



Access Control Verification for Everyone

Andrew Gacek

Automated Reasoning in Identity, Amazon Web Services

September 22, 2020

Access Analyzer

Monitor access to resources

How it works

[Create analyzer](#)

Getting started [↗](#)

- [What is Access Analyzer?](#)
- [Access Analyzer User Guide](#)



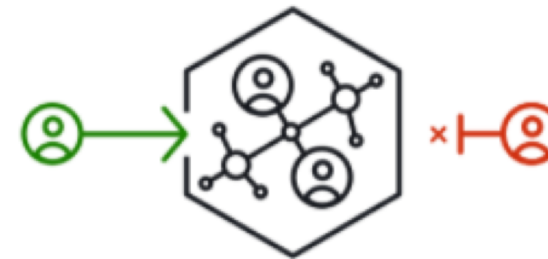
1 Create an analyzer

You can set the scope for the analyzer to an organization or an AWS account. This is your zone of trust. The analyzer scans all of the supported resources within your zone of trust.



2 Review active findings

When Access Analyzer finds a policy that allows access to a resource from outside of your zone of trust, it generates an active finding. Findings include details about the access so that you can take action.

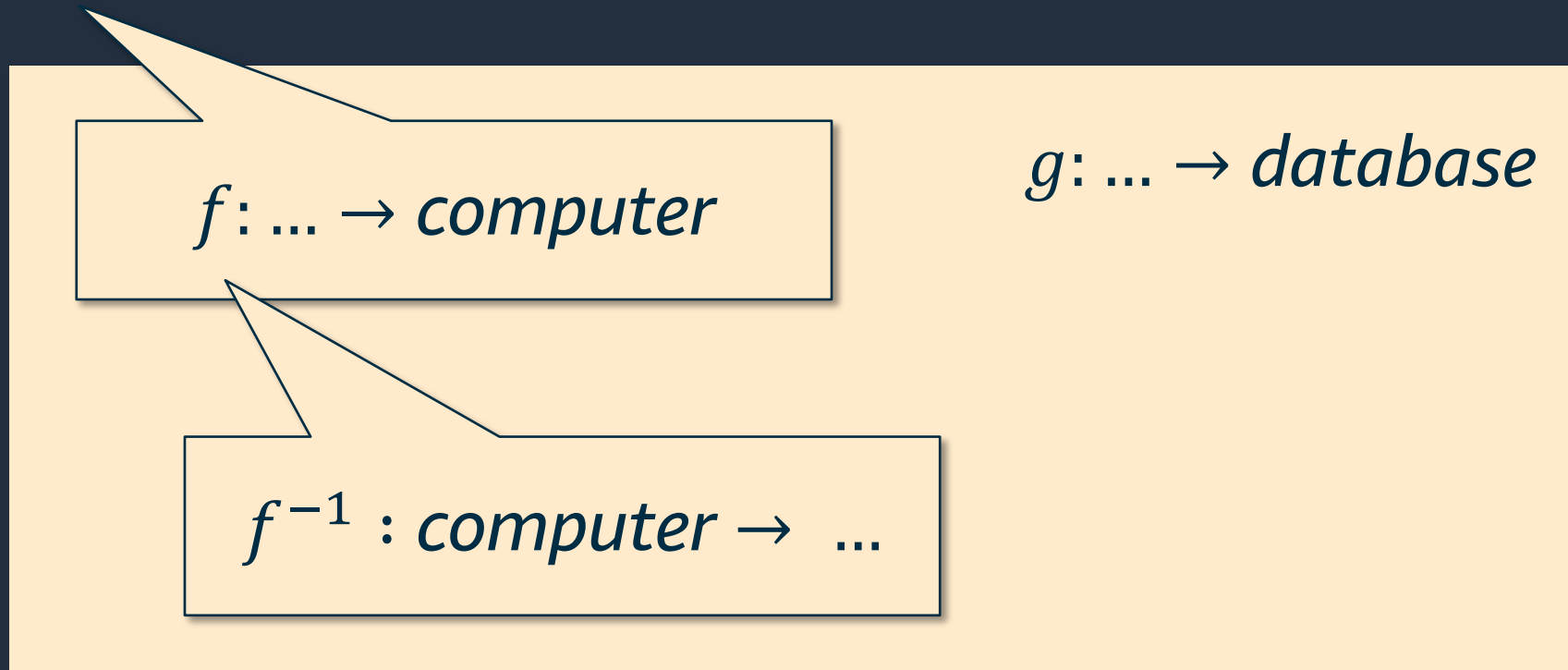


3 Take action

If the access is intended, you can archive the finding so that you can focus on reviewing active findings. If the access is not intended, you can resolve the finding by modifying the policy to remove access to the resource.

What is cloud computing?

“on-demand delivery of IT resources via the Internet with pay-as-you-go pricing.”



Amazon Web Services



Identity & Access Management (IAM) Policy



Virtual Private Cloud (VPC)



Simple Storage Service (S3)

Example Policy

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "111122223333"
  },
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::my-bucket/*"
},
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "*",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "StringNotEquals": {
      "aws:SourceVpc": "vpc-abc"
    }
  }
}
```

What is (automated) reasoning?

Using the rules of a system to logically infer its possible behaviors

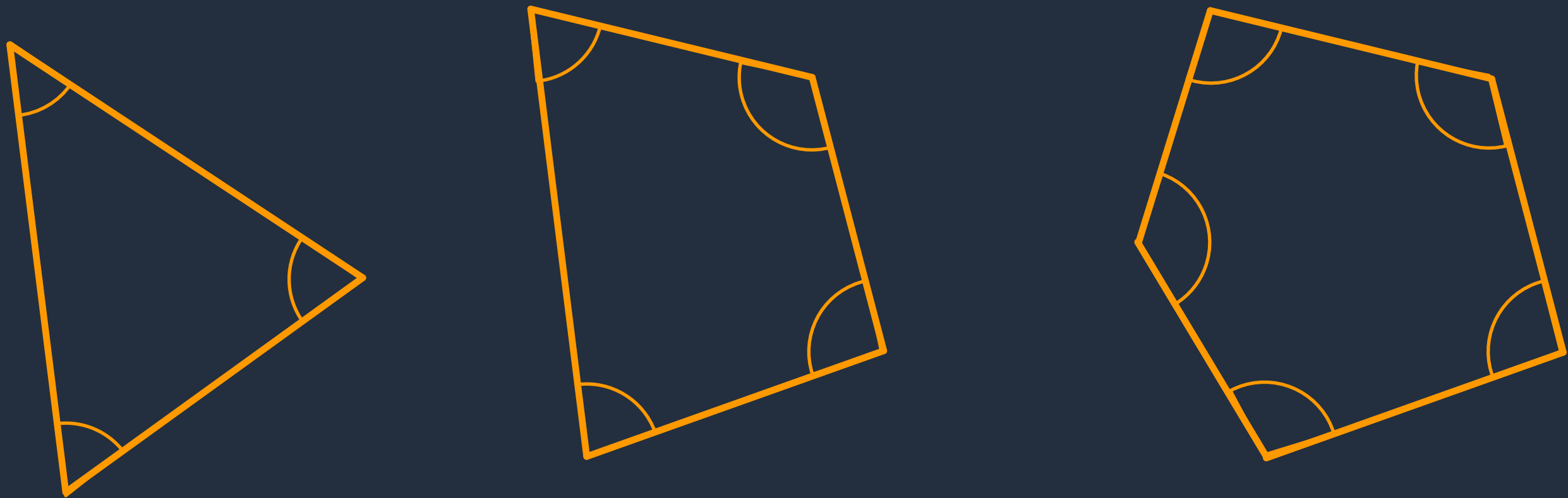
We must *know* the rules

Rules for

- Identities
- Resources
- Applications
- Accounts
- Organizations

Logic tells us how the rules interact

Sum of interior angles of a polygon



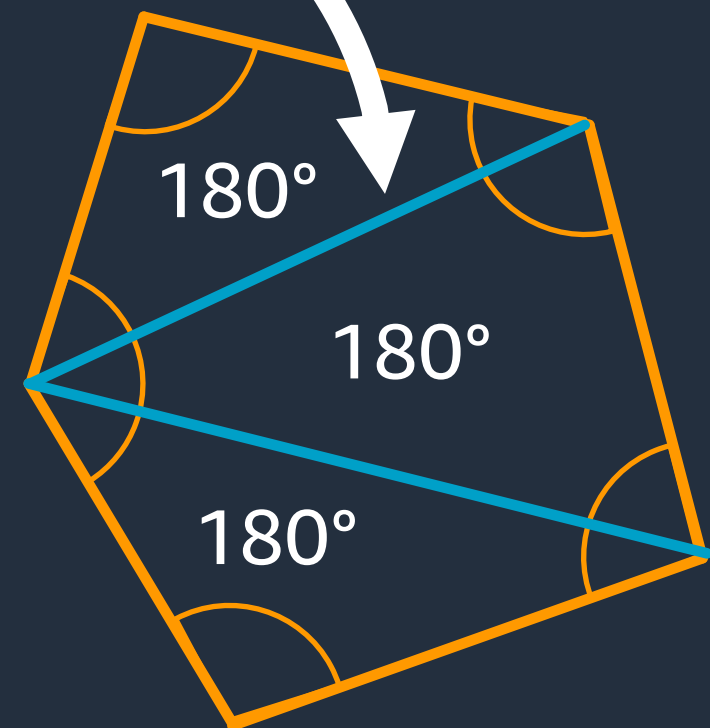
Sum of interior angles of a polygon: rules only

n-sided polygon $\rightarrow 180^\circ(n - 2)$



Sum of interior angles of a polygon: rules + logic

Triangle $\rightarrow 180^\circ$



“Is my-bucket accessible outside of vpc-abc?”

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "111122223333"
  },
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::my-bucket/*"
},
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "*",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "StringNotEquals": {
      "aws:SourceVpc": "vpc-abc"
    }
  }
}
```

“Is my-bucket accessible outside of vpc-abc?”

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "111122223333"
  },
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::my-bucket/*"
},
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "*",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "StringNotEquals": {
      "aws:SourceVpc": "vpc-abc"
    }
  }
}
```

```
Request
Principal: 111122223333:user/Jane
Action:    s3:GetObject
Resource:  my-bucket/my-file
Condition:
  aws:SourceIp:      192.0.2.3
  aws:CurrentTime:   2019-12-05T12:34:56Z
  aws:MultiFactorAuthAge: 1234
  aws:MultiFactorAuthPresent: true
  aws:PrincipalAccount: 111122223333
```

“Is my-bucket accessible outside of vpc-abc?”

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "111122223333"
  },
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::my-bucket/*"
},
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "*",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "StringNotEquals": {
      "aws:SourceVpc": "vpc-abc"
    }
  }
}
```

“Is my-bucket accessible outside of vpc-abc?”

