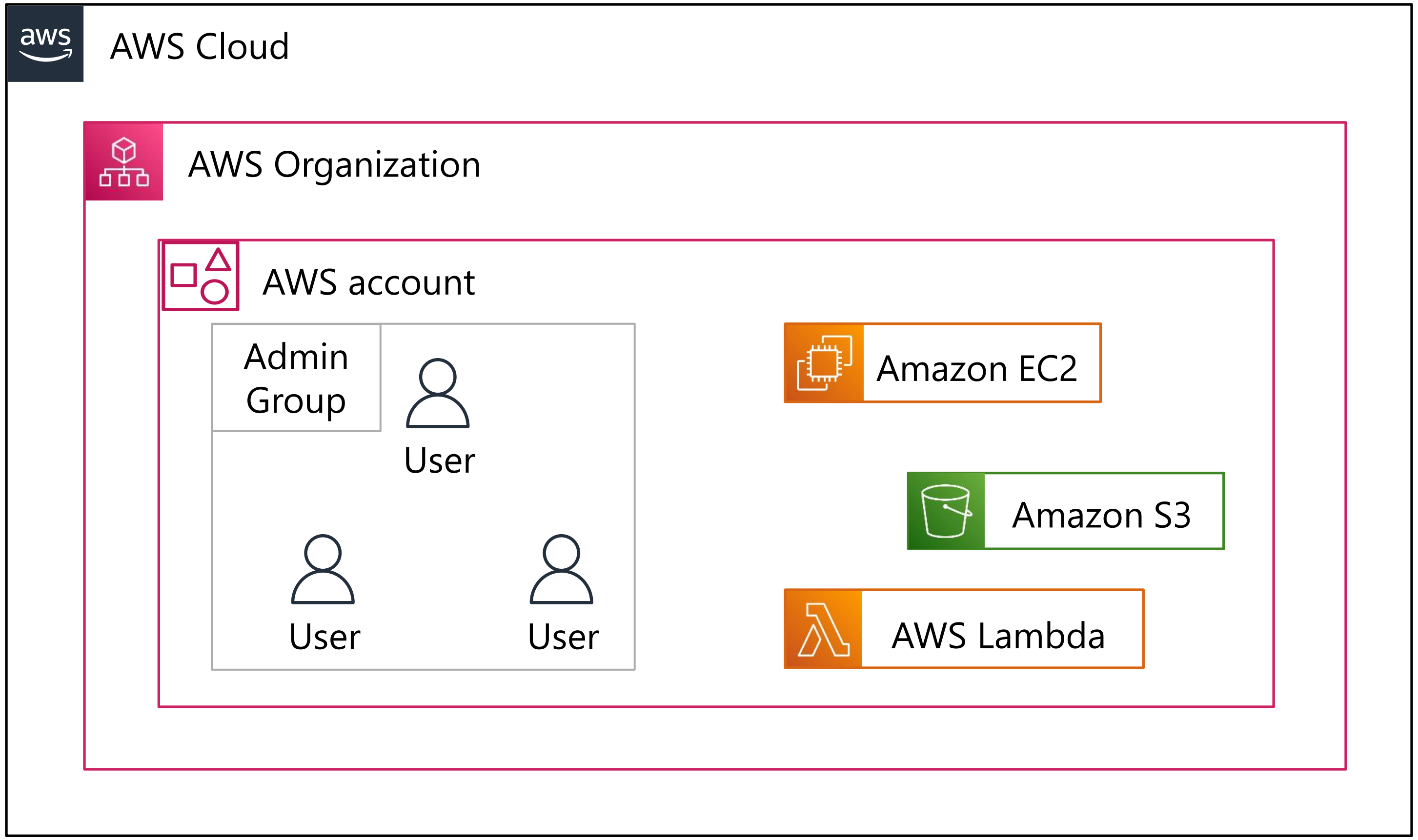
AWS IAM Users

- Created within AWS accounts
- Passwords to access the AWS Mgmt. Console
- Access keys to make programmatic requests



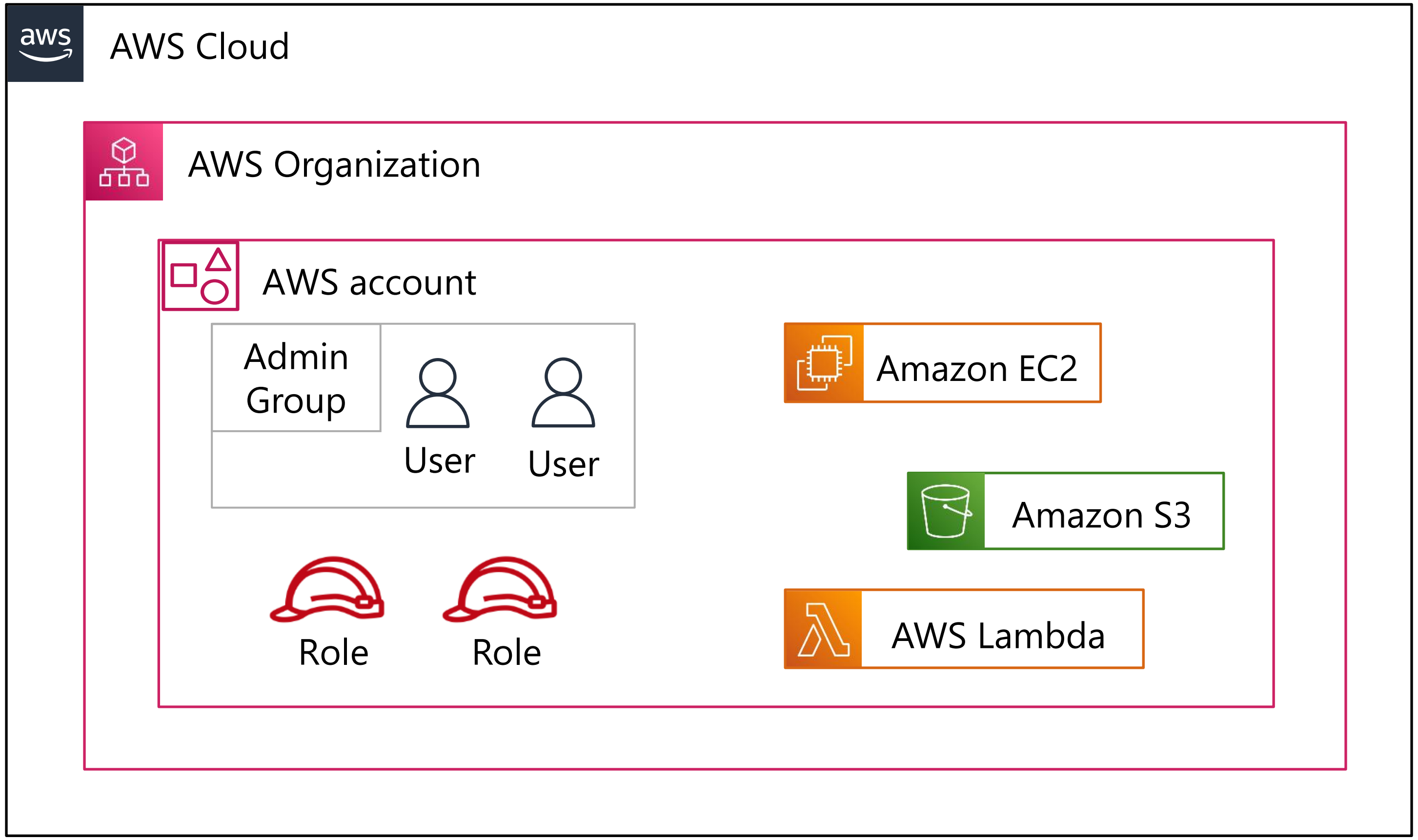
AWS IAM Groups

- Collection of IAM users
- Define permissions for multiple users
- Can only contain users and not other groups



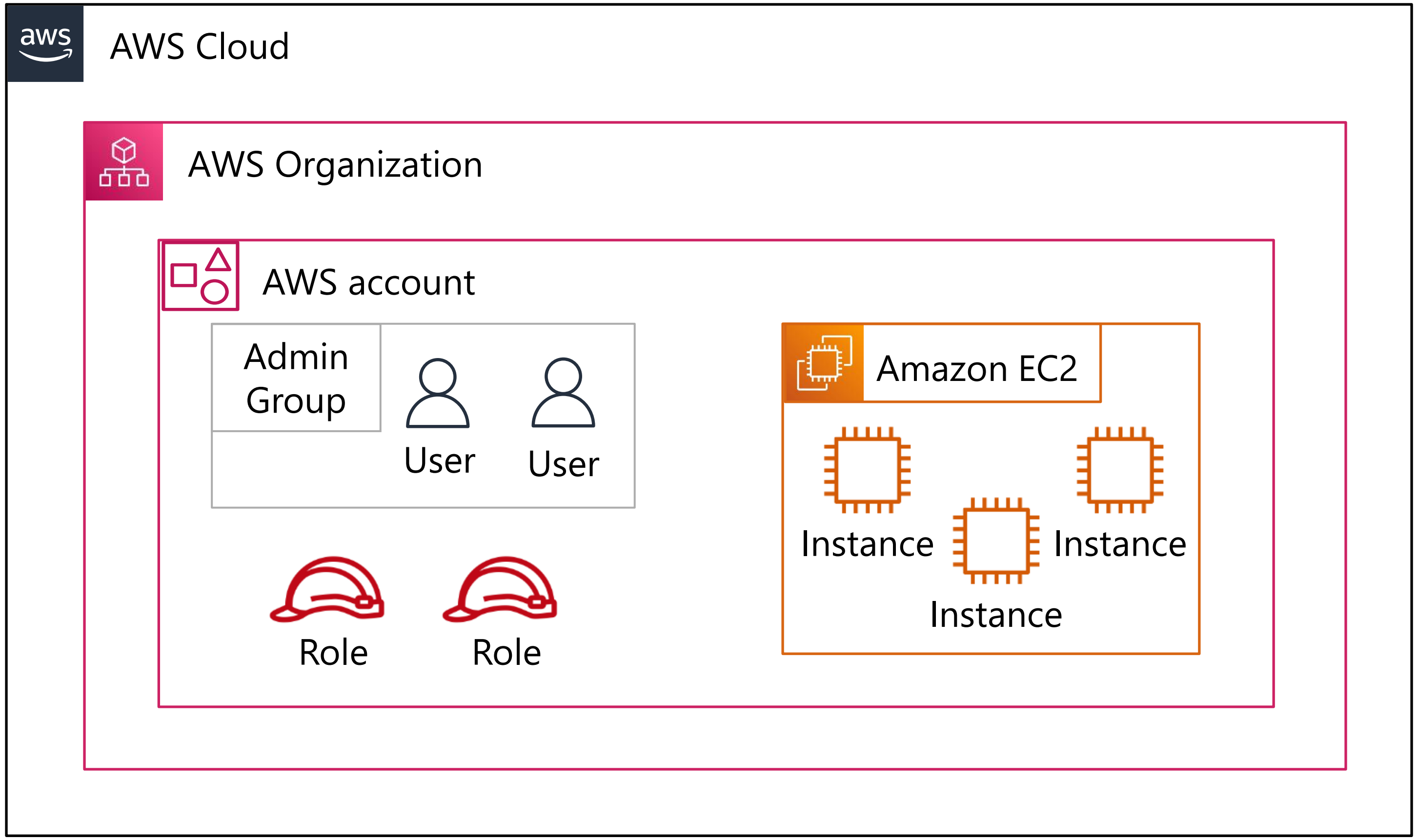
AWS IAM Roles

- IAM identity that has specific permissions
- Assumed by anyone who needs it (i.e. User, App)
- Temporary security credentials

