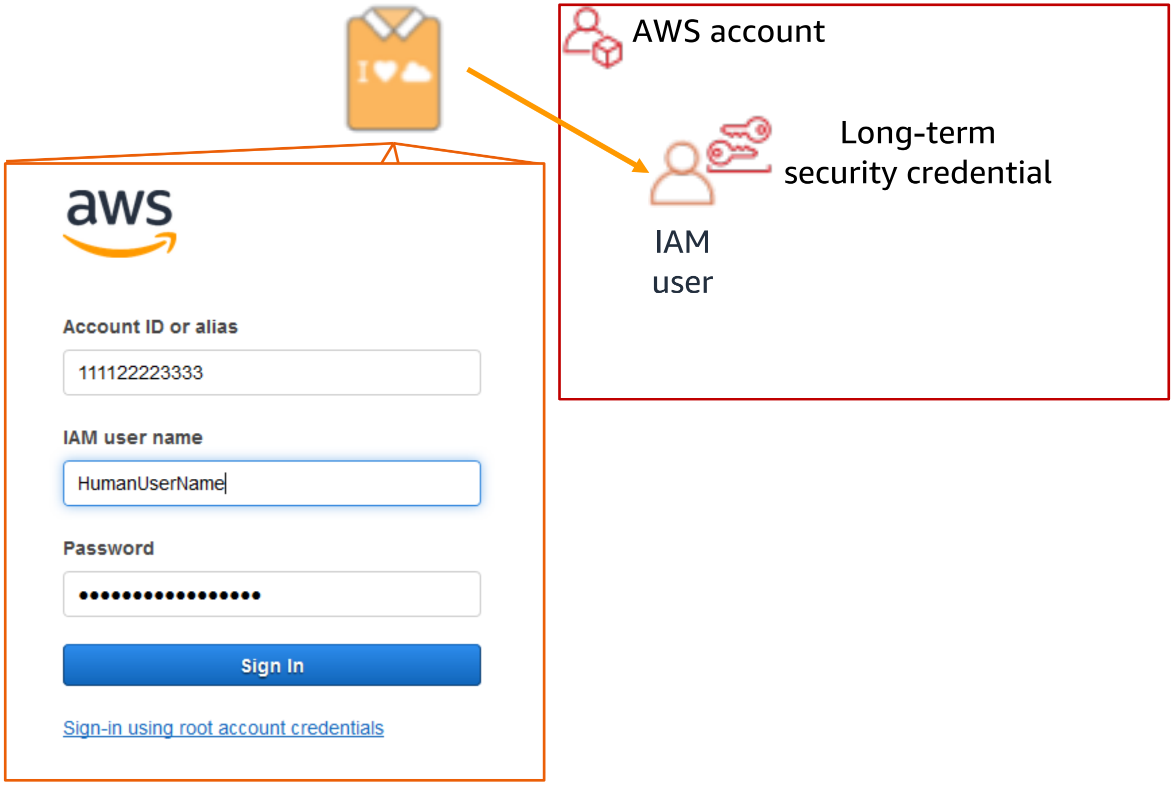
AWS IAM

aws RE:INFORCE

AWS IAM

- **What it is**
  - I – Authentication: Support for human and application caller identities
  - AM – Authorization: Powerful, flexible permissions language for controlling access to cloud resources
- **Why it matters to you**: Every AWS service uses IAM to authenticate and authorize API calls
- **What builders need to know**
  - How to make authenticated API calls to AWS from IAM identities
  - Basic fluency in IAM policy language
  - Where to find and how to understand service-specific authorization control details