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "111122223333"
  },
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::my-bucket/*"
},
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "*",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "StringNotEquals": {
      "aws:SourceVpc": "vpc-abc"
    }
  }
}
```

Request
Principal:
Action:
Resource:
Condition:

“Is my-bucket accessible outside of vpc-abc?”

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "111122223333"
  },
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::my-bucket/*"
},
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "*",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "StringNotEquals": {
      "aws:SourceVpc": "vpc-abc"
    }
  }
}
```

Request
Principal: 111122223333
Action:
Resource:
Condition:

Request
Principal: (not 111122223333)
Action:
Resource:
Condition:

“Is my-bucket accessible outside of vpc-abc?”

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "111122223333"
  },
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::my-bucket/*"
},
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "*",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "StringNotEquals": {
      "aws:SourceVpc": "vpc-abc"
    }
  }
}
```

Request
Principal: 111122223333
Action:
Resource:
Condition:

Request
Principal: (not 111122223333)
Action:
Resource:
Condition:

“Is my-bucket accessible outside of vpc-abc?”

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "111122223333"
  },
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::my-bucket/*"
},
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "*",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "StringNotEquals": {
      "aws:SourceVpc": "vpc-abc"
    }
  }
}
```

Request

Principal: 111122223333
Action: s3:GetObject
Resource:
Condition:

Request

Principal: 111122223333
Action: (not s3:GetObject)
Resource:
Condition:

Request

Principal: (not 111122223333)
Action:
Resource:
Condition:

“Is my-bucket accessible outside of vpc-abc?”

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "111122223333"
  },
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::my-bucket/*"
},
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "*",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "StringNotEquals": {
      "aws:SourceVpc": "vpc-abc"
    }
  }
}
```

Request
Principal: 111122223333
Action: s3:GetObject
Resource:
Condition:

Request
Principal: 111122223333
Action: (not s3:GetObject)
Resource:
Condition:

Request
Principal: (not 111122223333)
Action:
Resource:
Condition:

“Is my-bucket accessible outside of vpc-abc?”

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "111122223333"
  },
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::my-bucket/*"
},
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "*",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "StringNotEquals": {
      "aws:SourceVpc": "vpc-abc"
    }
  }
}
```



“Is my-bucket accessible outside of vpc-abc?”

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "111122223333"
  },
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::my-bucket/*"
},
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "*",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "StringNotEquals": {
      "aws:SourceVpc": "vpc-abc"
    }
  }
}
```



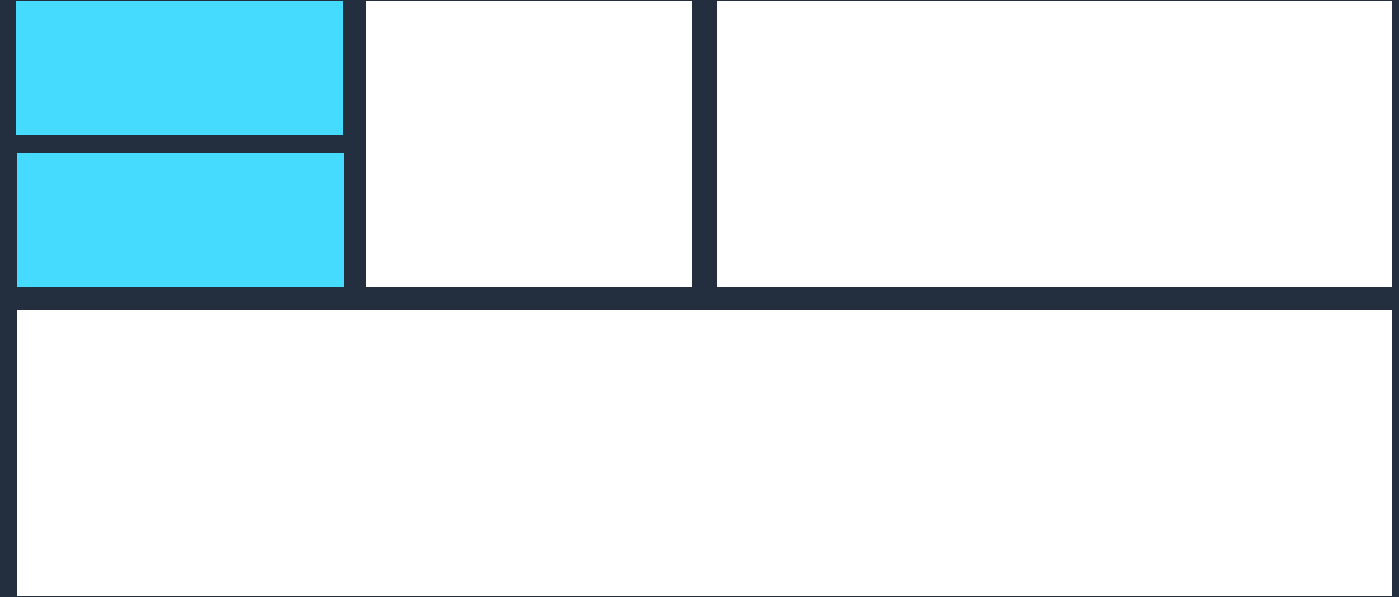
“Is my-bucket accessible outside of vpc-abc?”

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "111122223333"
  },
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::my-bucket/*"
},
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "*",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "StringNotEquals": {
      "aws:SourceVpc": "vpc-abc"
    }
  }
}
```



“Is my-bucket accessible outside of vpc-abc?”

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "111122223333"
  },
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::my-bucket/*"
},
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "*",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "StringNotEquals": {
      "aws:SourceVpc": "vpc-abc"
    }
  }
}
```



“Is my-bucket accessible outside of vpc-abc?”

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "111122223333"
  },
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::my-bucket/*"
},
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "*",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "StringNotEquals": {
      "aws:SourceVpc": "vpc-abc"
    }
  }
}
```

No

These requests are
from vpc-abc

These requests are
not allowed

AWS Security Blog

How AWS uses automated reasoning to help you achieve security at scale

by Andrew Gacek | on 20 JUN 2018 | in [Security, Identity, & Compliance](#) | [Permalink](#) | [Comments](#) | [Share](#)

At AWS, we focus on achieving security at scale to diminish risks to your business. Fundamental to this approach is ensuring your policies are configured in a way that helps protect your data, and the Automated Reasoning Group (ARG), an advanced innovation team at AWS, is using [automated reasoning](#) to do it.

What is automated reasoning, you ask? It's a method of formal verification that automatically generates and checks mathematical proofs which help to prove the correctness of systems; that is, fancy math that proves things are working as expected. If you want a deeper understanding of automated reasoning, check out this [re:Invent session](#). While the applications of this methodology are vast, in this post I'll explore one specific aspect: analyzing policies using an internal Amazon service named Zelkova.

What is Zelkova? How will it help me?

Zelkova uses automated reasoning to analyze policies and the future consequences of policies. This includes [AWS Identity and Access Management](#) (IAM) policies, [Amazon Simple Storage Service](#) (S3) policies, and other resource policies. These policies dictate *who* can (or can't) do *what* to *which* resources. Because Zelkova uses *automated* reasoning, you no longer need to think about what questions you need to ask about your policies. Using fancy math, as mentioned above, Zelkova will automatically derive the questions *and* answers you need to be asking about your policies, improving confidence in your security configuration(s).

Semantic-based Automated Reasoning for AWS Access Policies using SMT

John Backes, Pauline Bolignano, Byron Cook, Catherine Dodge, Andrew Gacek, Kasper Luckow, Neha Rungta, Oksana Tkachuk, Carsten Varming
Amazon Web Services

Abstract—Cloud computing provides on-demand access to IT resources via the Internet. Permissions for these resources are defined by expressive access control policies. This paper presents a formalization of the Amazon Web Services (AWS) policy language and a corresponding analysis tool, called ZELKOVA, for verifying policy properties. ZELKOVA encodes the semantics of policies into SMT, compares behaviors, and verifies properties. It provides users a sound mechanism to detect misconfigurations of their policies. ZELKOVA solves a PSPACE-complete problem and is invoked many millions of times daily.

I. INTRODUCTION

Cloud computing provides on-demand access to IT resources via the Internet. The convenience of accessing resources in the cloud is made secure by user-specified *access control policies*. An access control policy is an expressive specification of what resources can be accessed, by whom, and under what conditions. Properly configured policies are an important part of an organization's security posture. The

In this paper, we present the development and application of ZELKOVA, a policy analysis tool designed to reason about the semantics of AWS access control policies. ZELKOVA translates policies and properties into Satisfiability Modulo Theories (SMT) formulas and uses SMT solvers to check the validity of the properties. We use off-the-shelf solvers and an in-house extension of Z3 called Z3AUTOMATA.

ZELKOVA reasons about all possible permissions allowed by a policy in order to verify properties. For example, ZELKOVA can answer the questions “Is this resource accessible by a particular user?” and “Can an arbitrary user write to this resource?”. The property to be verified is specified in the policy language itself, eliminating the need for a different specification or formalism for properties. In addition, ZELKOVA provides many built-in checks for common properties.

The SMT encoding uses the theory of strings, regular expressions, bit vectors, and integer comparisons. The use of the wildcards *** (any number of characters) and *?* (exactly one


Zelkova Demo

Zelkova demo: “Is my-bucket accessible outside of vpc-abc?”

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "11112222333"
  },
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::my-bucket/*"
},
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "*",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "StringNotEquals": {
      "aws:SourceVpc": "vpc-abc"
    }
  }
}
```

?

AND



```
{
  "Effect": "Allow",
  "Principal": "*",
  "Action": "*",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "StringEquals": {
      "aws:SourceVpc": "vpc-abc"
    }
  }
}
```

Zelkova demo: “Is my-bucket accessible outside of vpc-abc?”