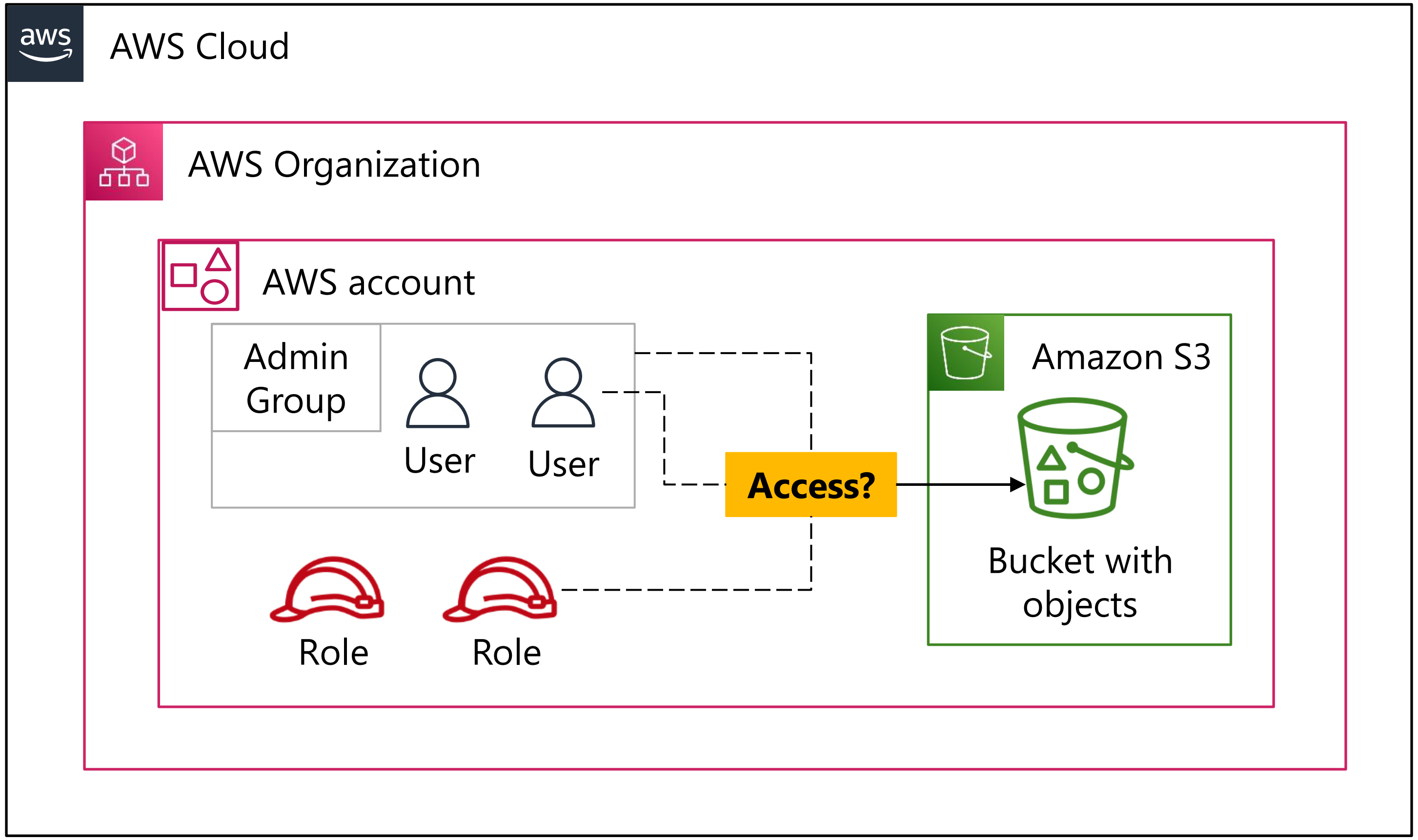


AWS IAM Roles

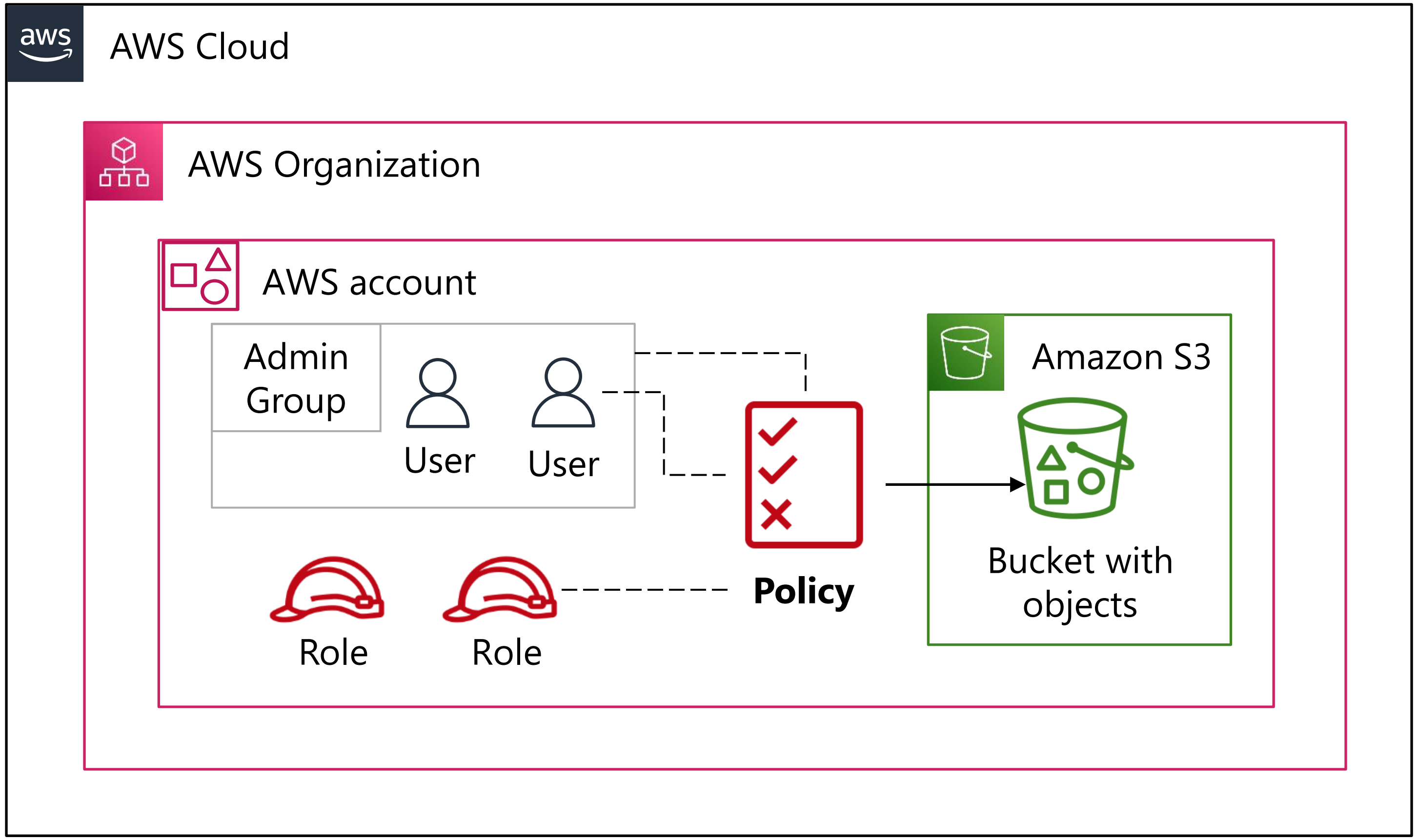
- IAM identity that has specific permissions
- Assumed by anyone who needs it (i.e. User, App)
- Temporary security credentials



How Do We Manage Access to Resources?



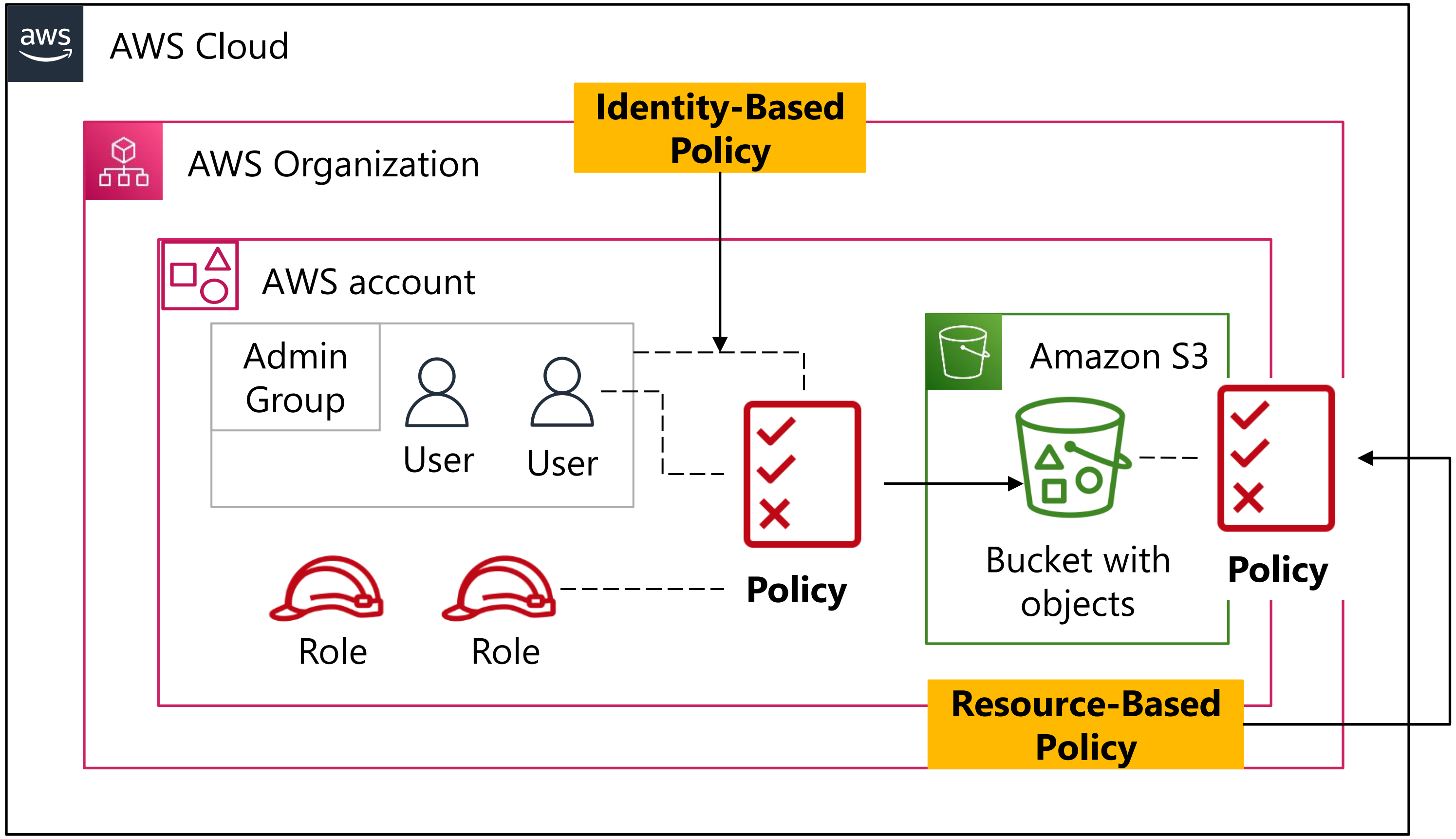
How Do We Manage Access to Resources?