AWS identities for human callers: IAM users



AWS account

IAM user

Long-term security credential

Account ID or alias

111122223333

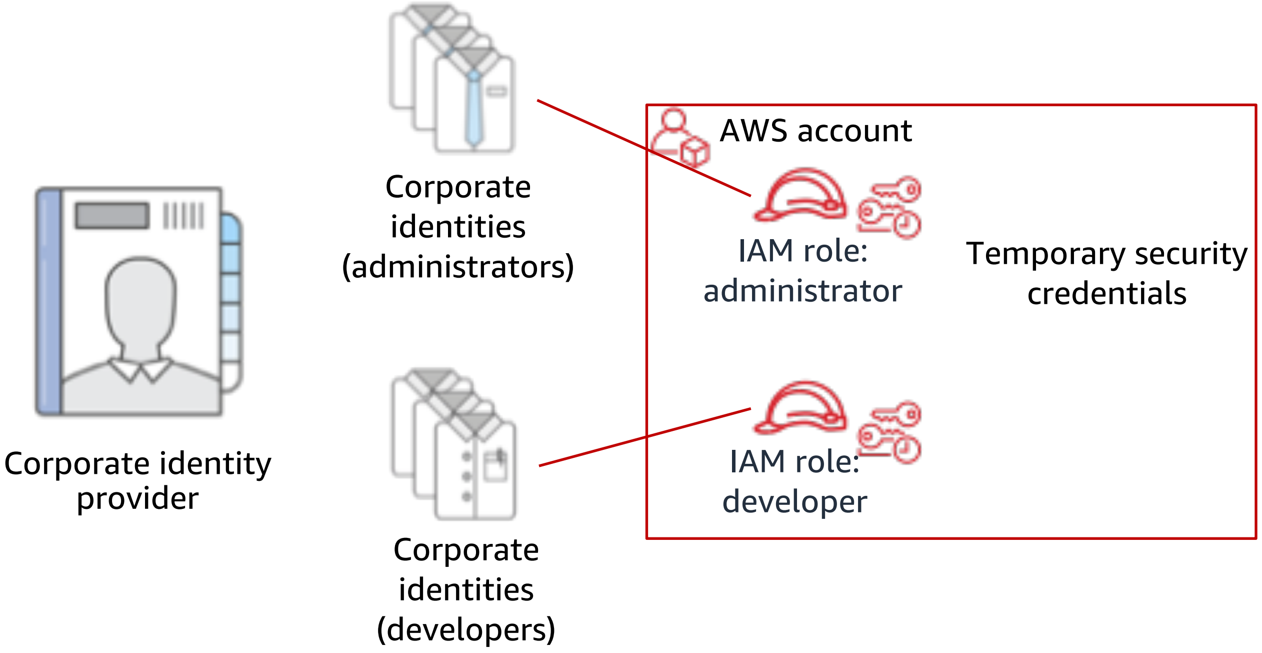
IAM user name

HumanUserName

Password

•••••••••••••••••

AWS identities for human callers: Federated identities


Corporate identity provider
Corporate identities (administrators)
Corporate identities (developers)
AWS account
IAM role: administrator
Temporary security credentials
IAM role: developer