```
>> aws zelkova compare-policies --items file://items.json
```


```
{
  "Items": [
    {
      "Comparison": "LESS_PERMISSIVE"
    }
  ]
}
```



Access Control Verification (for Everyone)

Zelkova demo: “Is my-bucket accessible outside of vpc-abc?”

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "11112222333"
  },
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::my-bucket/*"
},
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "*",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "StringNotEquals": {
      "aws:SourceVpc": "vpc-abc"
    }
  }
}
```



```
{
  "Effect": "Allow",
  "Principal": "*",
  "Action": "*",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "StringEquals": {
      "aws:SourceVpc": "vpc-abc"
    }
  }
}
```

Zelkova: Asking the right questions

FSV301

How Millennium Management achieves provable security with AWS Zelkova

Byron Cook
Director, Automated Reasoning Group
Amazon Web Services

Mark Horta
Head of Cloud Engineering
Millennium Management

Aaron Fagan
Principal Cloud Security Engineer
Millennium Management

aws SUMMIT

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

SEC 301

The Theory and Math Behind Data Privacy and Security Assurance

Neha Rungta
Principal Software Development Engineer
AWS

Dan Peebles
Senior Software Development Engineer
Bridgewater Associates

Greg Frascadore
Security Architect
Bridgewater Associates

aws re:Invent

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

SEC 320

Policy Verification and Enforcement at Scale with AWS

Kai Huang
Vice President
Goldman Sachs

Sujoy Saha
Software Engineer
Goldman Sachs

Victor Padron-Blanco
Software Engineer
Goldman Sachs

With an introduction by:
Byron Cook
Director, Automated Reasoning at AWS

aws re:Invent

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

SEC 302

How LogMeIn Automates Governance and Empowers Developers at Scale

Cameron Worrell
Solutions Architect
AWS

Brian Galura
Principal Technical Operations Architect
LogMeIn Inc.

aws re:Invent

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

Zelkova: Asking the right questions

AWS Professional Services

Helping you achieve your desired business outcomes with AWS

Zelkova: Asking the right questions

S3 buckets Discover the console

All access types

[+ Create bucket](#) [Edit public access settings](#) [Empty](#) [Delete](#) 4 Buckets 1 Regions

<input type="checkbox"/> Bucket name ▼	Access ▼	Region ▼	Date created ▼
<input type="checkbox"/> reinvent-zelkova-bucket1	Bucket and objects not public	US East (N. Virginia)	Oct 12, 2018 3:24:29 PM GMT-0500
<input type="checkbox"/> reinvent-zelkova-bucket2	Bucket and objects not public	US East (N. Virginia)	Oct 12, 2018 3:24:34 PM GMT-0500
<input type="checkbox"/> reinvent-zelkova-bucket3	Public	US East (N. Virginia)	Oct 12, 2018 3:24:41 PM GMT-0500
<input type="checkbox"/> reinvent-zelkova-bucket4	Bucket and objects not public	US East (N. Virginia)	Oct 12, 2018 3:24:47 PM GMT-0500

Zelkova: Asking the right questions

AWS Config

Record and evaluate configurations of your AWS resources

Get started with AWS Config

s3-bucket-ssl-requests-only

Checks whether S3 buckets have policies that require requests to use Secure Socket Layer (SSL).

S3 . Zelkova

s3-bucket-server-side-encryption-ena...

Checks that your Amazon S3 bucket either has S3 default encryption enabled or that the S3 bucket policy explicitly denies put-object requests without server side encryption.

S3 . Zelkova

s3-bucket-policy-grantee-check

Checks that the access granted by the Amazon S3 bucket is restricted to any of the AWS principals, federated users, service principals, IP addresses, or VPCs that you

S3 . Zelkova

Zelkova: Asking the right questions

AWS Zelkova
User Guide

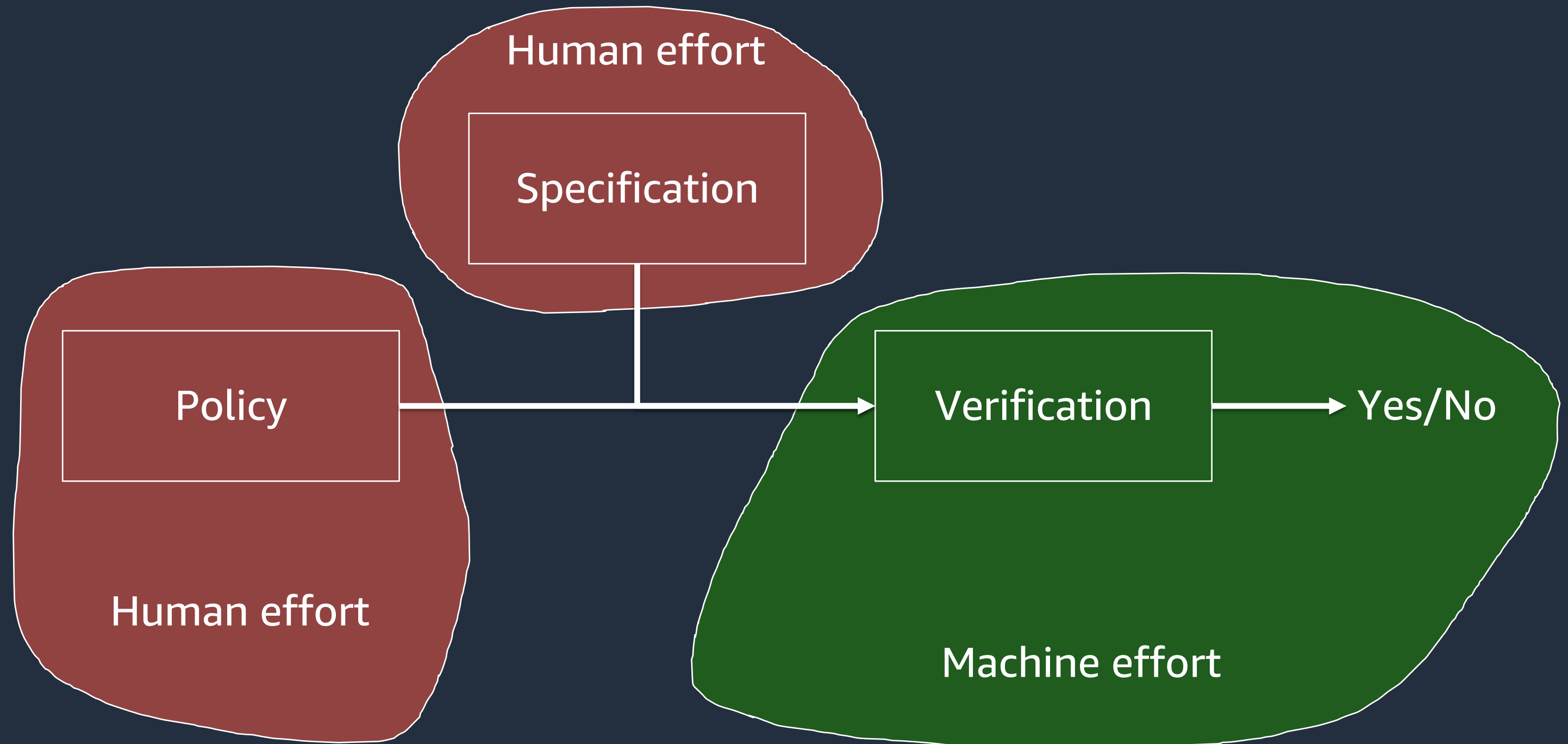


Zelkova
Zelkova API Reference

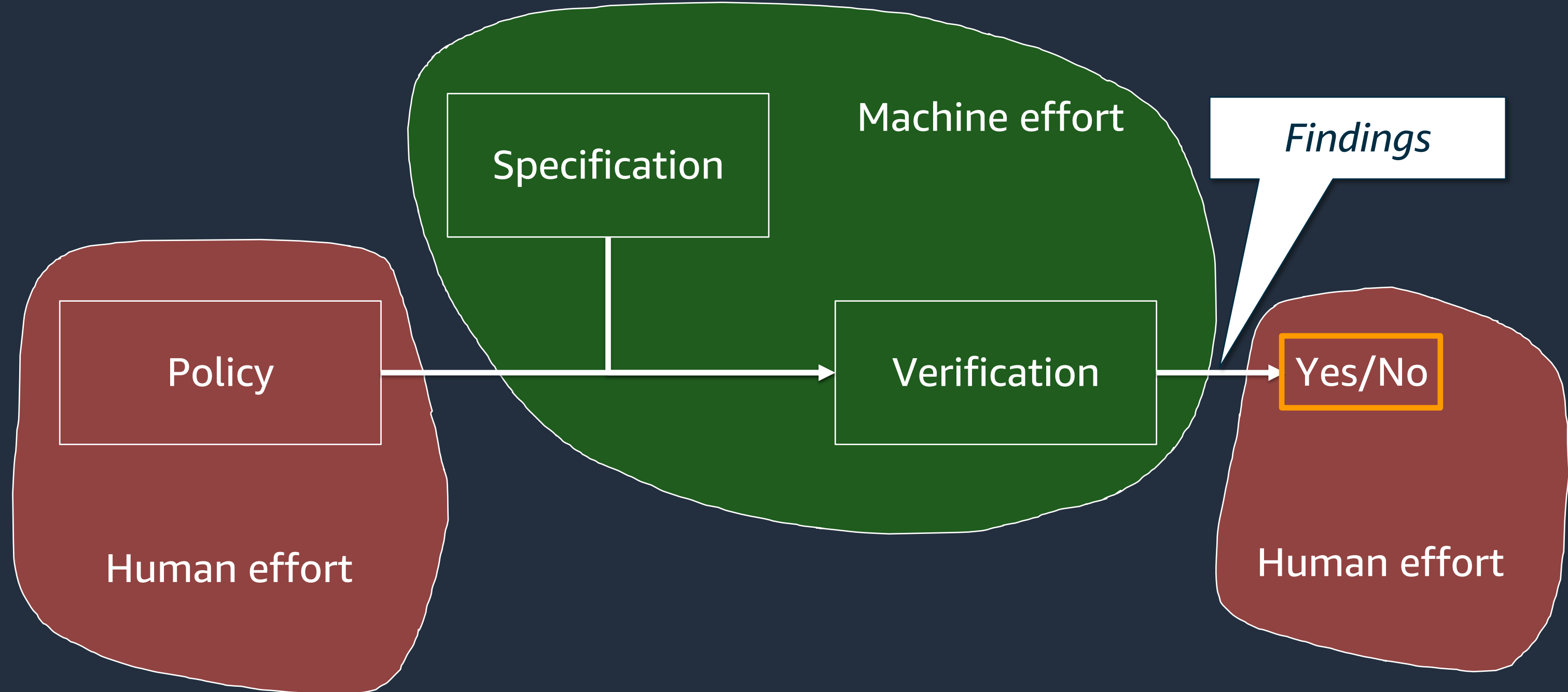
Confidential



Traditional verification approach



Access Analyzer verification approach



Desired properties of findings

Sound – *Every* access is represented by *some* finding

Precise – findings *adhere closely* to the allowed access

Compact – the set of findings is *small*

Structure of findings

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "111122223333"
  },
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::my-bucket/*"
},
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "*",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "StringNotEquals": {
      "aws:SourceVpc": "vpc-abc"
    }
  }
}
```

Principal: 111122223333
Principal: *
aws:SourceVpc: vpc-abc

has

Action: s3:GetObject
Action: *

access to

Resource: my-bucket

Structure of findings

Does 111122223333 have * access to my-bucket?

```
Principal: 111122223333
Principal: *
aws:SourceVpc: vpc-abc
```

has

Does * have GetObject access to my-bucket?

```
Action: s3:GetObject
Action: *
```

Does * have * access to my-bucket?

access to

Does 111122223333 with vpc-abc have GetObject
access to my-bucket?

```
Resource: my-bucket
```

Structure of findings

Does 111122223333 have * access to my-bucket?

Does * have GetObject access to my-bucket?

Does * have * access to my-bucket?

Does 111122223333 with vpc-abc have GetObject access to my-bucket?