IAM Policies

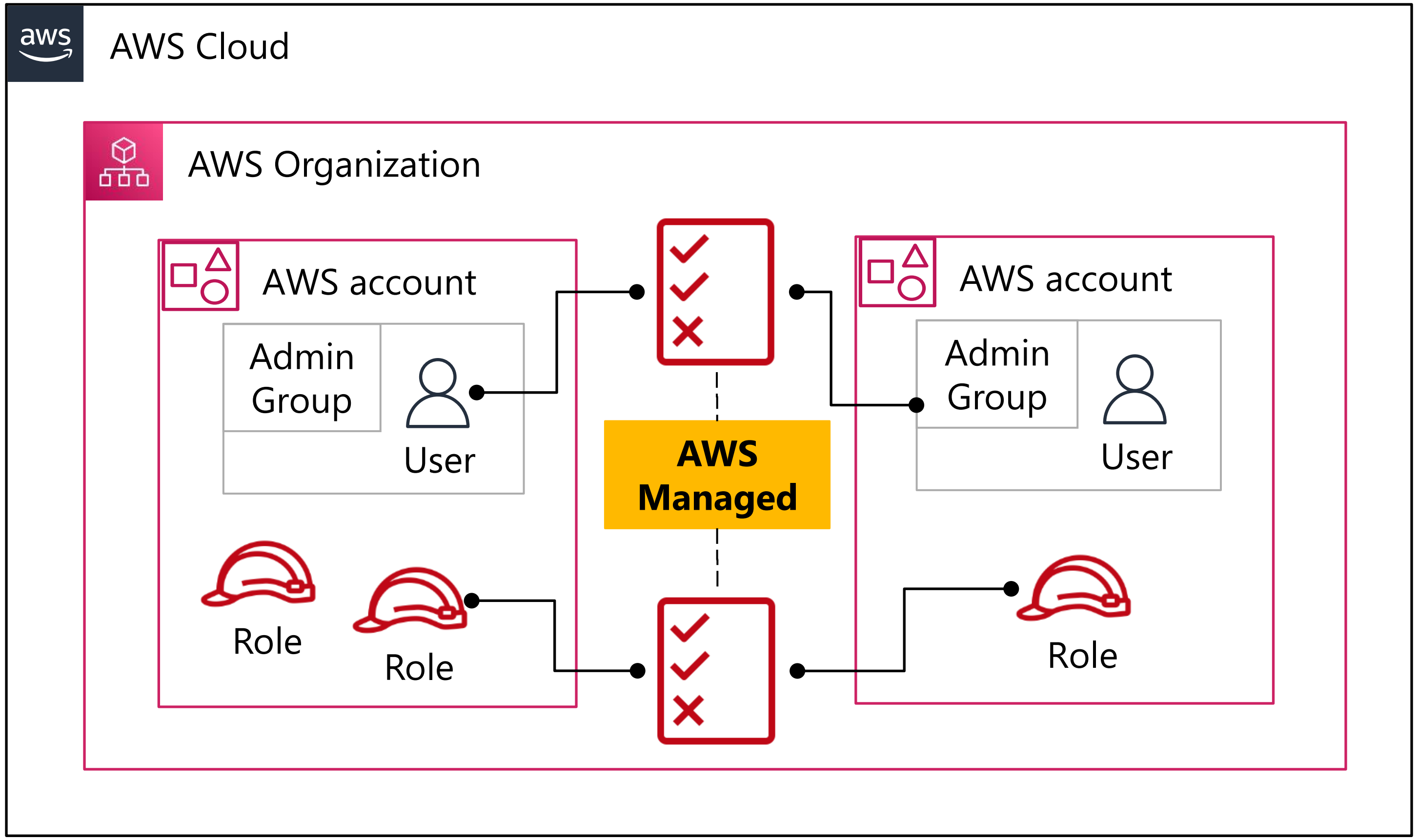
IAM Policies

- Determine whether to allow or deny access
- Define identity or resource permissions.
- Can be attached to IAM identities (users, groups of users, or roles) or AWS resources



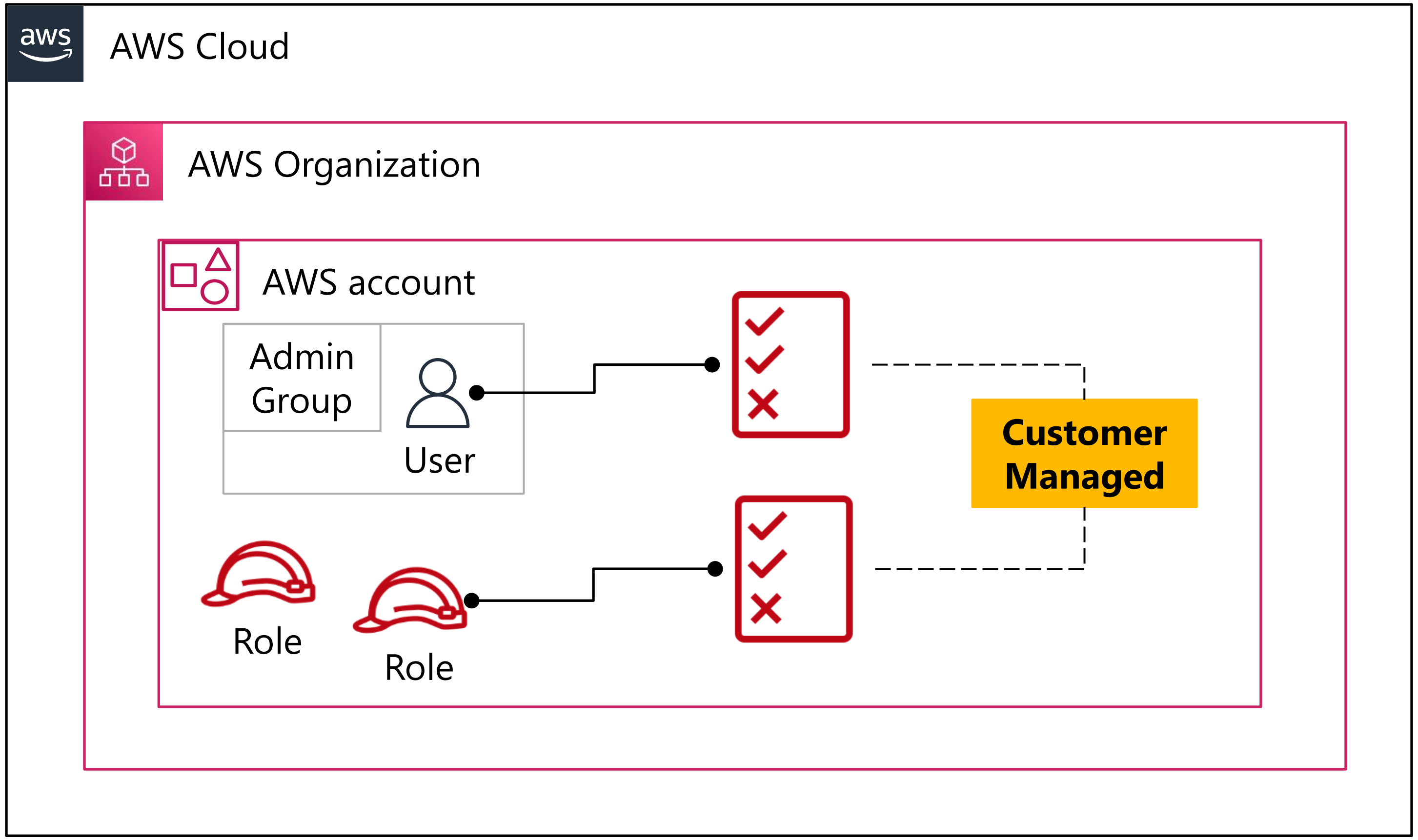
IAM Policies
AWS Managed

- Created and managed by AWS
- You cannot modify the policy
- Can be attached to identities in different accounts



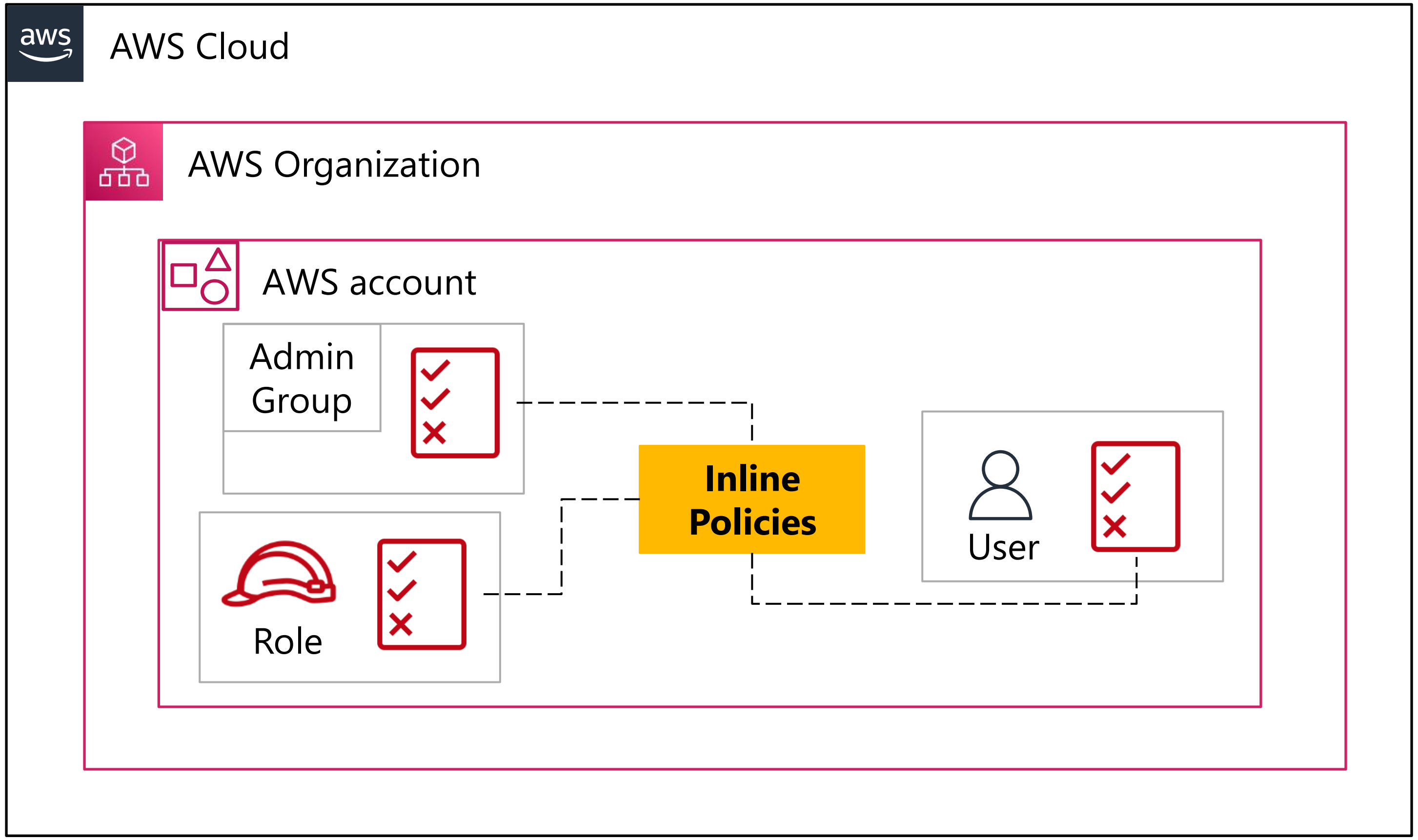
IAM Policies Customer Managed

- Created and managed within Account
- Use an AWS managed policy to start your own
- Can be attached to identities in one account



IAM Policies
Inline Policies

- Embedded in IAM identities (user, group or role)
- A strict one-to-one relationship
- Policy is part of the identity (not reusable)