AWS identities for non-human callers

Amazon EC2 instance

AWS Lambda function

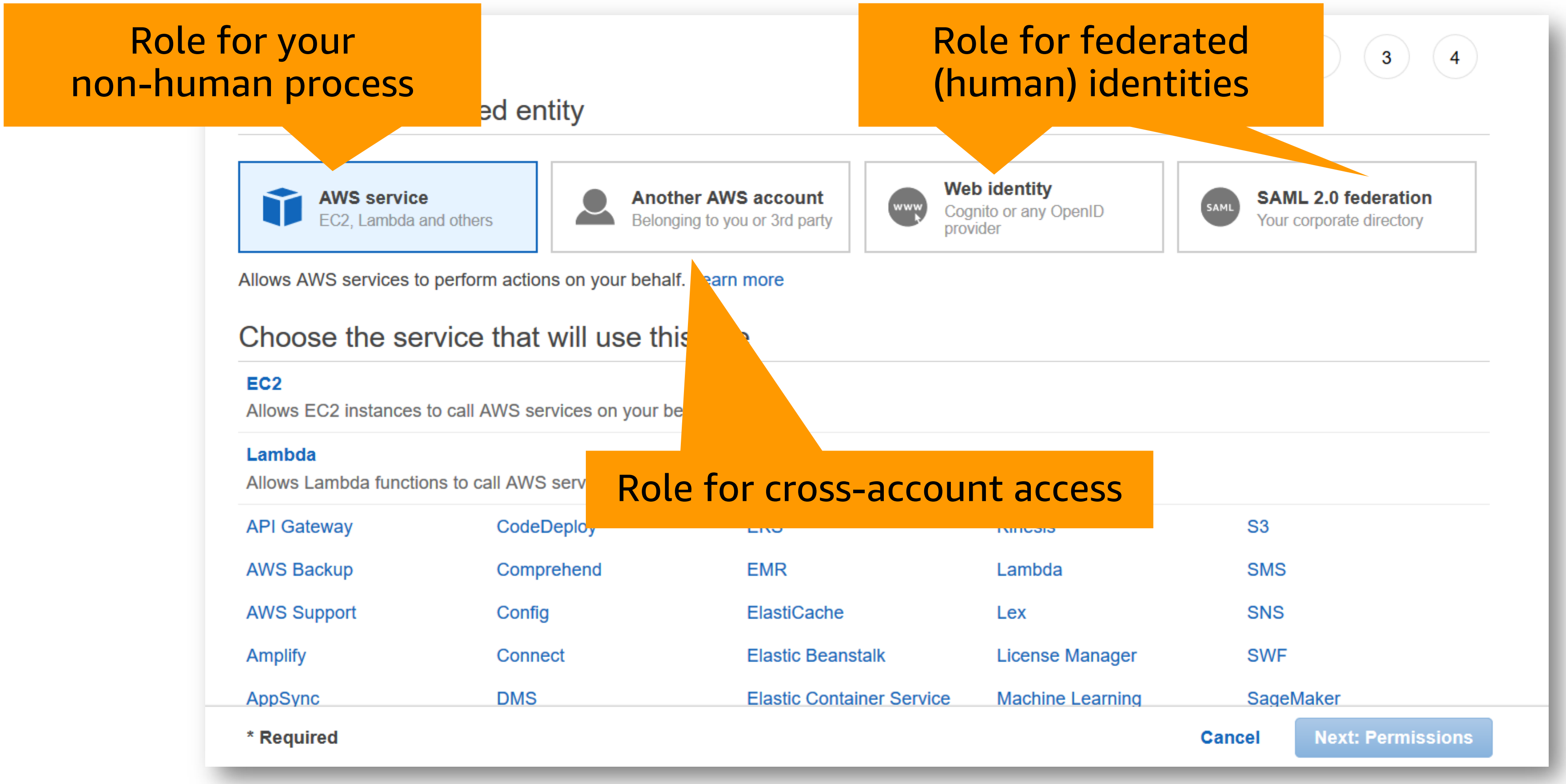
Amazon SageMaker notebook

AWS Glue crawler

Amazon ECS task

…and many others

Creating a role in the AWS Management Console



Role for your non-human process

Role for federated (human) identities

Role for cross-account access

ed entity

AWS service
EC2, Lambda and others

Another AWS account
Belonging to you or 3rd party

Web identity
Cognito or any OpenID provider

SAML 2.0 federation
Your corporate directory

Allows AWS services to perform actions on your behalf. earn more

Choose the service that will use this e

EC2
Allows EC2 instances to call AWS services on your be

Lambda
Allows Lambda functions to call AWS serv

API Gateway
AWS Backup
AWS Support
Amplify
AppSync

CodeDeploy
Comprehend
Config
Connect
DMS

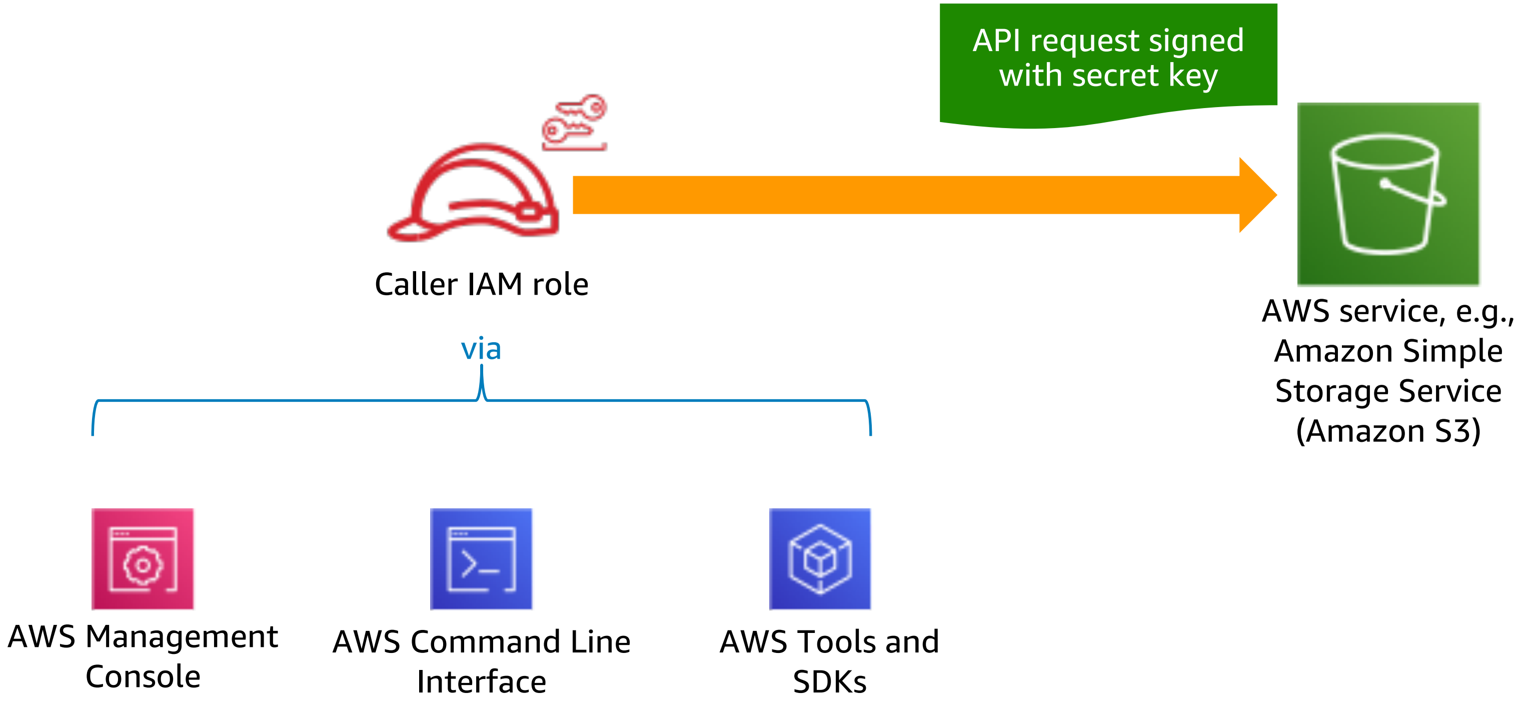
EKS
EMR
ElastiCache
Elastic Beanstalk
Elastic Container Service