```
Principal: 111122223333  
Principal: *  
aws:SourceVpc: vpc-abc
```

has

```
Action: s3:GetObject  
Action: *
```

access to

```
Resource: my-bucket
```

Example finding

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "111122223333"
  },
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::my-bucket/*"
},
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "*",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "StringNotEquals": {
      "aws:SourceVpc": "vpc-abc"
    }
  }
}
```

Finding

Principal: 111122223333

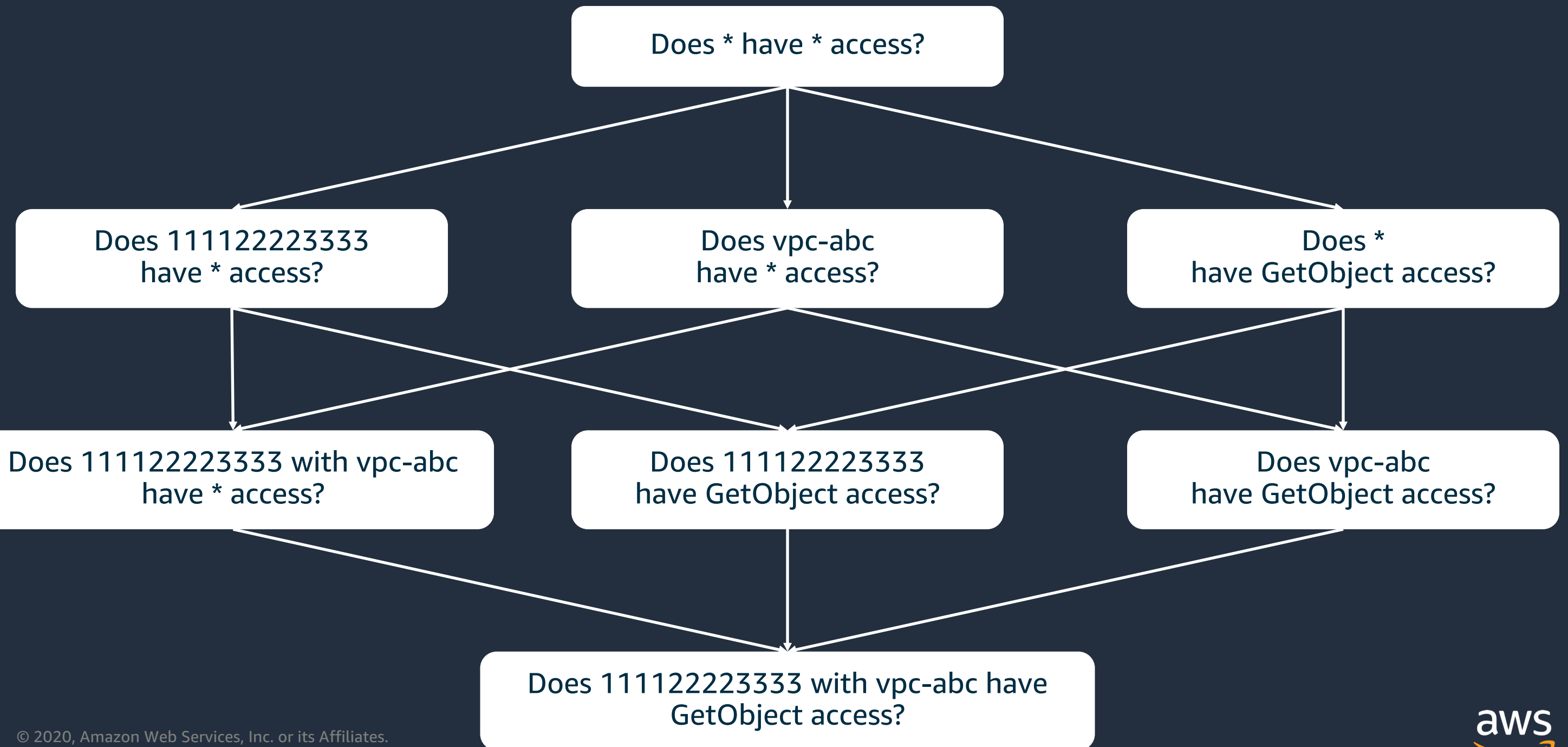
Action: s3:GetObject

Resource: my-bucket

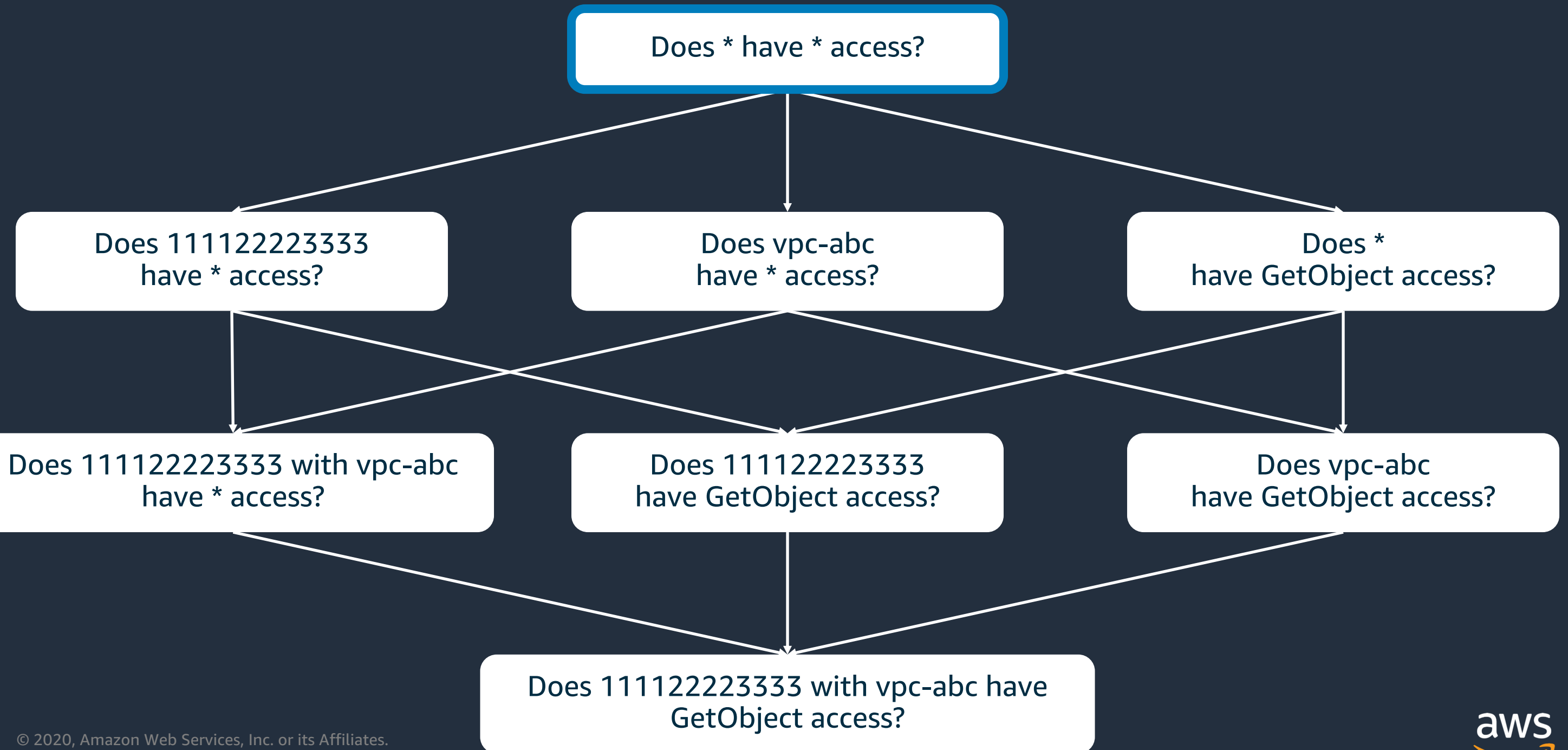
Condition:

aws:SourceVpc: vpc-abc

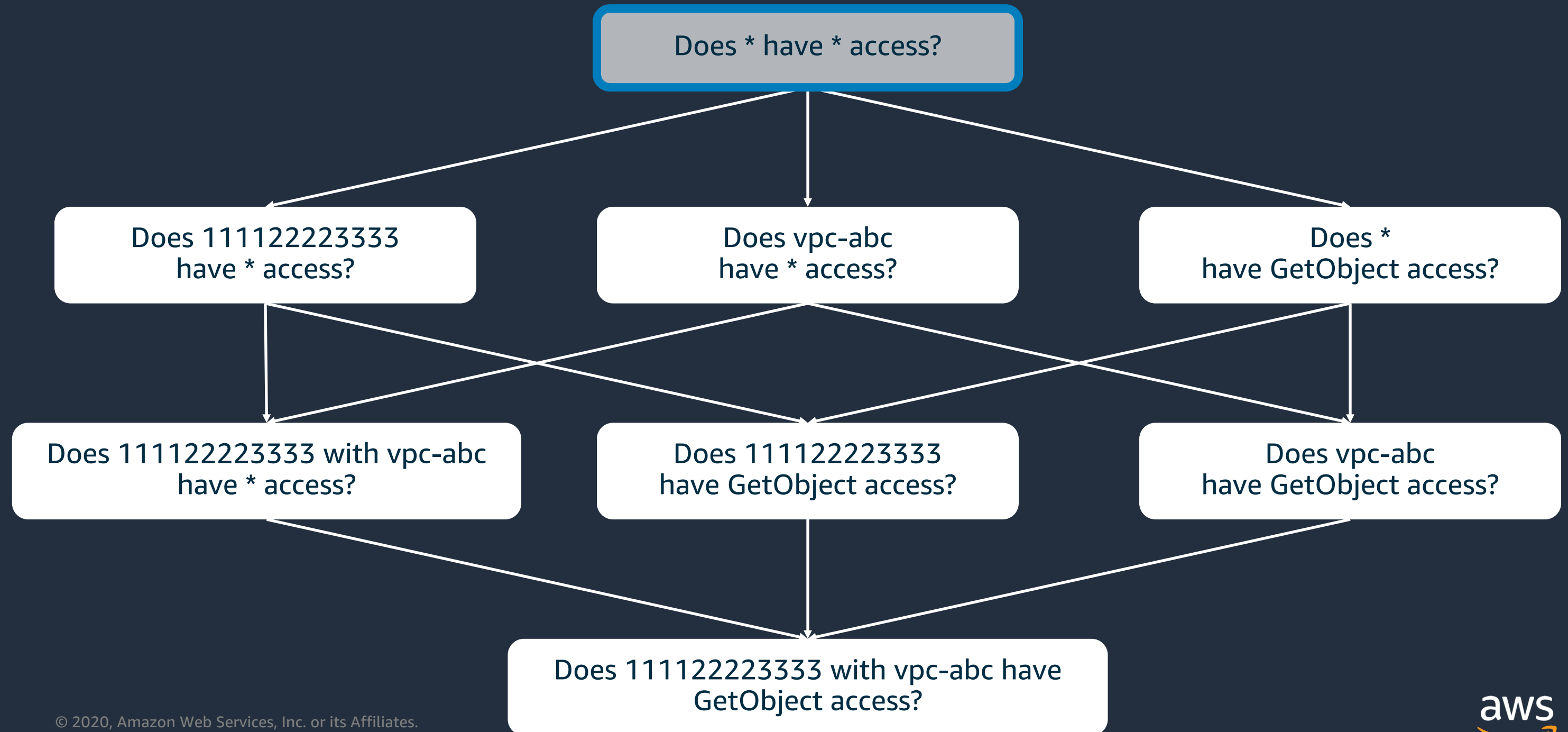
Twenty questions



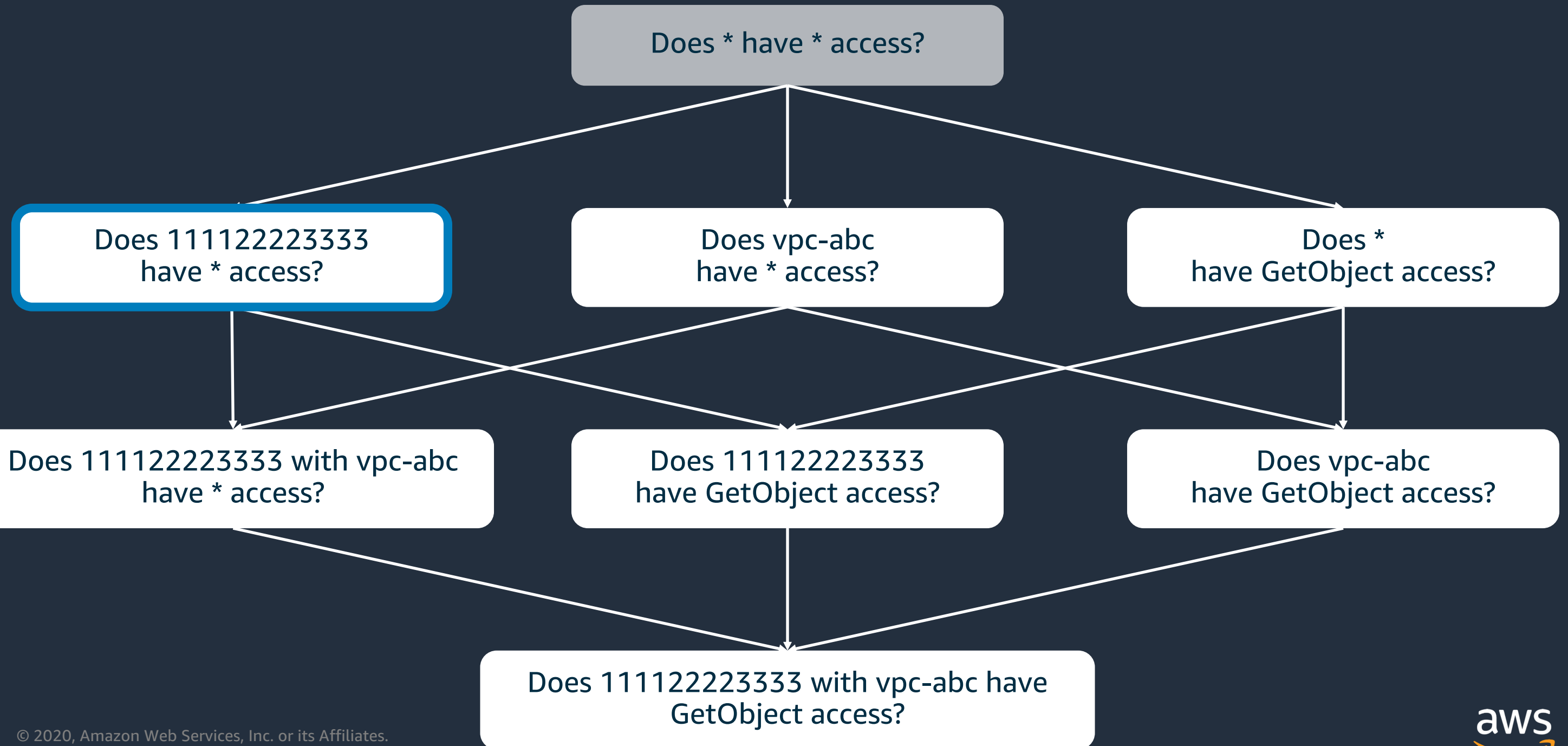
Twenty questions



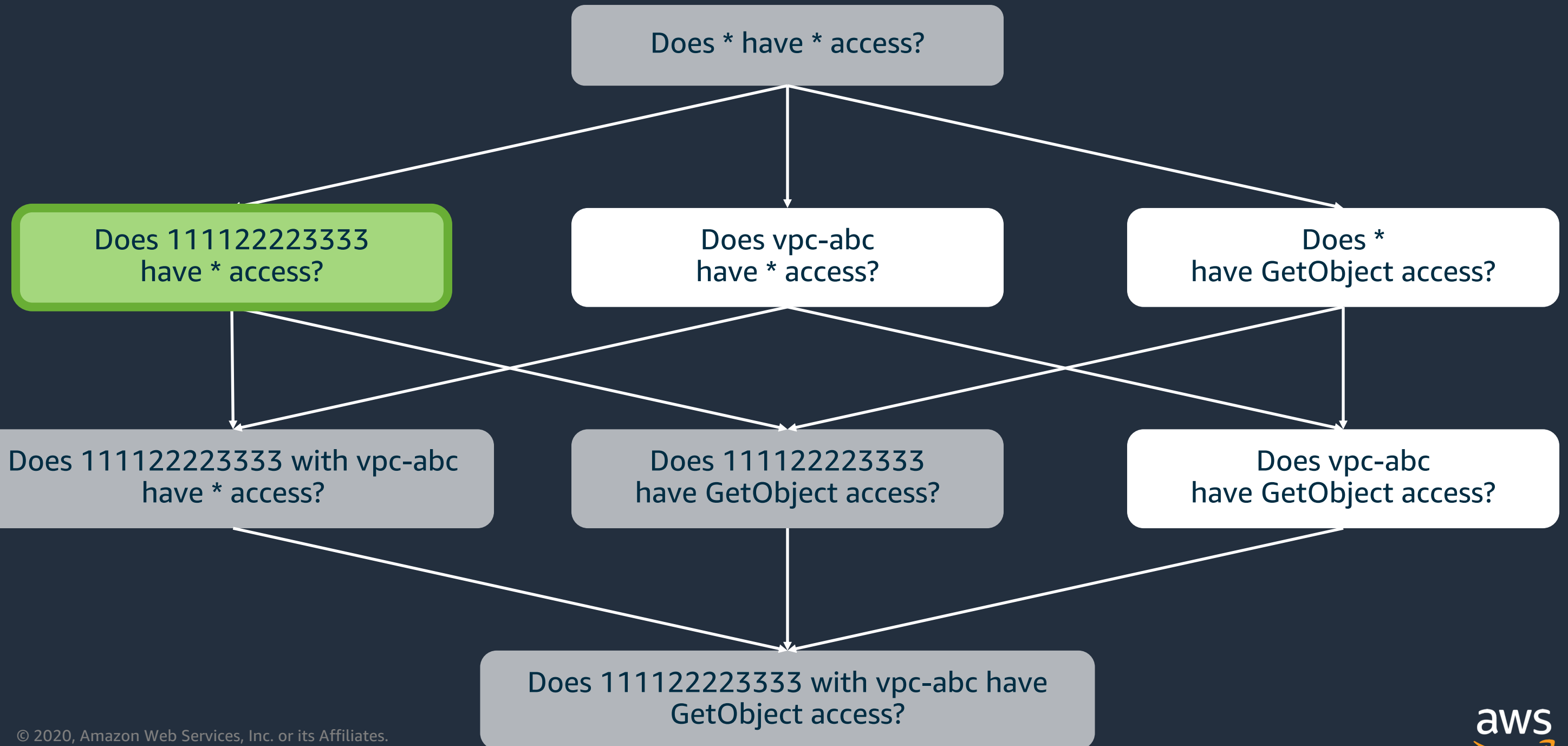
Twenty questions



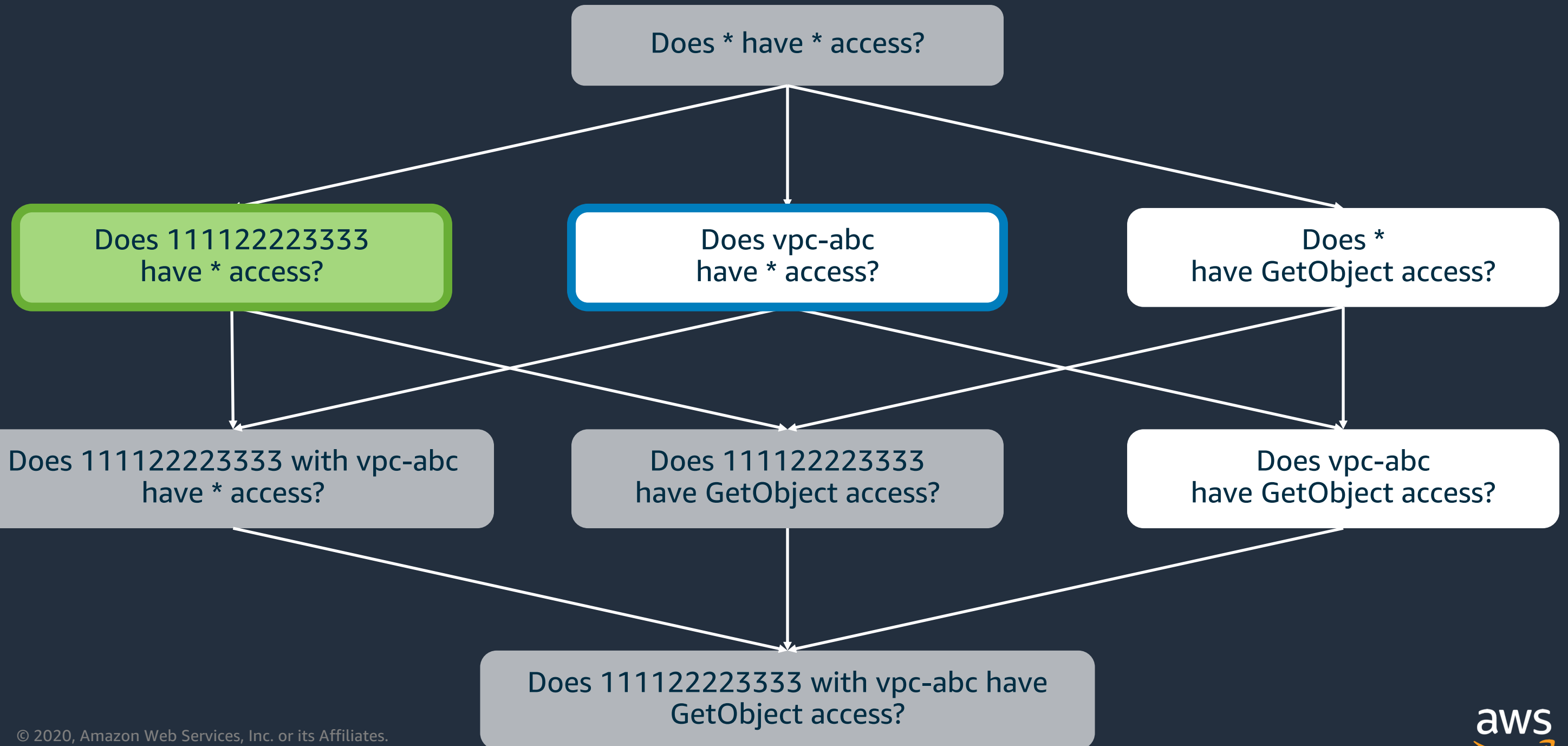
Twenty questions



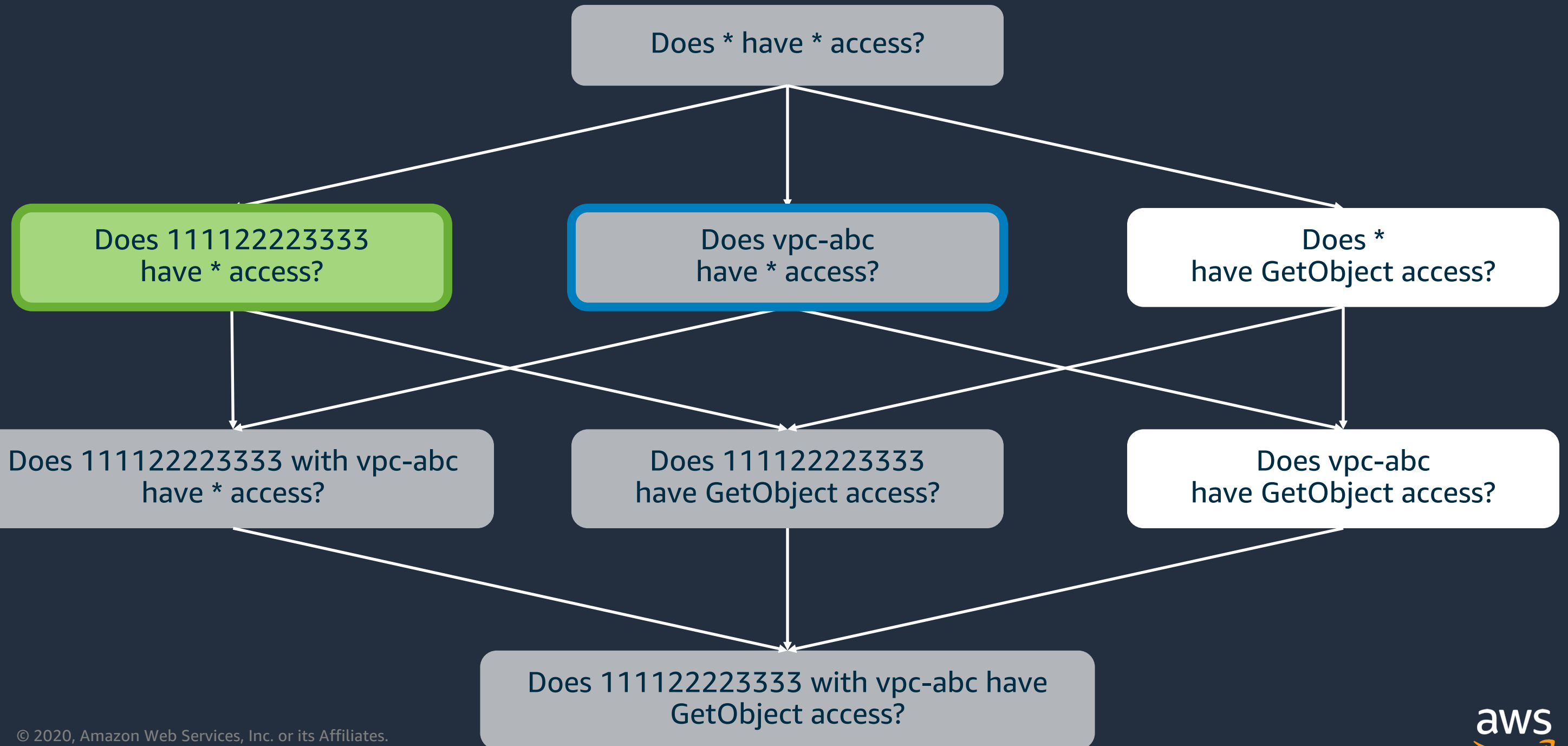
Twenty questions



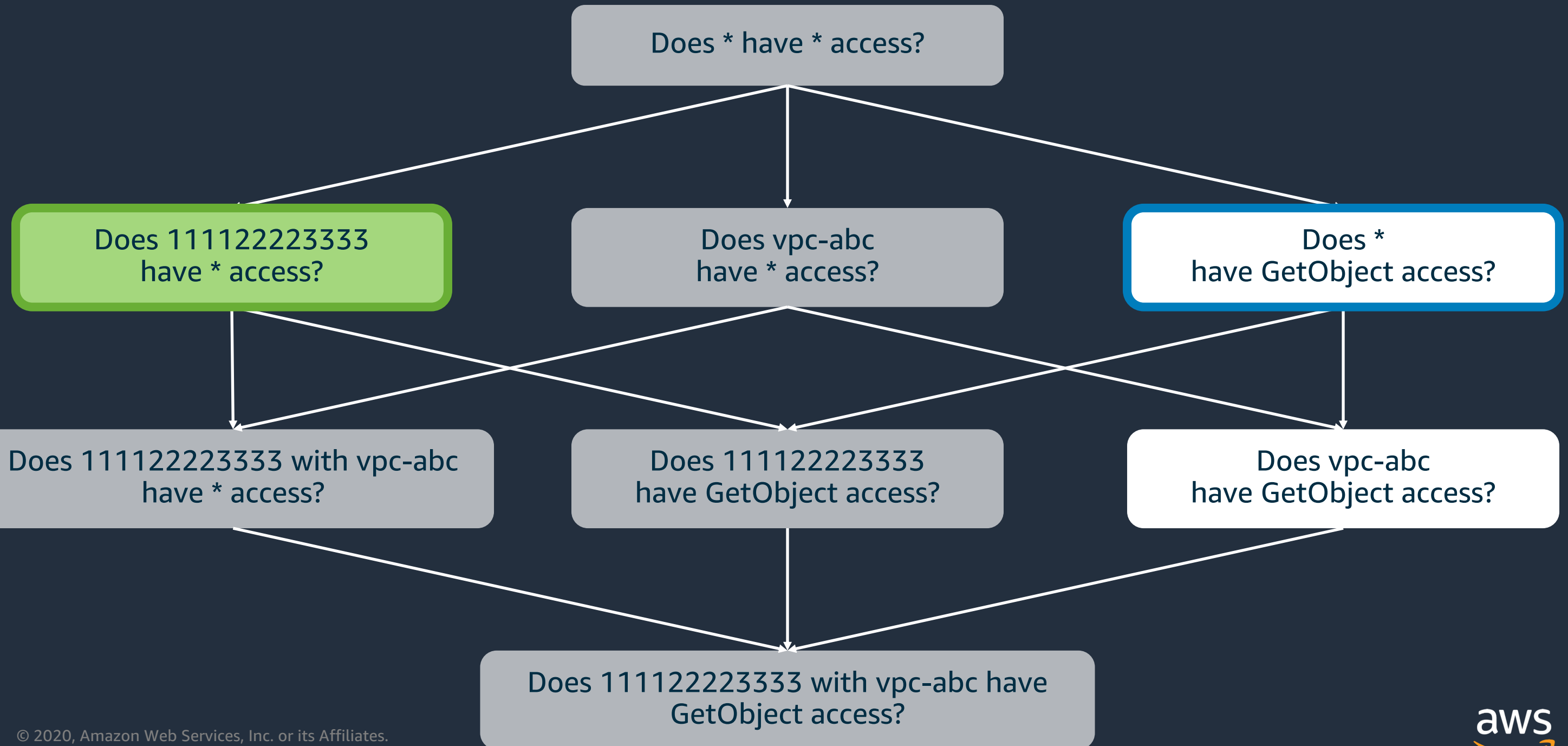
Twenty questions



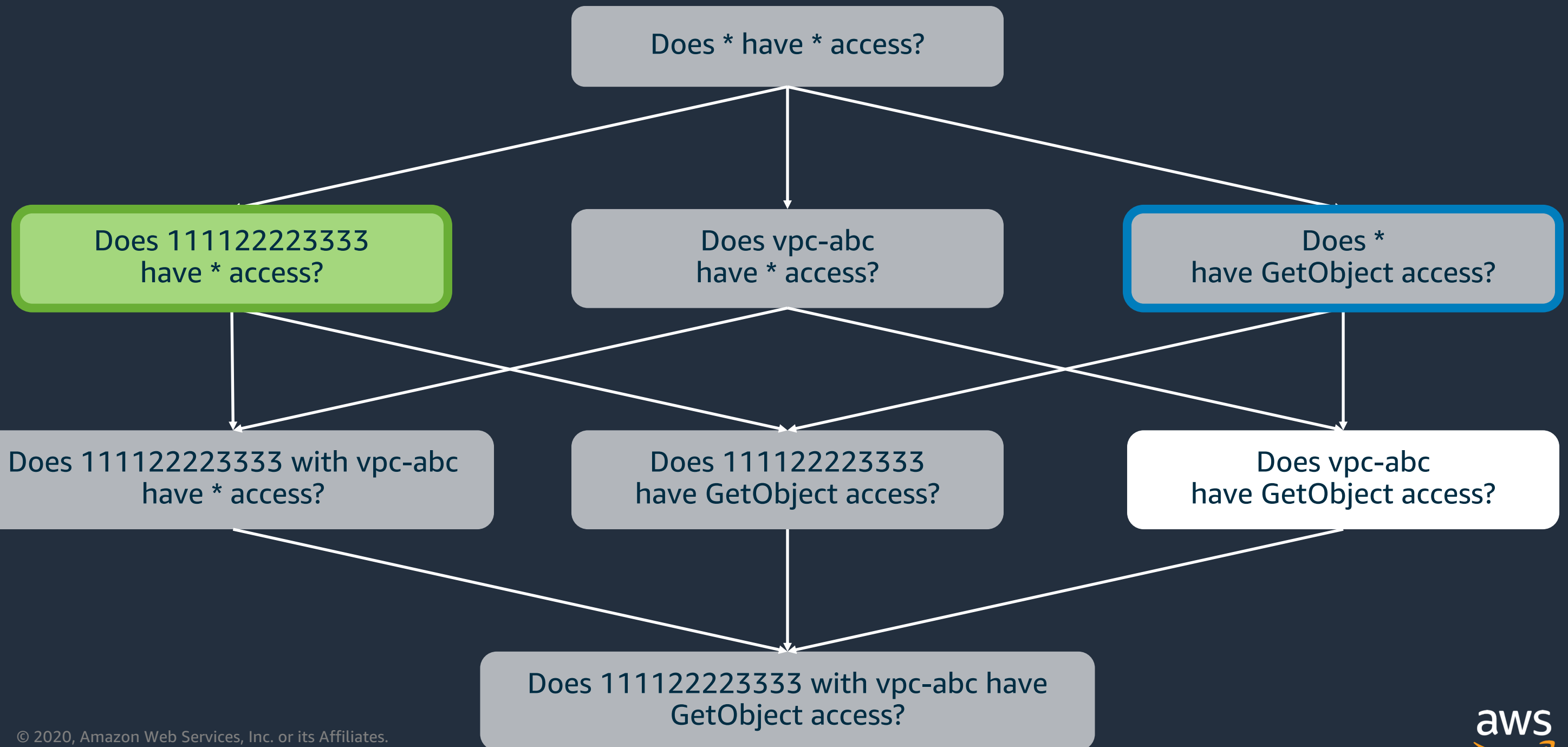
Twenty questions



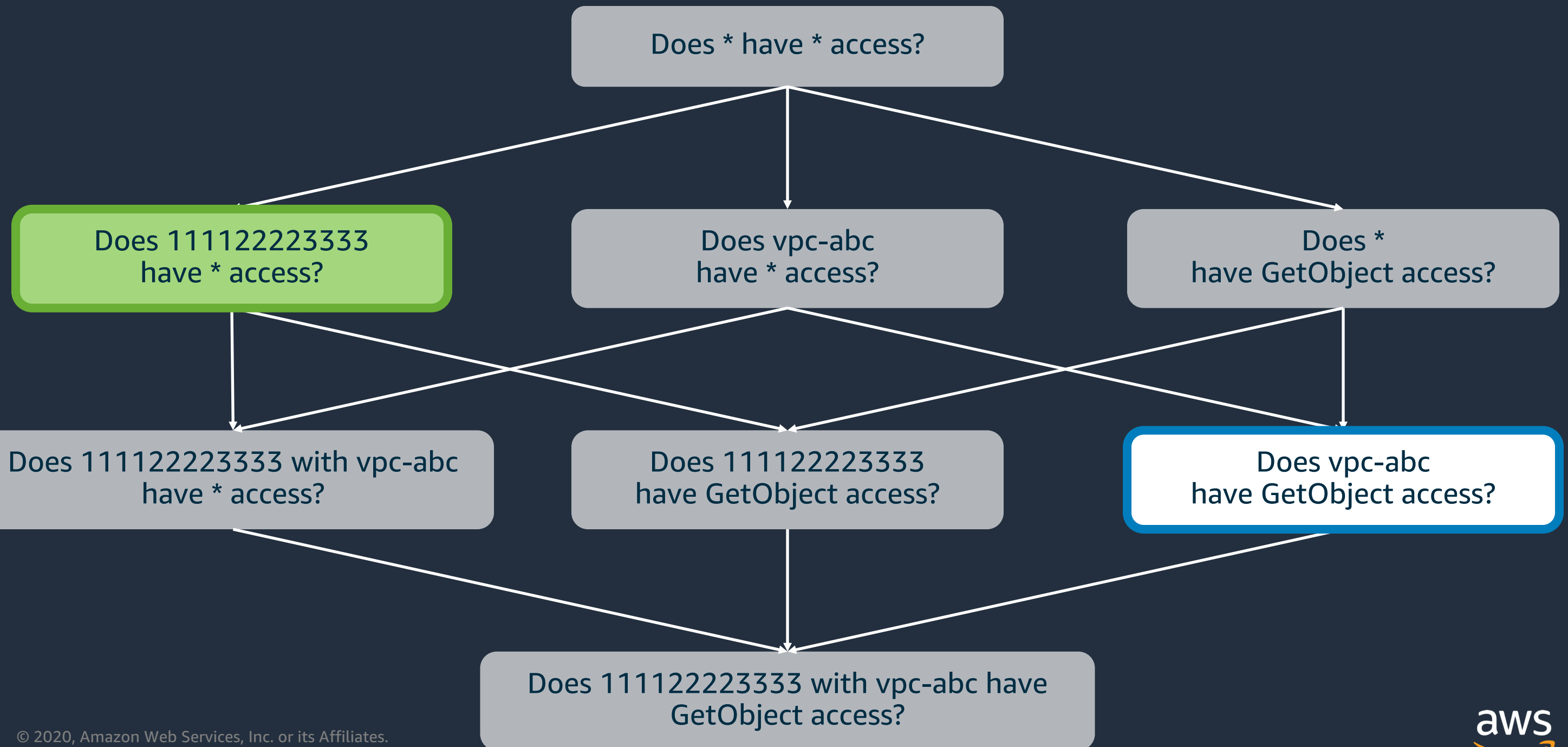