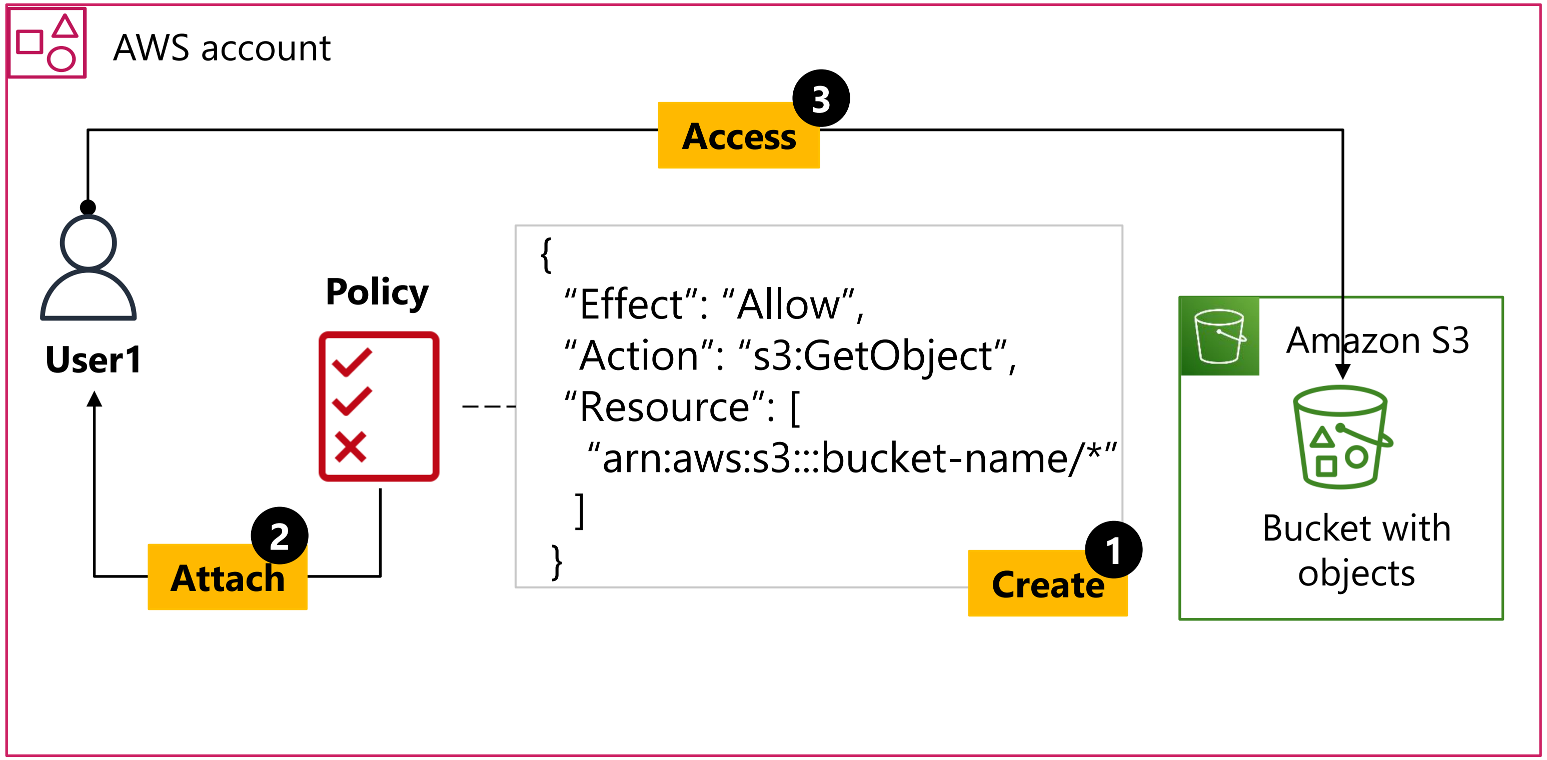
IAM Policies

User -> Accessing -> S3 Bucket

AWS IAM User Accessing S3 Bucket

Identity-based

What actions can be done on which buckets

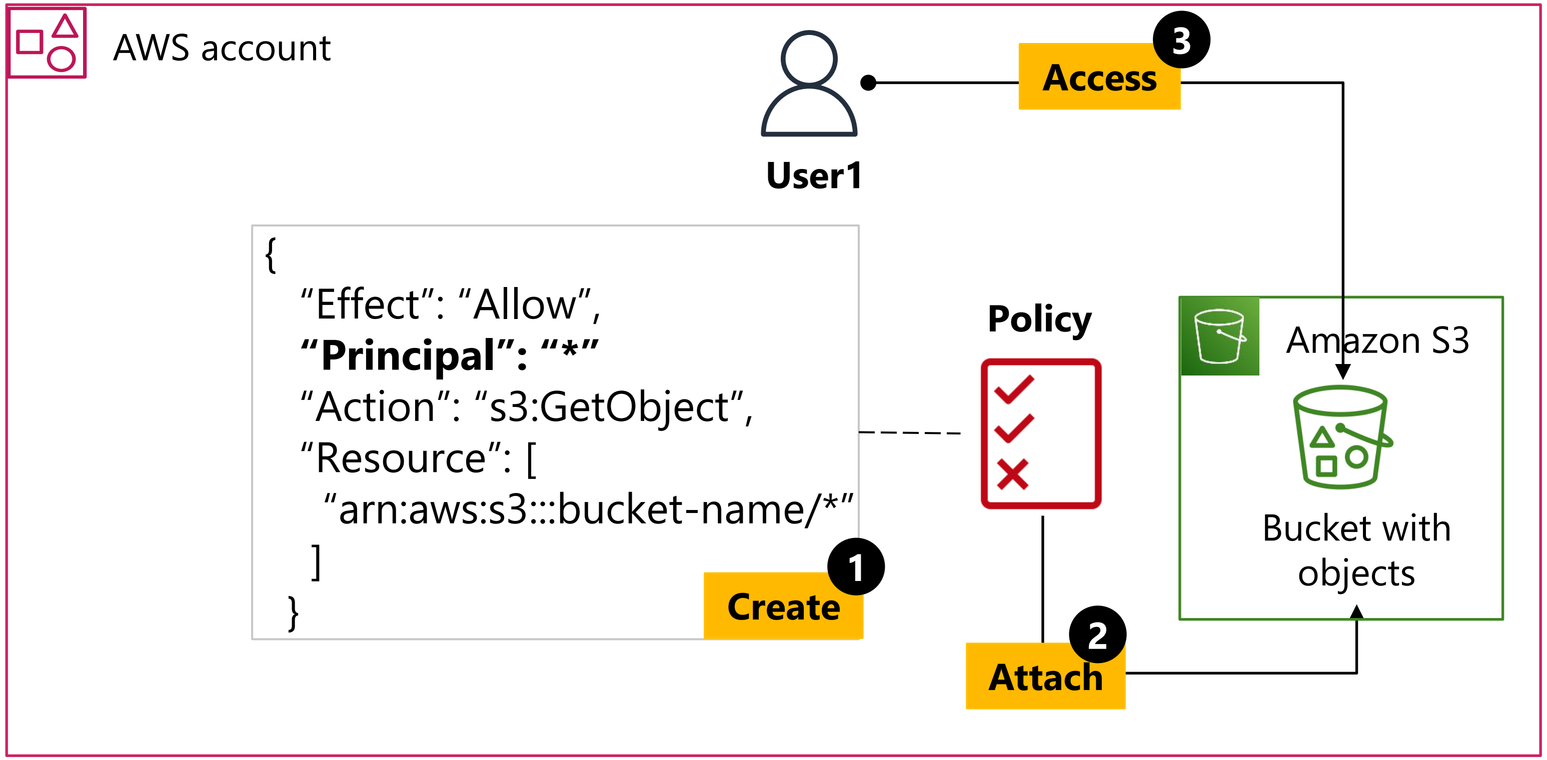


```
{
  "Effect": "Allow",
  "Action": "s3:GetObject",
  "Resource": [
    "arn:aws:s3:::bucket-name/*"
  ]
}
```

AWS IAM User Accessing S3 Bucket

Resource-based

Who can do what actions on one bucket

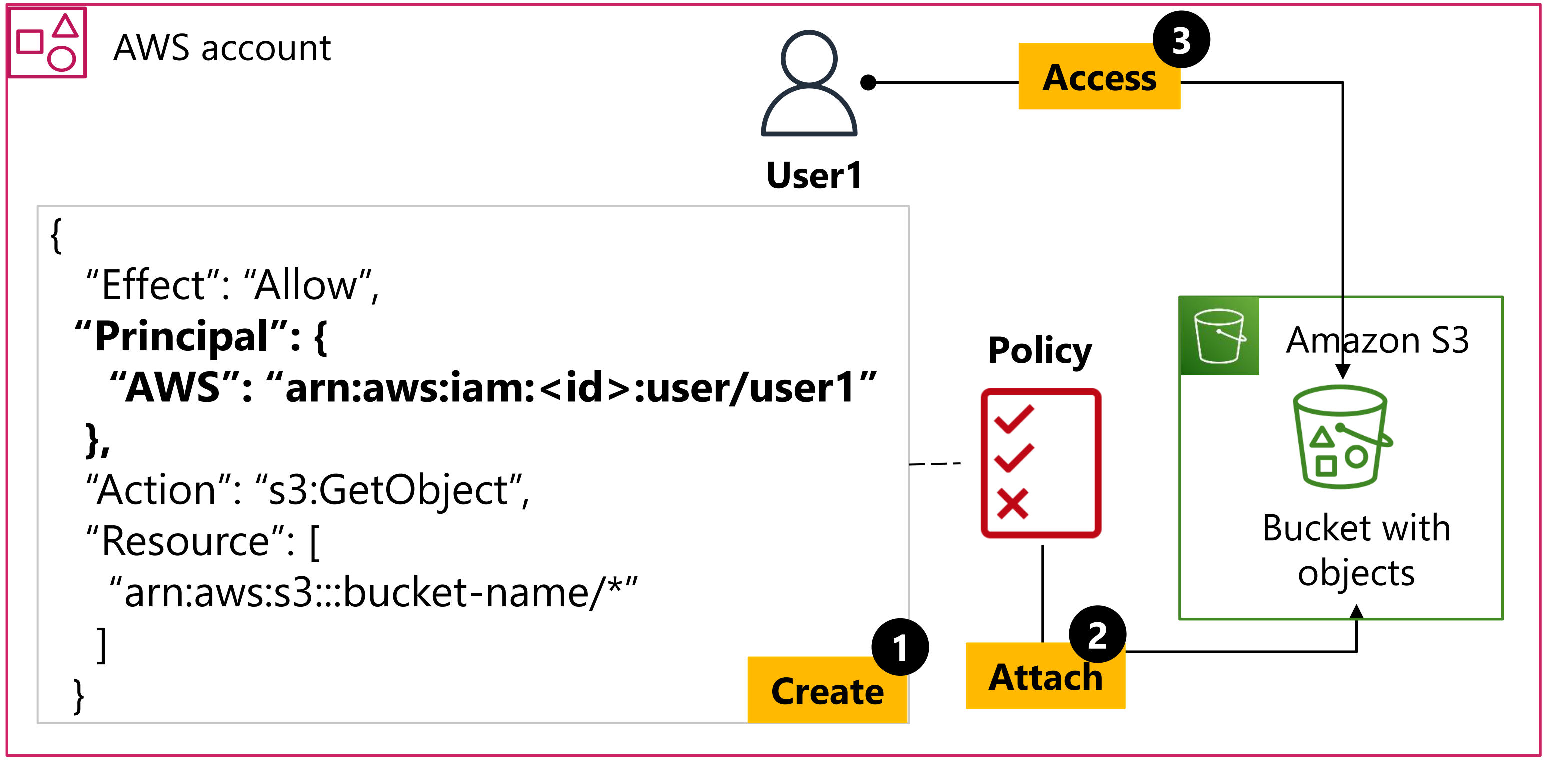


```
{
  “Effect”: “Allow”,
  “Principal”: “*”
  “Action”: “s3:GetObject”,
  “Resource”: [
    “arn:aws:s3:::bucket-name/*”
   ]
 }
```