Kinesis
Lambda
Lex
License Manager
Machine Learning

S3
SMS
SNS
SWF
SageMaker

* **Required**

How an authentication works in AWS



API request signed with secret key

Caller IAM role

via

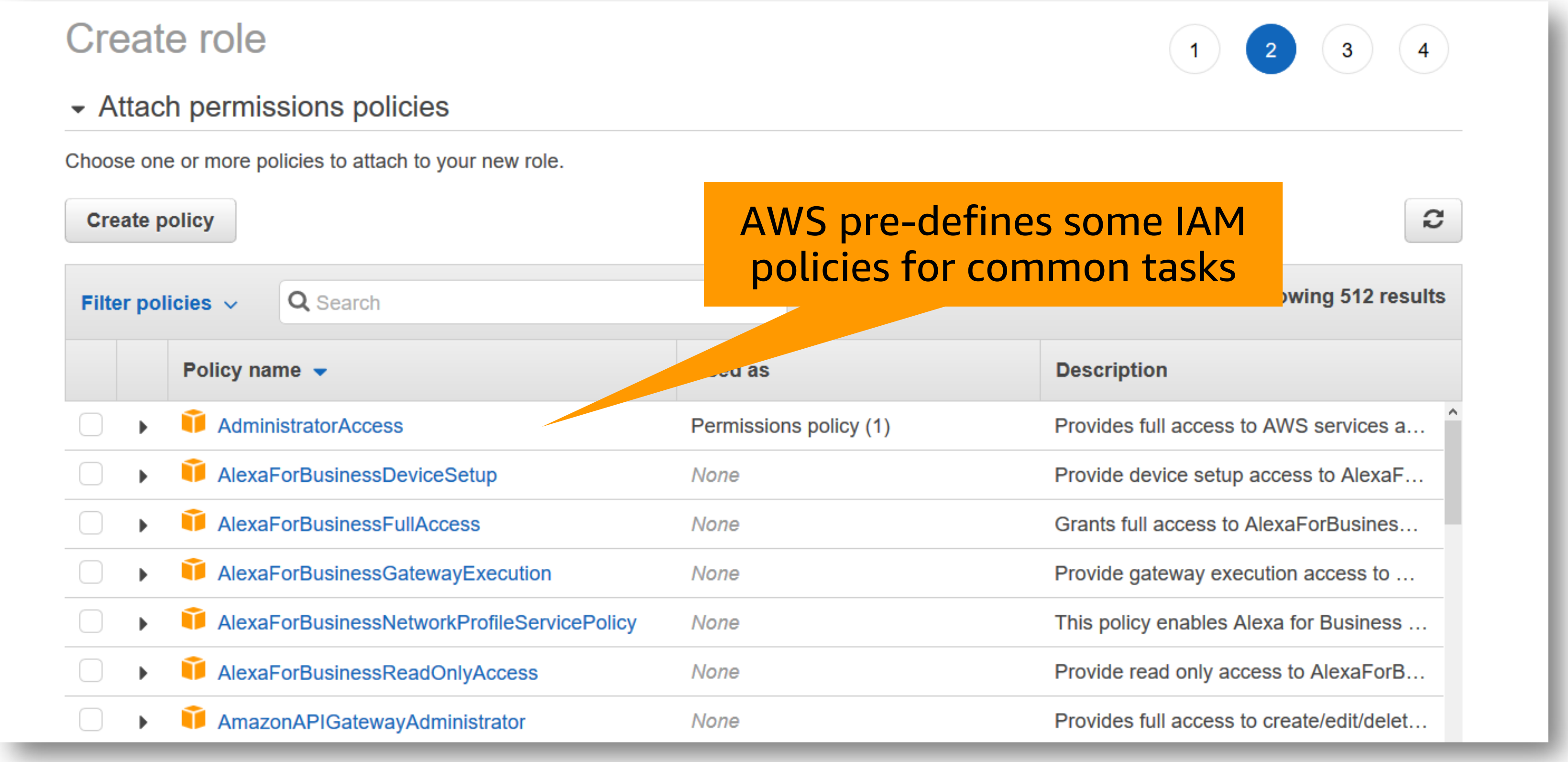
AWS Management Console

AWS Command Line Interface

AWS Tools and SDKs

AWS service, e.g., Amazon Simple Storage Service (Amazon S3)

AWS-managed policies for common sets of permissions



▾ Attach permissions policies

Choose one or more policies to attach to your new role.

owing 512 results

		Policy name ▾	ed as	Description
□	▸	AdministratorAccess	Permissions policy (1)	Provides full access to AWS services a…
□	▸	AlexaForBusinessDeviceSetup	*None*	Provide device setup access to AlexaF…
□	▸	AlexaForBusinessFullAccess	*None*	Grants full access to AlexaForBusines…
□	▸	AlexaForBusinessGatewayExecution	*None*	Provide gateway execution access to …
□	▸	AlexaForBusinessNetworkProfileServicePolicy	*None*	This policy enables Alexa for Business …
□	▸	AlexaForBusinessReadOnlyAccess	*None*	Provide read only access to AlexaForB…
□	▸	AmazonAPIGatewayAdministrator	*None*	Provides full access to create/edit/delet…

AWS pre-defines some IAM policies for common tasks

Reading and writing IAM policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Allow or deny?

What can (or can’t) you do?

What can (or can’t) you do it to?

In English: Allowed to take all Amazon DynamoDB actions

Reading and writing IAM policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:BatchGetItem",
        "dynamodb:GetItem",
        "dynamodb:Query"
      ],
      "Resource": "*"
    }
  ]
}
```

In English: Allowed to take only a few specific Amazon DynamoDB actions

Reading and writing IAM policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:BatchGetItem",
        "dynamodb:GetItem",
        "dynamodb:Query",
      ],
      "Resource": [
        "arn:aws:dynamodb:us-east-2:111122223333:table/MyTableName",
        "arn:aws:dynamodb:us-east-2:111122223333:table/MyTableName/index/*"
      ]
    }
  ]
}
```

In English: Allowed to take specific Amazon DynamoDB actions on a specific table and its indexes

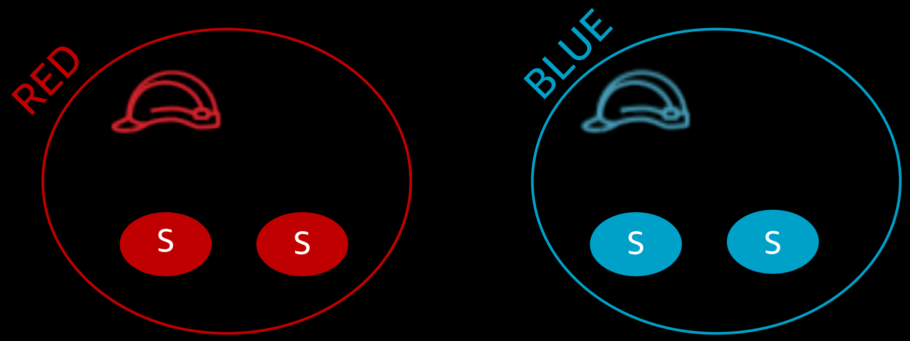
This is an Amazon Resource Name (ARN);
All AWS services use them, and they follow this format