Twenty questions



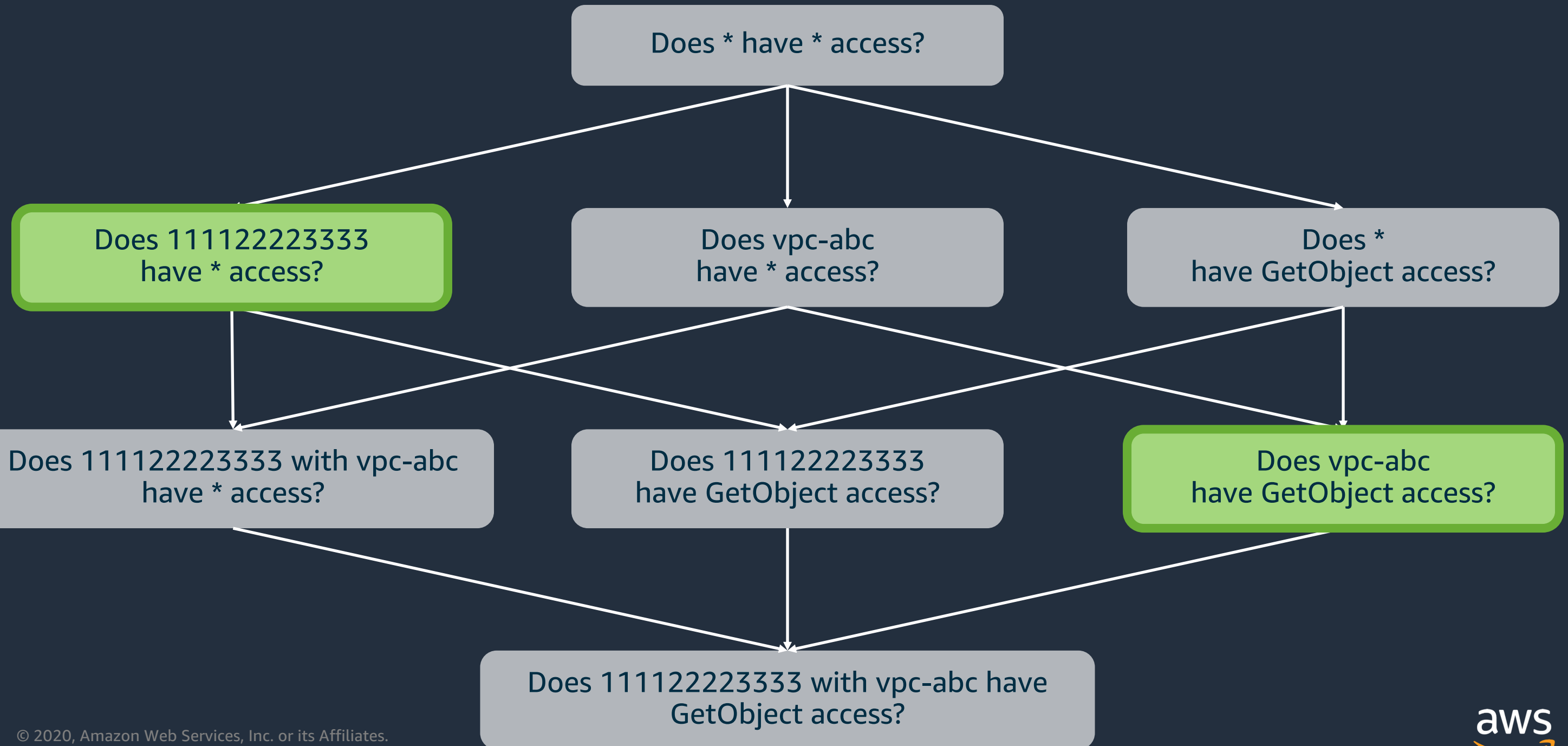
Twenty questions



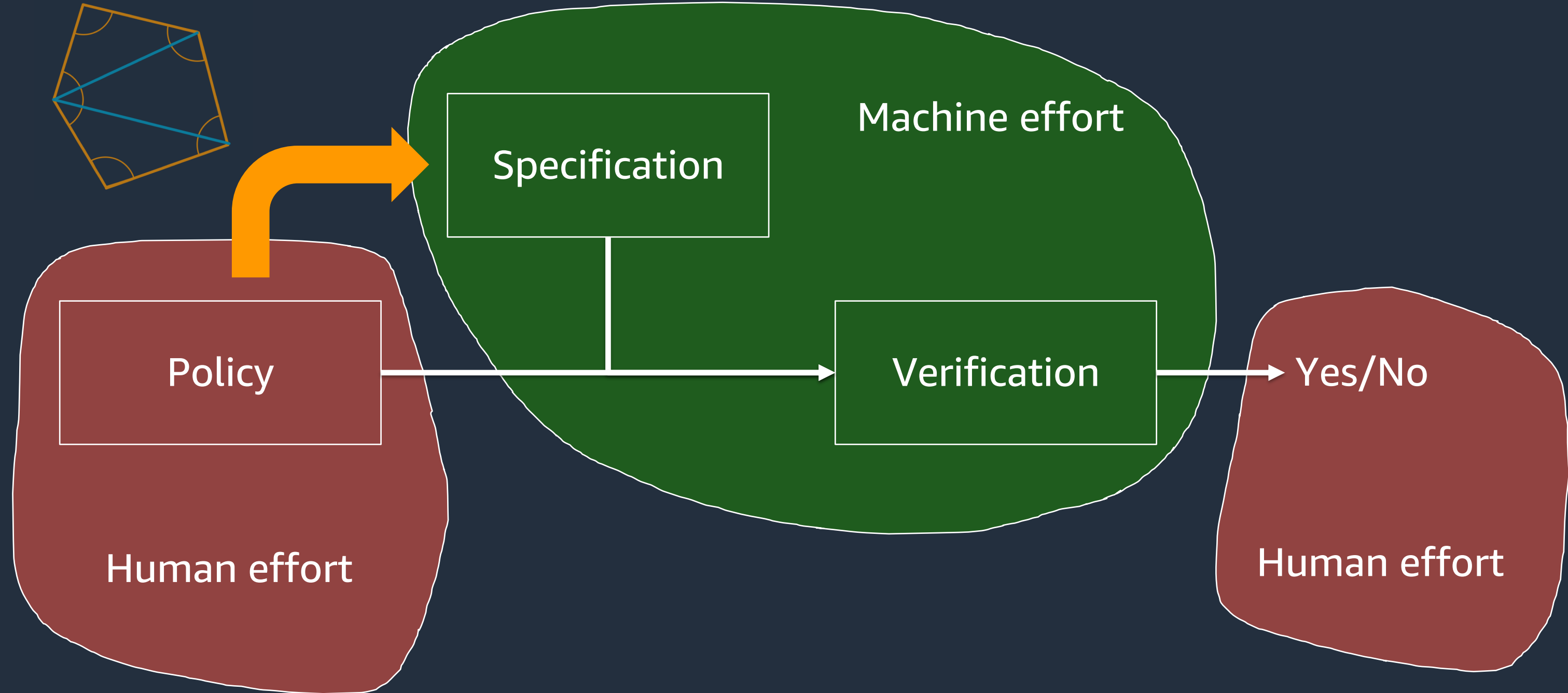
Twenty questions



Twenty questions



Access Analyzer verification approach



Access Analyzer Demo

Introducing AWS Identity and Access Management (IAM) Access Analyzer

Posted On: Dec 2, 2019

[AWS Identity and Access Management \(IAM\) Access Analyzer](#) is a new feature that makes it simple for security teams and administrators to check that their policies provide only the intended access to resources. Resource policies allow customers to granularly control who is able to access a specific resource and how they are able to use it across the entire cloud environment. With one click in the [IAM console](#), customers can enable IAM Access Analyzer across their account to continuously analyze permissions granted using policies associated with their Amazon S3 buckets, AWS KMS keys, Amazon SQS queues, AWS IAM roles, and AWS Lambda functions.