AWS IAM User Accessing S3 Bucket

Resource-based

Who can do what actions on one bucket

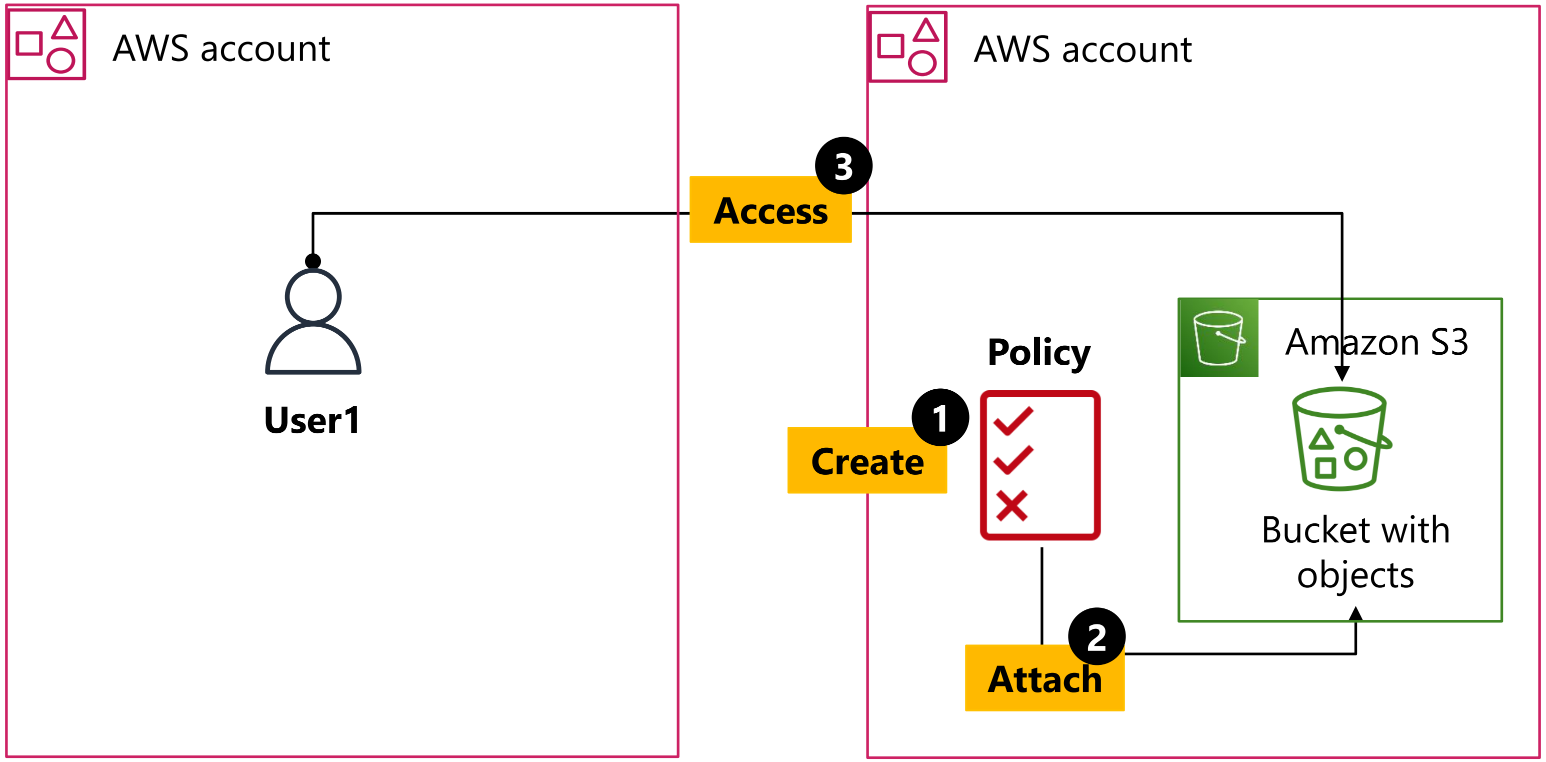


```
{
  "Effect": "Allow",
 "Principal": {
   "AWS": "arn:aws:iam:<id>:user/user1"
  },
  "Action": "s3:GetObject",
  "Resource": [
   "arn:aws:s3:::bucket-name/*"
   ]
 }
```

AWS IAM User Accessing S3 Bucket

Resource-based

Who can do what actions on one bucket

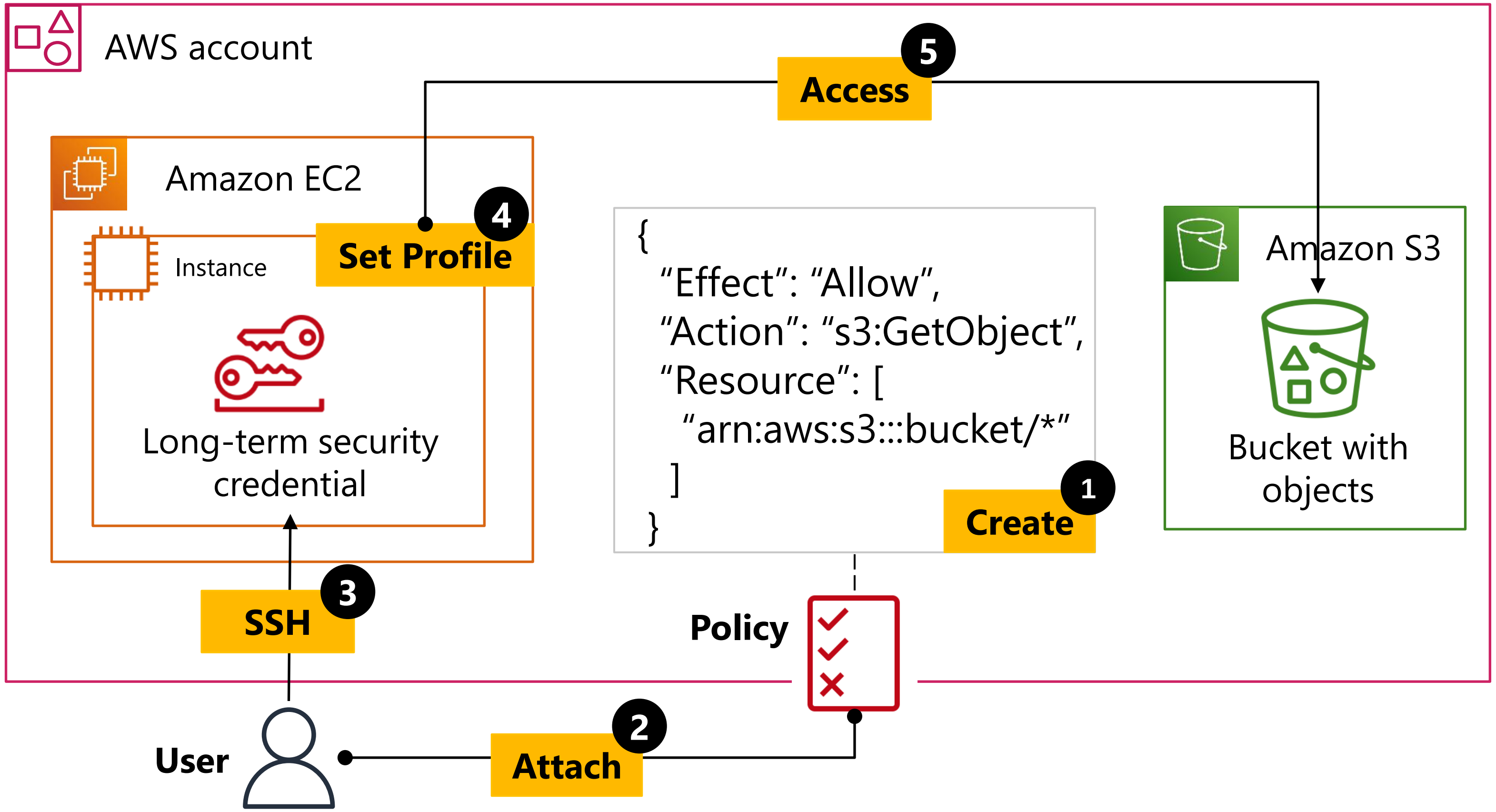


AWS IAM

EC2 Instance (App) -> Accessing -> S3 Bucket

EC2 Accessing S3 Bucket (Basic)

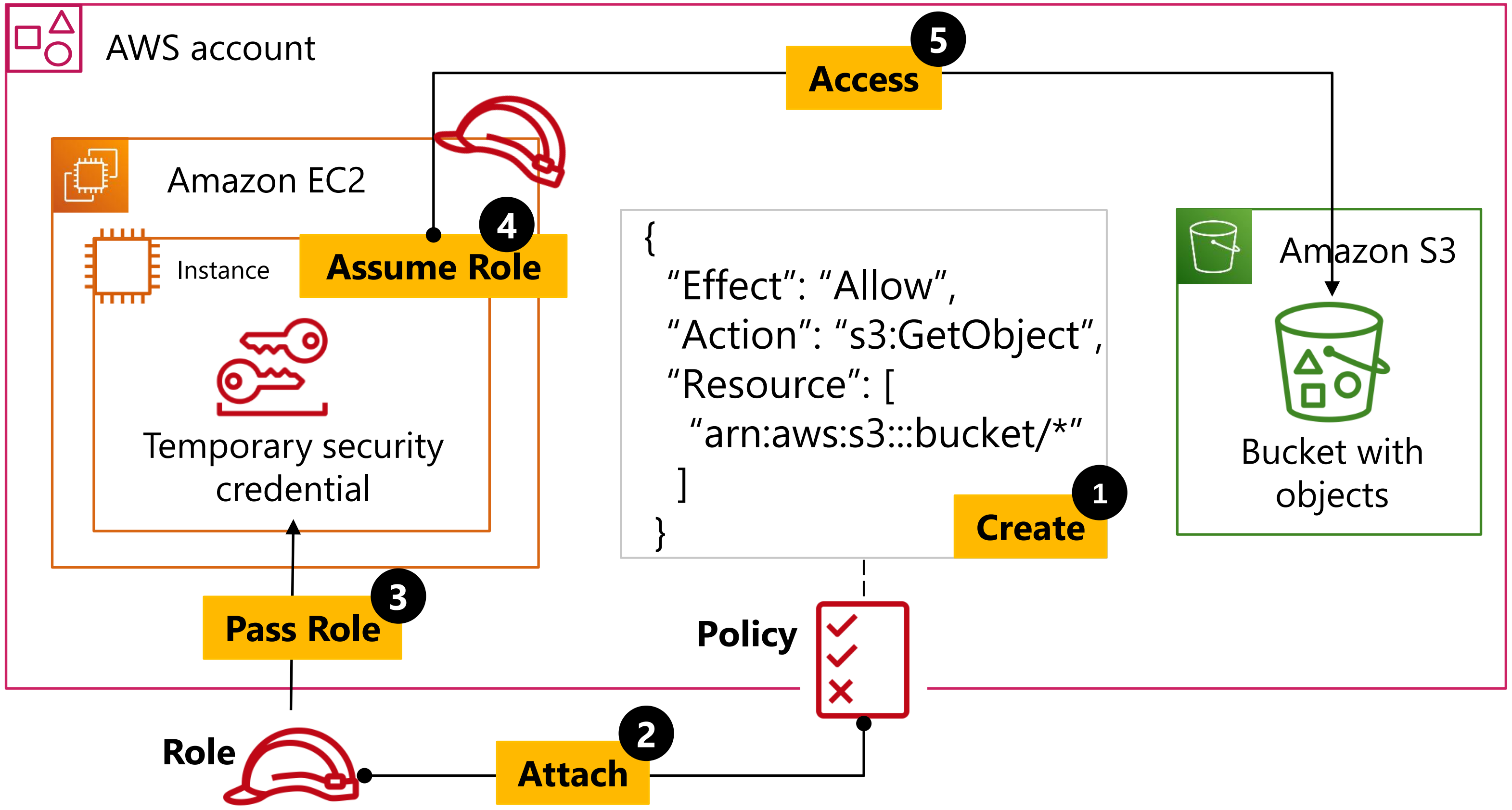
- **Create** policy to allow access to an S3 bucket
- **Attach** policy to **IAM user**
- **Configure** a profile with user access key
- **Access** the S3 bucket with the saved profile



```
{
  "Effect": "Allow",
  "Action": "s3:GetObject",
  "Resource": [
    "arn:aws:s3:::bucket/*"
  ]
}
```

EC2 Accessing S3 Bucket with an IAM Role

- **Create** policy to allow access to an S3 bucket
- **Attach** policy to **IAM role**
- **Pass IAM role** to EC2 service (instance)
- **Assume** role and **access** the S3 bucket