Reading and writing IAM policy

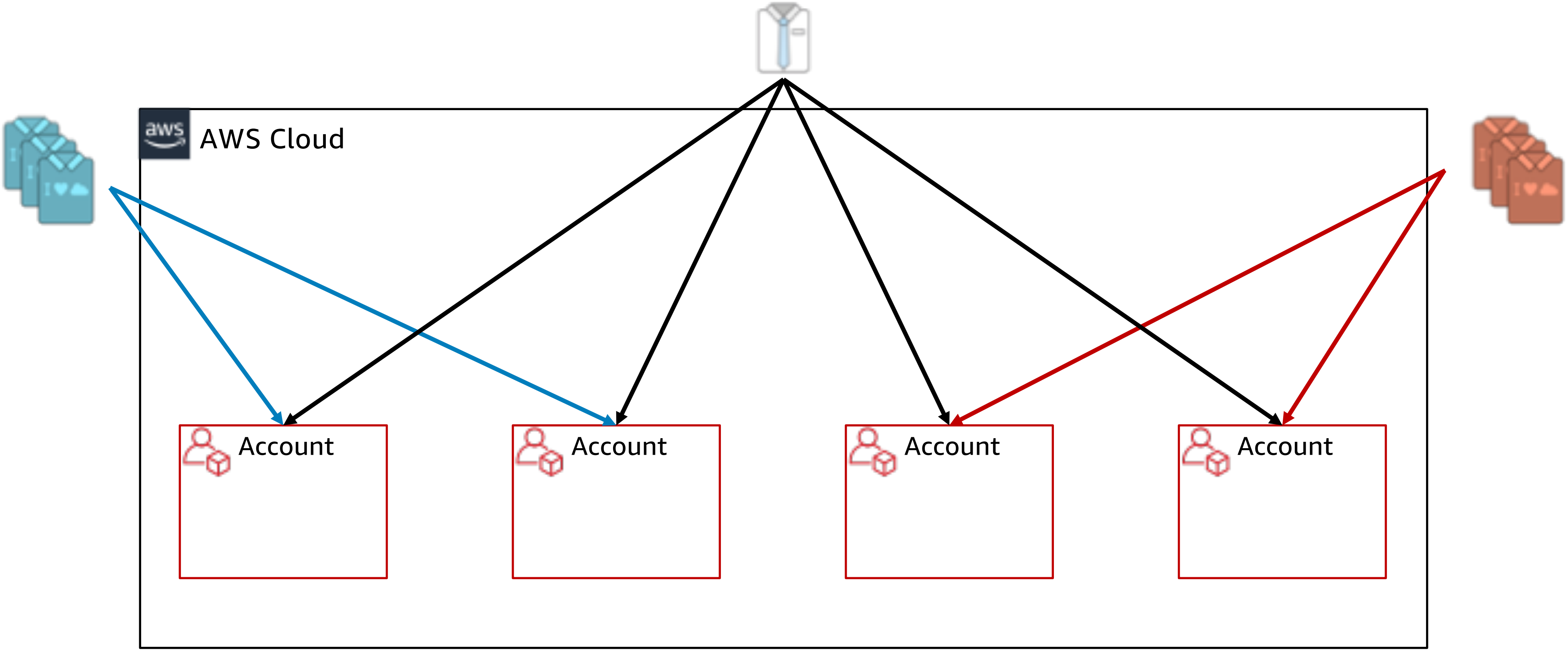
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "secretsmanager:ResourceTag/Project": "${aws:PrincipalTag/Project}"
        }
      }
    }
  ]
}
```

In English: You can read secrets whose project tag matches your own

Attribute-based access control (ABAC)