IAM Access Analyzer continuously monitors policies for changes, meaning customers no longer need to rely on intermittent manual checks in order to catch issues as policies are added or updated. Using IAM Access Analyzer, customers can proactively address any resource policies that violate their security and governance best practices around resource sharing and protect their resources from unintended access. IAM Access Analyzer delivers comprehensive, detailed findings through the AWS IAM, Amazon S3, and AWS Security Hub consoles and also through its APIs. Findings can also be exported as a report for auditing purposes. IAM Access Analyzer findings provide definitive answers of who has public and cross-account access to AWS resources from outside an account.

IAM Access Analyzer uses a form of mathematical analysis called automated reasoning, which applies logic and mathematical inference to determine all possible access paths allowed by a resource policy. This means that IAM Access Analyzer can evaluate hundreds or even thousands of policies across a customer's environment in seconds, and deliver comprehensive findings about resources that are accessible from outside the account. We call this [provable security](#).

With this launch, IAM Access Analyzer is available at no additional cost in the IAM console and through APIs in all commercial [AWS Regions](#). IAM Access Analyzer is also available through APIs in AWS GovCloud (US).

To learn more about IAM Access Analyzer, see

Introducing Access Analyzer for Amazon S3 to review access policies

Posted On: Dec 2, 2019

Access Analyzer for S3 is a new feature that monitors your access policies, ensuring that the policies provide only the intended access to your S3 resources. Access Analyzer for S3 evaluates your bucket access policies and enables you to discover and swiftly remediate buckets with potentially unintended access.

Access Analyzer for S3 alerts you when you have a bucket that is configured to allow access to anyone on the internet or that is shared with other AWS accounts. You receive insights or 'findings' into the source and level of public or shared access. For example, Access Analyzer for S3 will proactively inform you if read or write access were unintendedly provided through an access control list (ACL) or bucket policy. With these insights, you can immediately set or restore the intended access policy.

When reviewing results that show potentially shared access to a bucket, you can [Block All Public Access](#) to the bucket with a single click in the S3 Management console. You can also drill down into bucket level permission settings to configure granular levels of access. For specific and verified use cases that require public access, such as static website hosting, you can acknowledge and archive the findings on a bucket to record that you intend for the bucket to remain public or shared. You can revisit and modify these bucket configurations at any time. For auditing purposes, Access Analyzer for S3 findings can be downloaded as a CSV report.

To get started with Access Analyzer for S3, visit the IAM console to enable the [AWS Identity and Access Management \(IAM\) Access Analyzer](#). When you do this, Access Analyzer for S3 will automatically be visible in the S3 Management Console.

Access Analyzer for S3 is available at no additional cost in the S3 Management Console in all commercial [AWS Regions](#), excluding the AWS China (Beijing) Region and the AWS China (Ningxia) Region. Access Analyzer for S3 is also available through APIs in the AWS GovCloud (US) Regions.

To learn more, please read the [blog post](#).

IAM Access Analyzer is available at *no additional cost*

Access Analyzer for S3 is available at *no additional cost*

Access Analyzer

Monitor access to resources

How it works



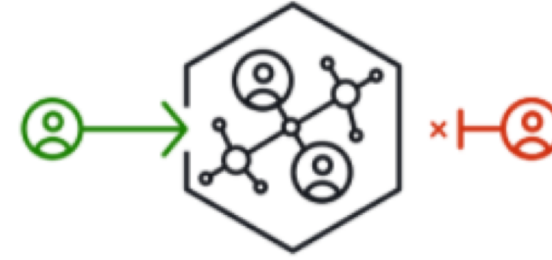
1 Create an analyzer

You can set the scope for the analyzer to an organization or an AWS account. This is your zone of trust. The analyzer scans all of the supported resources within your zone of trust.



2 Review active findings

When Access Analyzer finds a policy that allows access to a resource from outside of your zone of trust, it generates an active finding. Findings include details about the access so that you can take action.



3 Take action

If the access is intended, you can archive the finding so that you can focus on reviewing active findings. If the access is not intended, you can resolve the finding by modifying the policy to remove access to the resource.

Create analyzer

Getting started [↗](#)

- [What is Access Analyzer?](#)
- [Access Analyzer User Guide](#)

↺ Creating analyzer

↺ Creating analyzer

↺ Creating analyzer

↺ Creating analyzer

· Creating analyzer

✔ Analyzer creation is complete

4ff0daf3-28bc-4820-b43d-229d2be3a137 Info

 Rescan

Details

Finding ID

4ff0daf3-28bc-4820-b43d-229d2be3a137

Updated

a minute ago

Status

Active

Shared through

Bucket policy

Resource

[arn:aws:s3:::gacek-bucket-c](#) 

External principal

All Principals

Condition

Principal OrgID:
o-1
Source VPC:
vpc-b

Access level


Read

- s3:GetObject

Resource owner account
180286015604


Next steps

Intended access

If the access is intended, such as access necessary for business processes, you can archive the finding. This lets you focus on findings that are related to potential security risks. When you archive a finding, it's removed from Active findings and the status changes to  Archived.

Archive

Not intended

If the access isn't intended, it indicates a potential security risk. Use the console for the service associated with the resource to modify or remove the policy that grants the unintended access. To confirm that your change removed the access, choose **Rescan**. If the access is removed, the status changes to  **Resolved**.

Go to S3 console 

arn:aws:s3:::gacek-bucket-c

“The past is already written. The ink is dry.”

[IAM](#) > [Access Analyzer](#) > [Findings](#) > 4ff0daf3-28bc-4820-b43d-229d2be3a137

4ff0daf3-28bc-4820-b43d-229d2be3a137


Info

Rescan

Details

Finding ID	Updated	Status	Shared through
4ff0daf3-28bc-4820-b43d-229d2be3a137	a minute ago	Active	Bucket policy


Sensible findings

Details			
Finding ID 4ff0daf3-28bc-4820-b43d-229d2be3a137	Updated a minute ago	Status Active	Shared through Bucket policy
Resource arn:aws:s3:::gacek-bucket-c 	External principal All Principals	Condition Principal OrgID: o-1 Source VPC: vpc-b	Access level Read <ul style="list-style-type: none">s3:GetObject
Resource owner account 180286015604			

Pay-as-you-go specification


Next steps

Intended access

If the access is intended, such as access necessary for business processes, you can archive the finding. This lets you focus on findings that are related to potential security risks. When you archive a finding, it's removed from Active findings and the status changes to  Archived.

[Archive](#)

Not intended

If the access isn't intended, it indicates a potential security risk. Use the console for the service associated with the resource to modify or remove the policy that grants the unintended access. To confirm that your change removed the access, choose **Rescan**. If the access is removed, the status changes to  **Resolved**.

[Go to S3 console](#) 

arn:aws:s3:::gacek-bucket-c



Access Control Verification for Everyone



"All possible access paths are verified by mathematical proofs" 🥰



So cool that AWS now uses formal methods to analyze IAM Access!!



Oh, this is a big deal! Understanding IAM policy consequences is essential. This tool should make it a lot easier!

Identify Unintended Resource Access with AWS Identity and Access Management (IAM) Access Analyzer

9b90c684-401-473d-ac55-d2951ab31156	6 minutes ago	Active	
Resource arn:aws:kms:us-east-1:796744228948:key/A06385788-f529-487c-af53-c2665ad348b2	External principal (AWS Account) 418986291641	Condition -	Access level Write • kms:Decrypt • kms:Encrypt



🔒 The new IAM Access Analyzer is awesome! Glad to see AWS focusing on making it easier to verify workload security



Today we launched a first-of-its-kind service that uses automated reasoning to identify unintended resources access paths. It's pretty badass.



"IAM Access Analyzer provides answers of who has public and cross-account access to AWS resources."

Formal methods for the win:
"IAM Access Analyzer uses a form of mathematical analysis called automated reasoning, which applies logic and mathematical inference to ..."



Introducing AWS Identity and Access Management (IAM) Access Analyzer



New launch today—AWS IAM Access Analyzer—exciting provable security work out of the AWS Automated Reasoning Group [#reInvent2019](#)



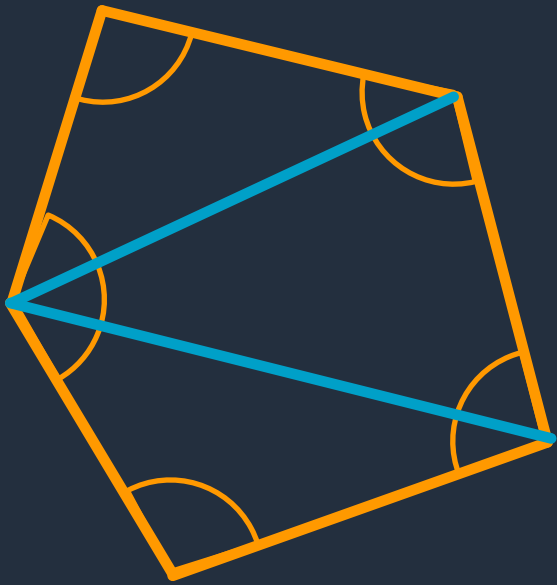
You don't need to be a logician to use IAM Access Analyzer. Turn this on now, it's available at no charge! [#reInvent](#) [#provablesecurity](#)



Just launched! IAM Access Analyzer: continuously monitor, comprehensively analyze, and gain certainty for cross account access controls. All backed by fancy math using automated reasoning. Come see it live in SEC316. But really go turn it on, it's quick.

Automated Reasoning at Amazon

<https://aws.amazon.com/security/provable-security/>



has



access to