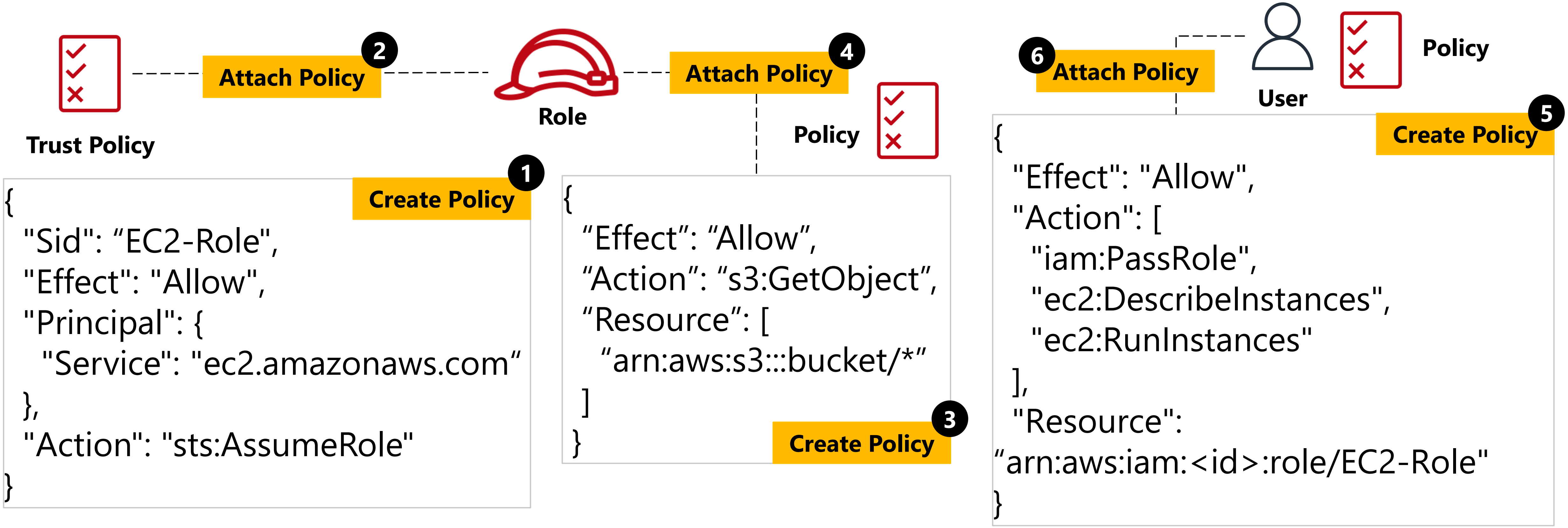
```
{
  "Effect": "Allow",
  "Action": "s3:GetObject",
  "Resource": [
    "arn:aws:s3:::bucket/*"
  ]
}
```

AWS IAM

Passing the Role

Recipe to Pass a Role (PassRole != API Call)

- A trust policy for the role that allows the service to assume the role
- An IAM policy attached to the role that determines what the role can do
- An IAM policy attached to the IAM user that allows the user to pass roles



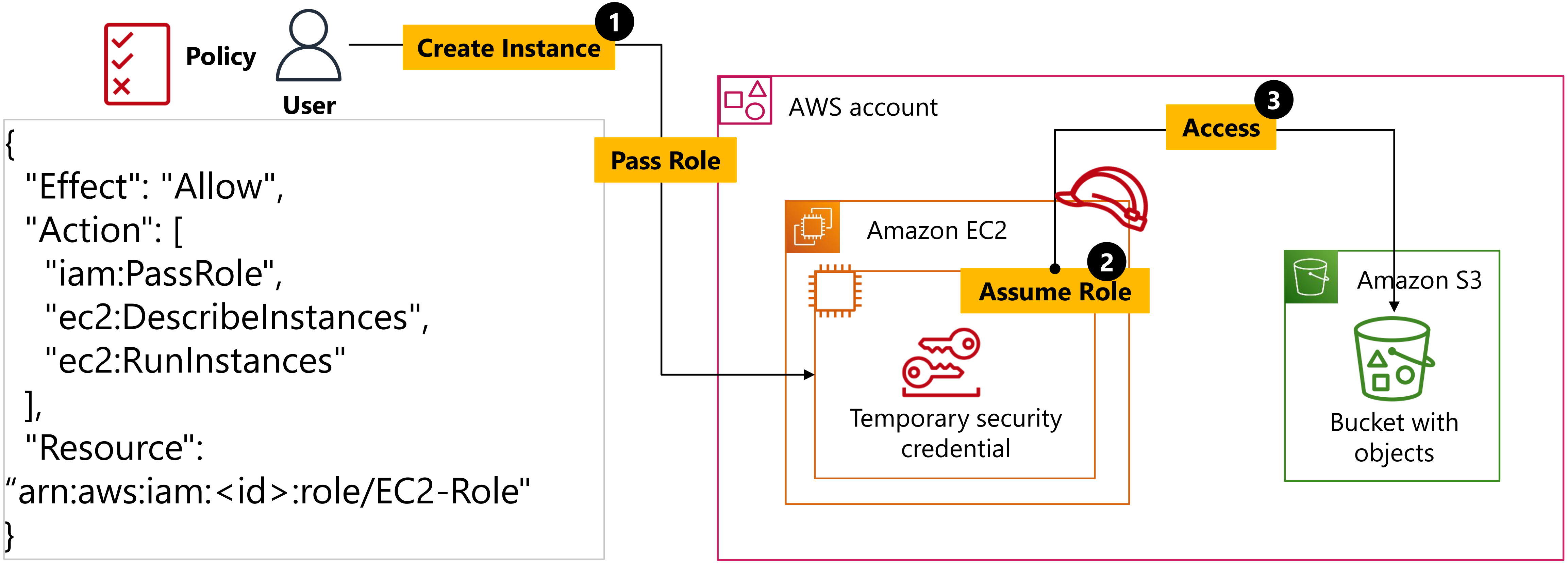
```
{
  "Sid": “EC2-Role",
  "Effect": "Allow",
  "Principal": {
    "Service": "ec2.amazonaws.com“
  },
  "Action": "sts:AssumeRole"
}
```

```
{
  “Effect”: “Allow”,
  “Action”: “s3:GetObject”,
  “Resource”: [
    “arn:aws:s3:::bucket/*”
  ]
}
```

```
{
  "Effect": "Allow",
  "Action": [
    "iam:PassRole",
    "ec2:DescribeInstances",
    "ec2:RunInstances"
  ],
  "Resource":
“arn:aws:iam:<id>:role/EC2-Role"
}
```

Recipe to Pass a Role (PassRole != API Call)

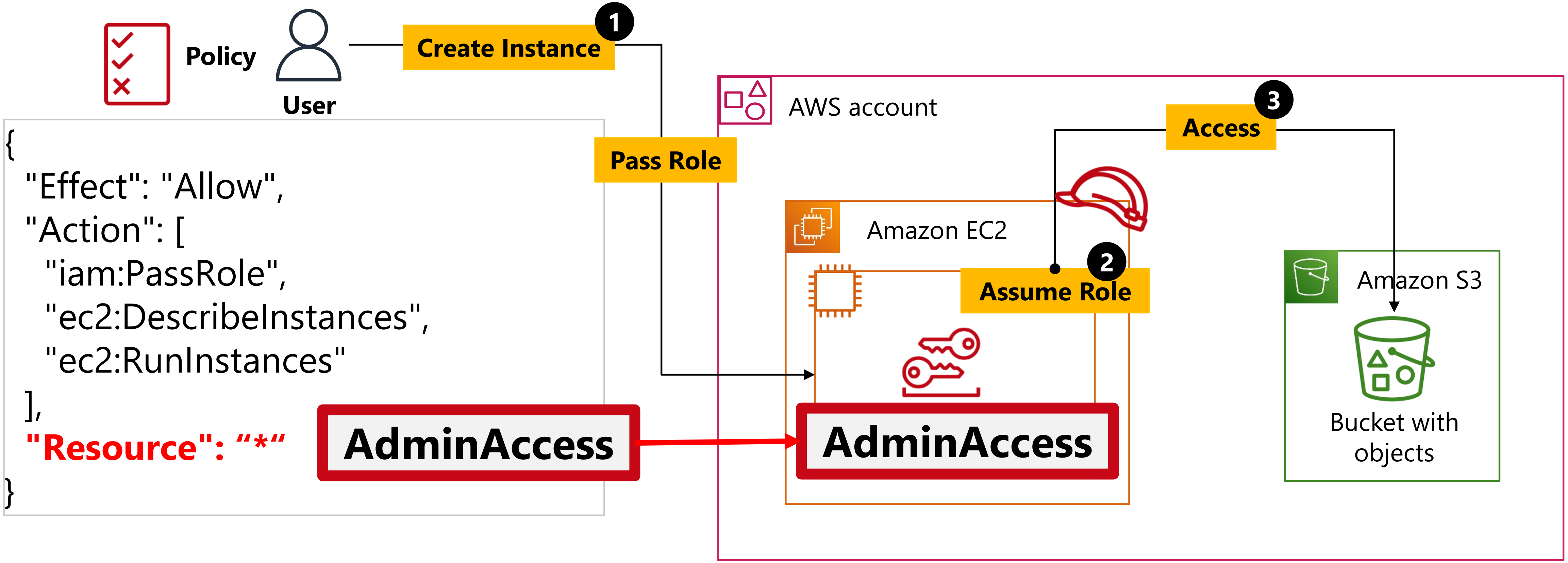
- A trust policy for the role that allows the service to assume the role
- An IAM policy attached to the role that determines what the role can do
- An IAM policy attached to the IAM user that allows the user to pass roles



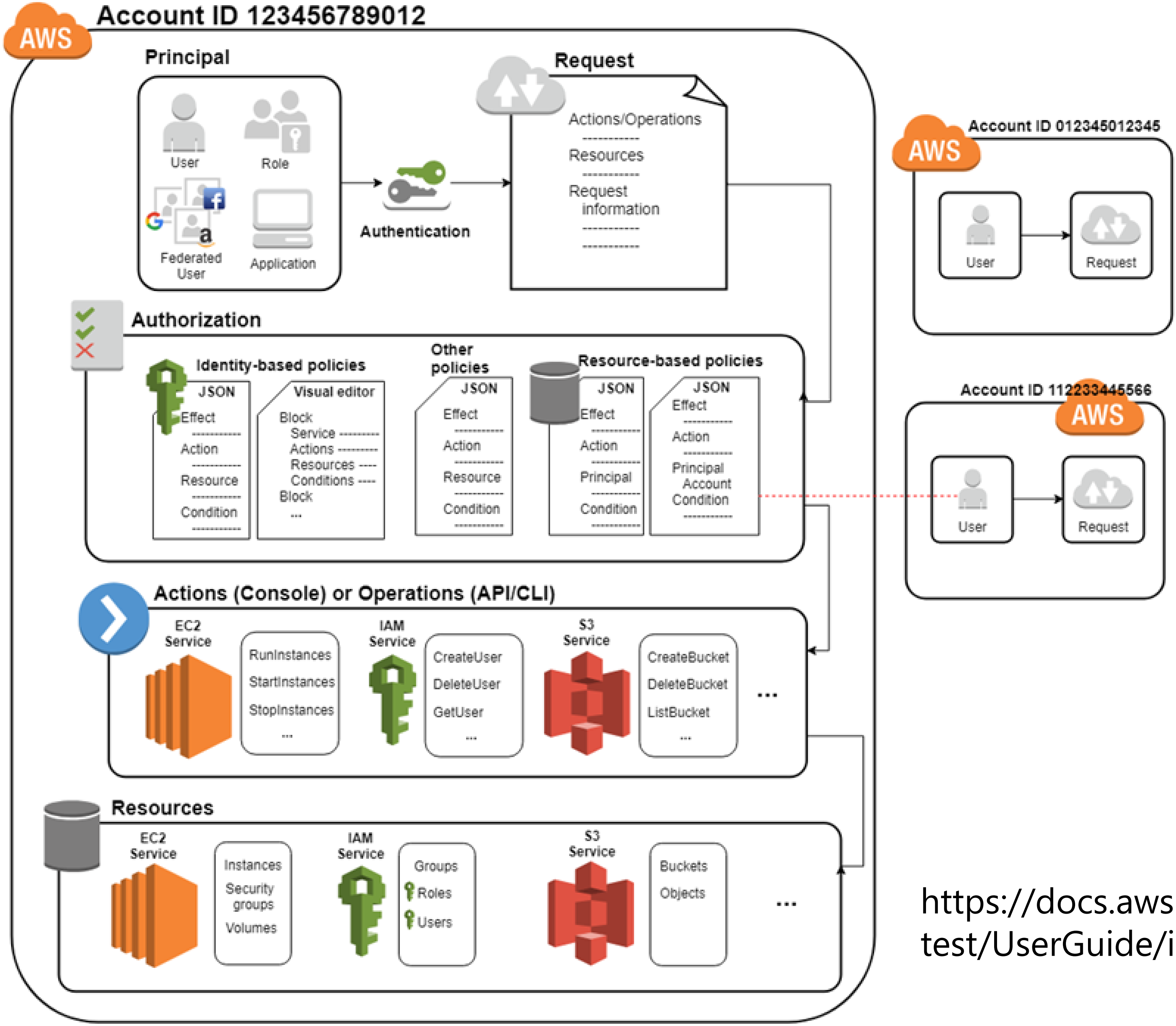
```
{
  "Effect": "Allow",
  "Action": [
    "iam:PassRole",
    "ec2:DescribeInstances",
    "ec2:RunInstances"
  ],
  "Resource":
“arn:aws:iam:<id>:role/EC2-Role"
}
```