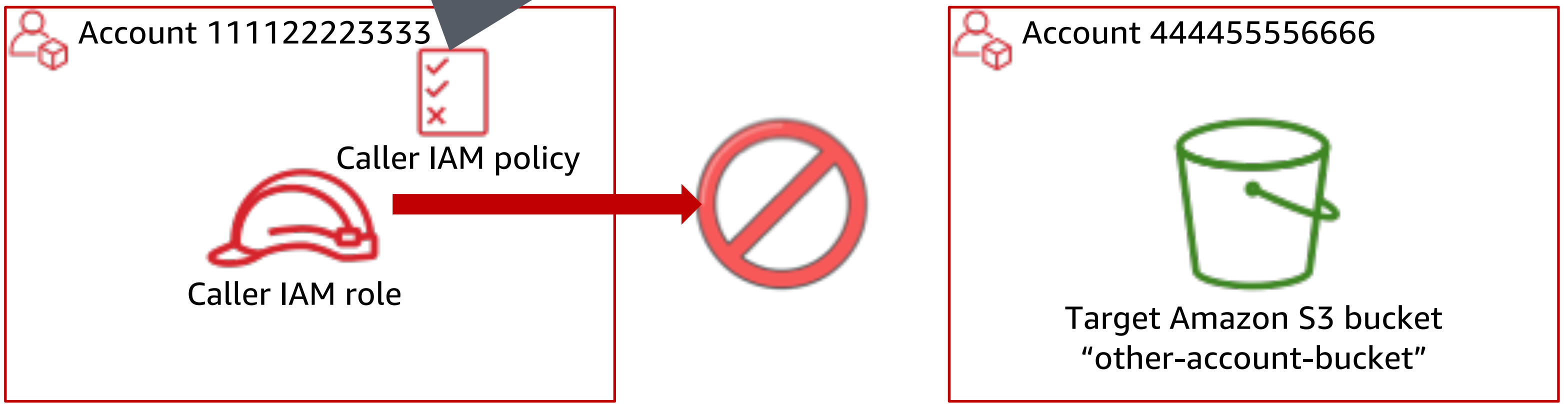


IAM in an AWS enterprise environment



Working across AWS account boundaries

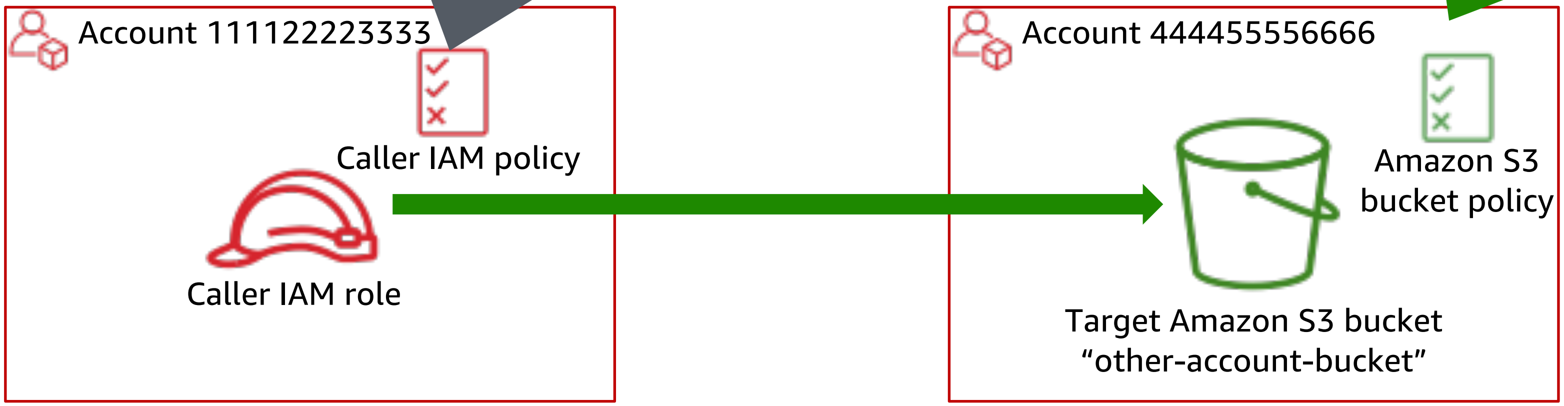
```
{
  “Effect”: “Allow”,
  “Action”: “s3:GetObject”,
  “Resource”: “arn:aws:s3:::other-account-bucket/*”
}
```



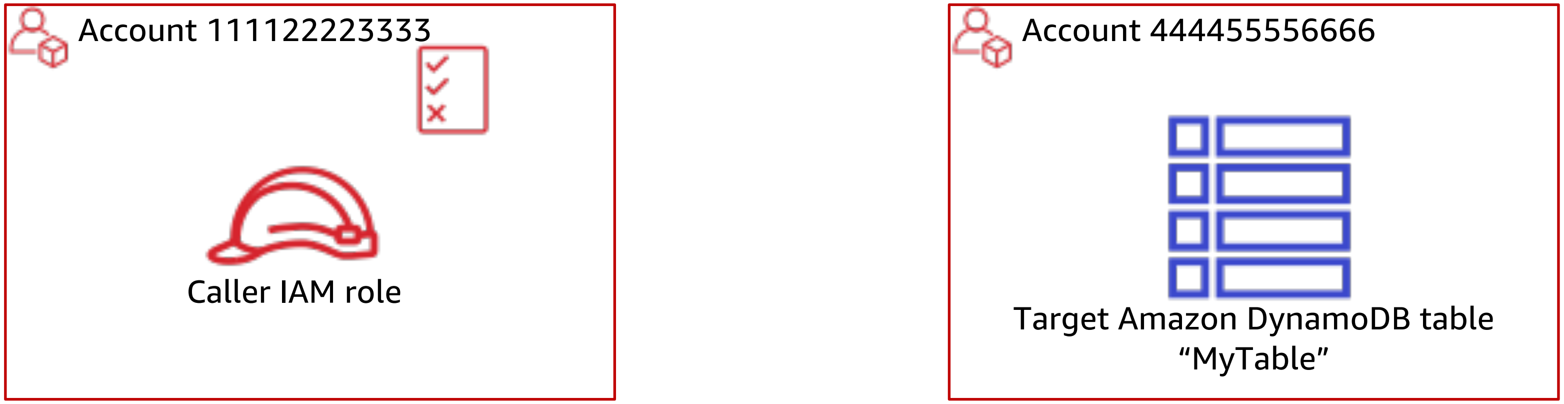
Working across AWS account boundaries