Recipe to Pass a Role (PassRole != API Call)

- A trust policy for the role that allows the service to assume the role
- An IAM policy attached to the role that determines what the role can do
- An IAM policy attached to the IAM user that allows the user to pass roles



```
{
  "Effect": "Allow",
  "Action": [
    "iam:PassRole",
    "ec2:DescribeInstances",
    "ec2:RunInstances"
  ],
  "Resource": “*“
}
```



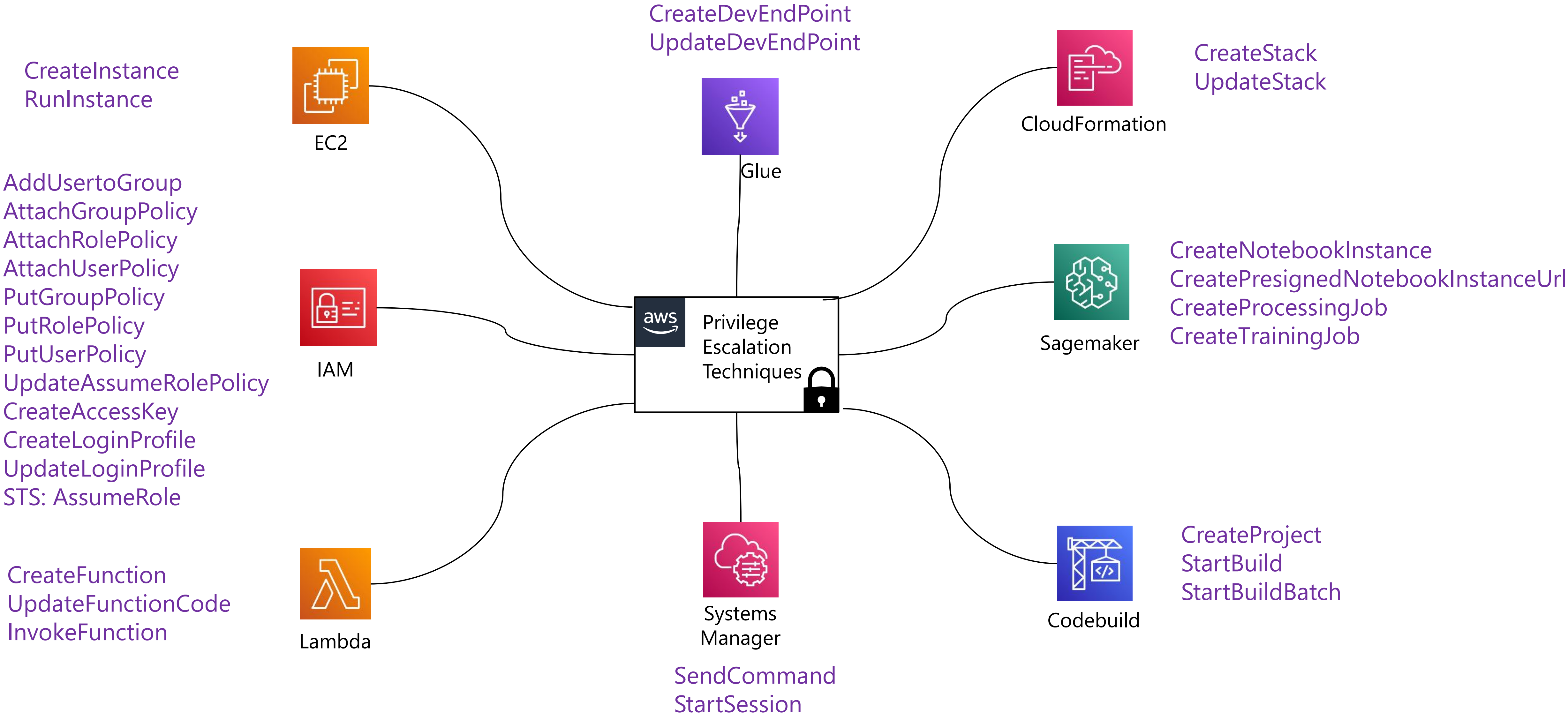
https://docs.aws.amazon.com/IAM/latest/UserGuide/intro-structure.html

4

AWS IAM Attack - Defend

- Privilege Escalation is a common tactic to take advantage of IAM misconfigurations
- 31 different IAM Privilege escalation techniques
  - Across 8 different AWS Services
  - 50 % techniques are for AWS IAM and atomic in nature.
  - 30 % techniques involves abusing Passrole permission to services
- Attack Scenario Deep Dive – Create Policy Version
- End-to-End Attack Scenario

Mapping Techniques to Services and Actions



Privilege Escalation Techniques

EC2

CreateInstance
RunInstance

Glue

CreateDevEndPoint
UpdateDevEndPoint

CloudFormation

CreateStack
UpdateStack

IAM

AddUsertoGroup
AttachGroupPolicy
AttachRolePolicy
AttachUserPolicy
PutGroupPolicy
PutRolePolicy
PutUserPolicy
UpdateAssumeRolePolicy
CreateAccessKey
CreateLoginProfile
UpdateLoginProfile
STS: AssumeRole

Sagemaker

CreateNotebookInstance
CreatePresignedNotebookInstanceUrl
CreateProcessingJob
CreateTrainingJob

Lambda

CreateFunction
UpdateFunctionCode
InvokeFunction

Systems Manager

SendCommand
StartSession

Codebuild

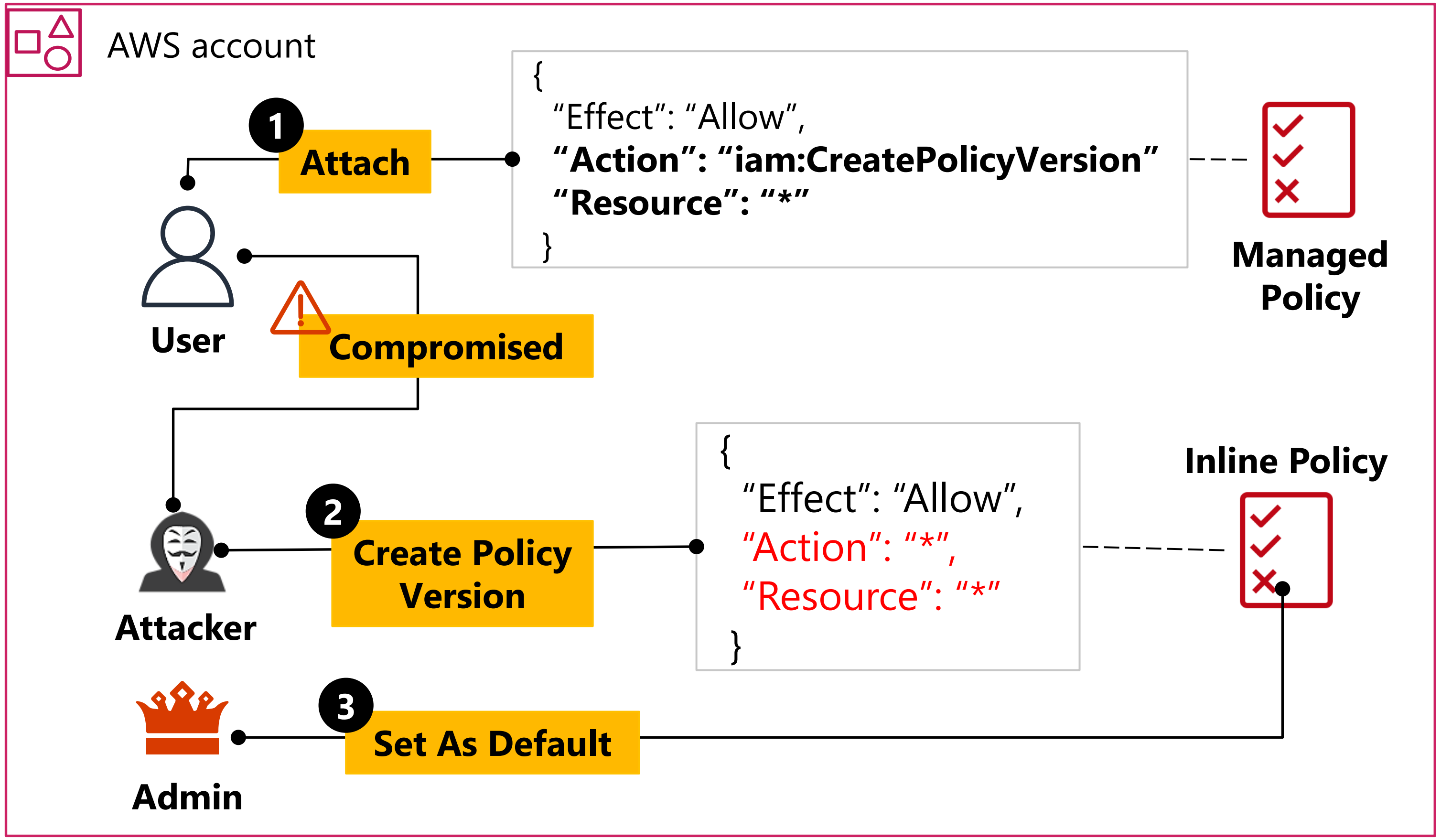
CreateProject
StartBuild
StartBuildBatch

AWS IAM Attack- Defend

Privilege Escalation – CreatePolicyVersion

CreatePolicyVersion Attacker Recipe

- Attacker compromises low privileged user
- User has policy attached (iam:CreatePolicyVersion).
- Attacker creates new policy version with inline admin policy (Allow * on all resources) and attaches to role.
- Attacker sets newly created policy version as default



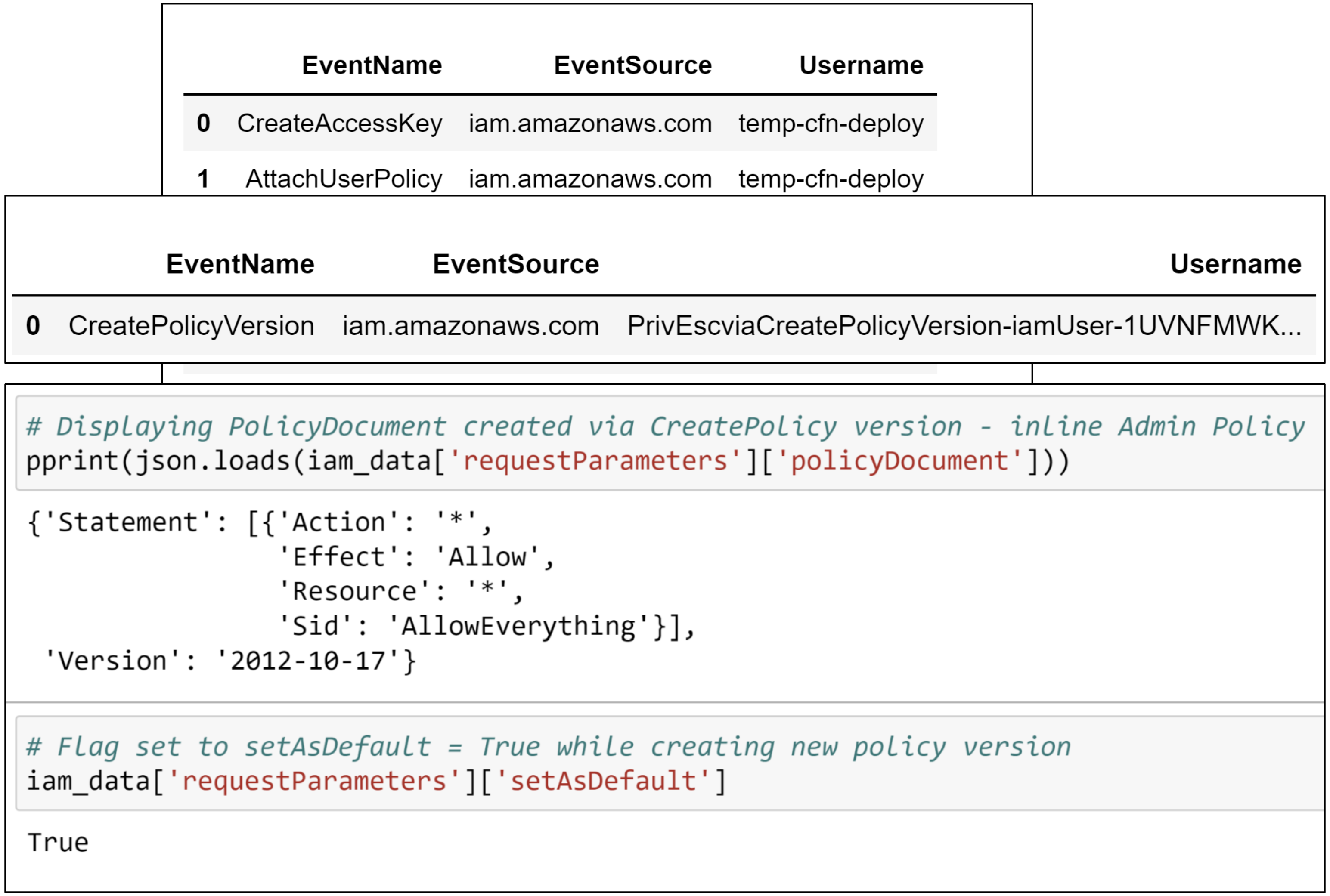
```
{
  "Effect": "Allow",
  "Action": "iam:CreatePolicyVersion"
  "Resource": "*"
}
```

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

Demo

CreatePolicyVersion Attacker Recipe

- Detect the creation of high privileged policies.
- Use CloudTrail log to monitor for single API **CreatePolicyVersion** with **SetDefault as True**.



	EventName	EventSource	Username
0	CreateAccessKey	iam.amazonaws.com	temp-cfn-deploy
1	AttachUserPolicy	iam.amazonaws.com	temp-cfn-deploy

	EventName	EventSource	Username
0	CreatePolicyVersion	iam.amazonaws.com	PrivEscviaCreatePolicyVersion-iamUser-1UVNFMWK...

```
# Displaying PolicyDocument created via CreatePolicy version - inline Admin Policy
pprint(json.loads(iam_data['requestParameters']['policyDocument']))
```

```
{'Statement': [{'Action': '*',
                'Effect': 'Allow',
                'Resource': '*',
                'Sid': 'AllowEverything'}],
 'Version': '2012-10-17'}
```

```
# Flag set to setAsDefault = True while creating new policy version
iam_data['requestParameters']['setAsDefault']
```

```
True
```

AWS IAM Attack- Defend

End-to-End Scenario

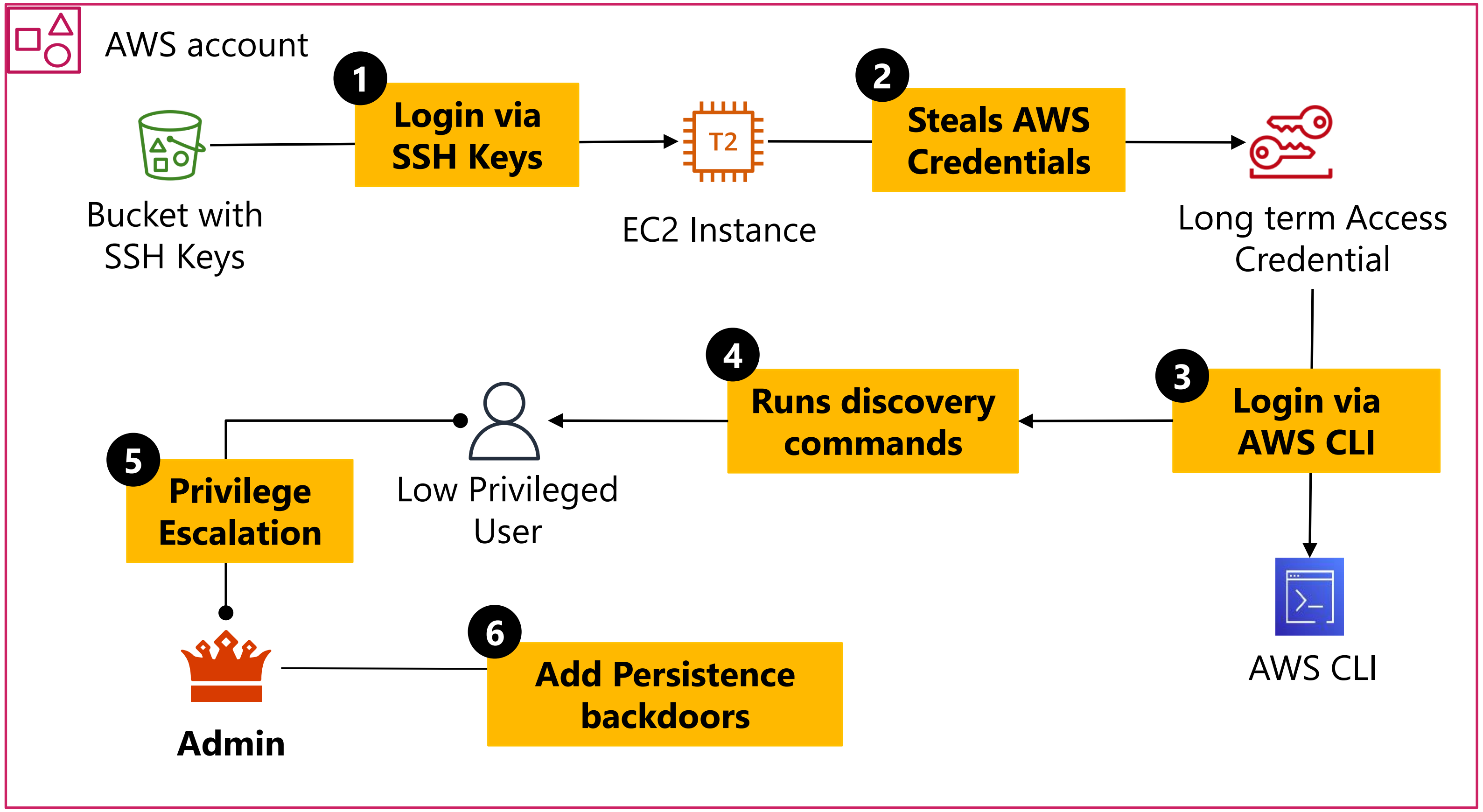
End to End Attack Scenario

Initial Access:

- Attacker finds SSH keys on publicly exposed buckets
- Login to SSH server via default accounts using SSH Keys.
- Scan host and steals aws creds from local files/env variables.
- Logs in via stolen access keys to AWS CLI.

Privilege Escalation

Persistence



Demo

AWS IAM – Defend – Detection Challenges

- Multiple paths of identity impersonation
- Evaluation of Customer managed and inline policies at runtime is complex.
  - Multiple conditions
  - Permission boundaries
- Lack of telemetry for Pass Role actions and need to do manual correlation to gather context for certain APIs.
- Enabling and Ingesting IAM Access Analyzer findings are additional steps and not set up by default per region.
- Open-source tools exists to identify identities with risky permissions, but this context is not available readily to correlate within CloudTrail events in SIEM.

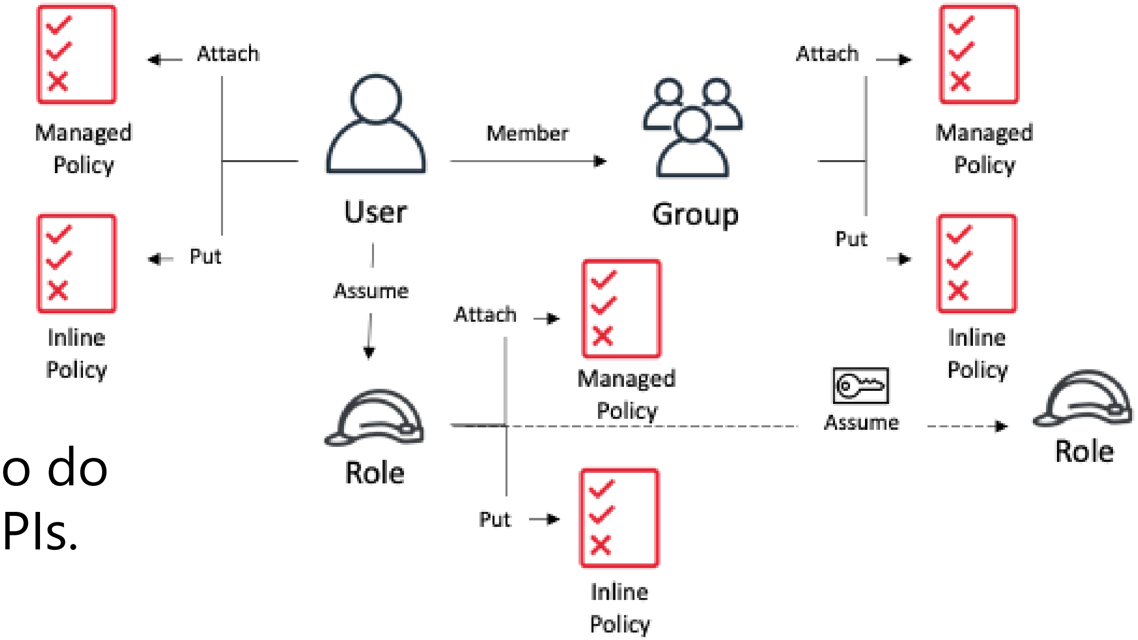


Image Source: AirIAM - Open Source Tool

AWS IAM
Defensive Guidance - Takeaway

AWS IAM – Defensive Guidance I

Detections to Alert/Investigate on:

- Creation of High Privileged Policies (managed and inline) via various APIs
- Resource creation/start-up attached to privileged Policies and associated identities - User, Role, Group, Instance Profile.

Hunting suspicious behavioral patterns for privilege escalation:

- Tracking chaining of multiple AssumeRole events by same identity.
- Tracking unusual AssumeRole events - UserIdentity to Role combination.
- Unusual Role with Instance Profile usage by Users.
- Unusual add , remove operations on the Instance Profiles.

AWS IAM – Defensive Guidance II

Policy Scope

Managed Policy: Looks for ARN with ***Admin***

Inline Policy:

- Parse PolicyDocument in API Calls.
- Look for overly permissive permissions
    - ❑ All actions on all resources (*:*),
    - ❑ AssumeRole for all roles,
    - ❑ All IAM actions for all resources
    - ❑ IAM Passrole action for all resources
    - ❑ KMS, Secret manager actions for all resources

Policy Relationships

Policy attached to Role, User, Group

Track lifecycle of privileged identities :

- Assume role operations of priv. roles.
- Membership changes of priv. groups
- Privileged role attached to Instance Profiles
- Passing of Privileged roles to services

Policy in Action

Monitor for Service API actions from Privileged identities
e.g.

- EC2 instance creation attached to Instance Profile with privileged roles.
- Notebook instance creation passed with privileged role
- Invoke Lambda function with privileged roles

References

Shoutout to existing research :

- CyberArc
- RhinosecurityLabs
- Bishopfox
- Appsecco
- Kloudle

More resource details in **research notes** on GitHub

https://aka.ms/SBTS22-Keynote-Resources

5

Free Resources

Jupyter Notebooks for end-to-end simulation

Security Datasets of individual techniques

Research notes/ Recommended reading

https://aka.ms/SBTS22-Keynote-Resources

Thank you!

https://aka.ms/SBTS22-Keynote-Resources

@Cyb3rWard0g

@ashwinpatil