```
{
  “Effect”: “Allow”,
  “Action”: “s3:GetObject”,
  “Resource”: “arn:aws:s3:::other-account-bucket/*”
}
```

```
{
  “Effect”: “Allow”,
  “Principal”: {
     “AWS”: “arn:aws:iam::111122223333:root”
  }
  “Action”: “s3:GetObject”,
  “Resource”: “arn:aws:s3:::other-account-bucket/*”
}
```



Working across AWS account boundaries



Account 111122223333

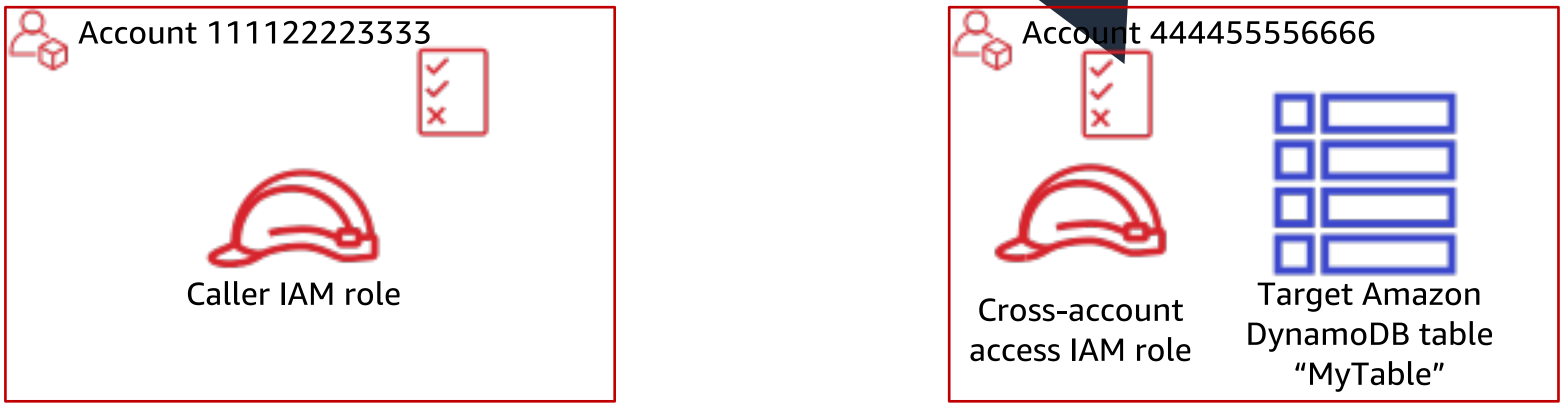
Caller IAM role

Account 444455556666

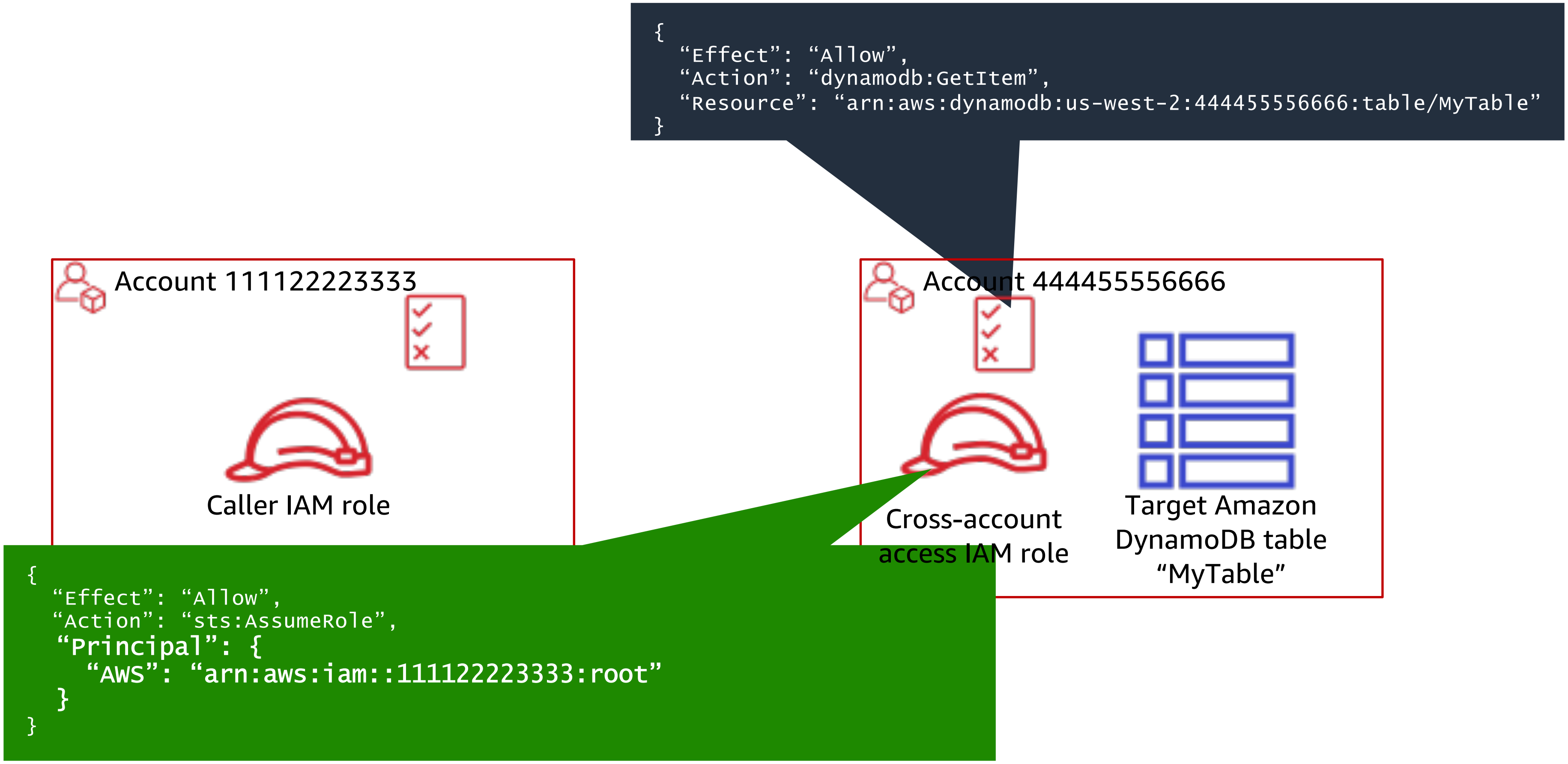
Target Amazon DynamoDB table “MyTable”

Working across AWS account boundaries

```
{
  “Effect”: “Allow”,
  “Action”: “dynamodb:GetItem”,
  “Resource”: “arn:aws:dynamodb:us-west-2:444455556666:table/MyTable”
}
```


Account 111122223333
Caller IAM role
Account 444455556666
Cross-account access IAM role
Target Amazon DynamoDB table “MyTable”

Working across AWS account boundaries

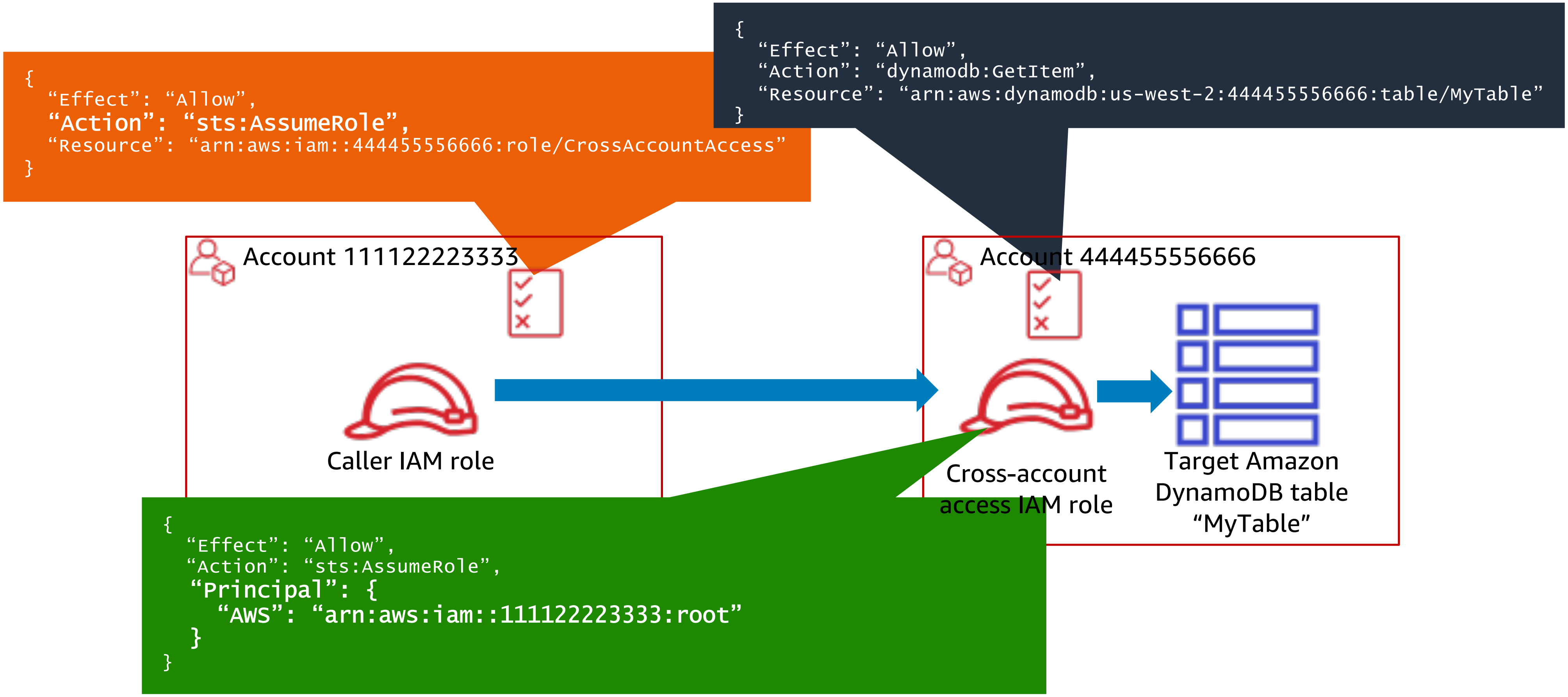


```
{
  “Effect”: “Allow”,
  “Action”: “dynamodb:GetItem”,
  “Resource”: “arn:aws:dynamodb:us-west-2:444455556666:table/MyTable”
}
```

Account 111122223333
Caller IAM role
Account 444455556666
Cross-account access IAM role
Target Amazon DynamoDB table “MyTable”

```
{
  “Effect”: “Allow”,
  “Action”: “sts:AssumeRole”,
  “Principal”: {
    “AWS”: “arn:aws:iam::111122223333:root”
  }
}
```

Working across AWS account boundaries



```
{
  “Effect”: “Allow”,
  “Action”: “sts:AssumeRole”,
  “Resource”: “arn:aws:iam::444455556666:role/CrossAccountAccess”
}
```

```
{
  “Effect”: “Allow”,
  “Action”: “dynamodb:GetItem”,
  “Resource”: “arn:aws:dynamodb:us-west-2:444455556666:table/MyTable”
}
```

```
{
  “Effect”: “Allow”,
  “Action”: “sts:AssumeRole”,
  “Principal”: {
    “AWS”: “arn:aws:iam::111122223333:root”
  }
}
```

Account 111122223333
Caller IAM role
Account 444455556666
Cross-account access IAM role
Target Amazon DynamoDB table “MyTable”

Thank you!

aws RE:INFORCE