



API Reference Guide

Amazon Verified Permissions



Amazon Verified Permissions: API Reference Guide

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Welcome	1
Actions	3
BatchIsAuthorized	4
Request Syntax	4
Request Parameters	5
Response Syntax	6
Response Elements	7
Errors	7
Examples	9
See Also	13
BatchIsAuthorizedWithToken	15
Request Syntax	15
Request Parameters	16
Response Syntax	18
Response Elements	18
Errors	19
Examples	21
See Also	25
CreateIdentitySource	26
Request Syntax	27
Request Parameters	27
Response Syntax	29
Response Elements	29
Errors	30
Examples	32
See Also	33
CreatePolicy	35
Request Syntax	35
Request Parameters	35
Response Syntax	37
Response Elements	37
Errors	38
Examples	41
See Also	45

CreatePolicyStore	47
Request Syntax	47
Request Parameters	47
Response Syntax	49
Response Elements	49
Errors	50
Examples	52
See Also	53
CreatePolicyTemplate	54
Request Syntax	54
Request Parameters	54
Response Syntax	56
Response Elements	56
Errors	57
Examples	59
See Also	60
DeleteIdentitySource	62
Request Syntax	62
Request Parameters	62
Response Elements	63
Errors	63
Examples	65
See Also	66
DeletePolicy	67
Request Syntax	67
Request Parameters	67
Response Elements	68
Errors	68
Examples	70
See Also	71
DeletePolicyStore	72
Request Syntax	72
Request Parameters	72
Response Elements	72
Errors	73
Examples	74

See Also	75
DeletePolicyTemplate	77
Request Syntax	77
Request Parameters	77
Response Elements	78
Errors	78
Examples	80
See Also	81
GetIdentitySource	82
Request Syntax	82
Request Parameters	82
Response Syntax	83
Response Elements	83
Errors	84
Examples	86
See Also	88
GetPolicy	89
Request Syntax	89
Request Parameters	89
Response Syntax	90
Response Elements	90
Errors	91
Examples	93
See Also	95
GetPolicyStore	96
Request Syntax	96
Request Parameters	96
Response Syntax	96
Response Elements	97
Errors	98
Examples	100
See Also	101
GetPolicyTemplate	102
Request Syntax	102
Request Parameters	102
Response Syntax	103

Response Elements	103
Errors	104
Examples	106
See Also	107
GetSchema	109
Request Syntax	109
Request Parameters	109
Response Syntax	109
Response Elements	110
Errors	111
Examples	113
See Also	114
IsAuthorized	116
Request Syntax	116
Request Parameters	116
Response Syntax	118
Response Elements	118
Errors	119
Examples	121
See Also	128
IsAuthorizedWithToken	129
Request Syntax	129
Request Parameters	130
Response Syntax	132
Response Elements	132
Errors	133
Examples	135
See Also	137
ListIdentitySources	138
Request Syntax	138
Request Parameters	138
Response Syntax	140
Response Elements	140
Errors	141
Examples	143
See Also	144

ListPolicies	145
Request Syntax	145
Request Parameters	145
Response Syntax	147
Response Elements	147
Errors	148
Examples	150
See Also	157
ListPolicyStores	158
Request Syntax	158
Request Parameters	158
Response Syntax	159
Response Elements	159
Errors	160
Examples	162
See Also	163
ListPolicyTemplates	164
Request Syntax	164
Request Parameters	164
Response Syntax	165
Response Elements	166
Errors	166
Examples	168
See Also	169
PutSchema	171
Request Syntax	171
Request Parameters	171
Response Syntax	172
Response Elements	172
Errors	173
Examples	175
See Also	177
UpdateIdentitySource	178
Request Syntax	178
Request Parameters	178
Response Syntax	180

Response Elements	180
Errors	181
Examples	183
See Also	184
UpdatePolicy	185
Request Syntax	185
Request Parameters	186
Response Syntax	187
Response Elements	187
Errors	189
Examples	191
See Also	192
UpdatePolicyStore	194
Request Syntax	194
Request Parameters	194
Response Syntax	195
Response Elements	195
Errors	196
Examples	198
See Also	199
UpdatePolicyTemplate	201
Request Syntax	201
Request Parameters	201
Response Syntax	203
Response Elements	203
Errors	204
Examples	206
See Also	207
Data Types	209
ActionIdentifier	211
Contents	211
See Also	212
AttributeValue	213
Contents	213
See Also	214
BatchIsAuthorizedInputItem	216

Contents	216
See Also	217
BatchIsAuthorizedOutputItem	218
Contents	218
See Also	219
BatchIsAuthorizedWithTokenInputItem	220
Contents	220
See Also	220
BatchIsAuthorizedWithTokenOutputItem	222
Contents	222
See Also	223
CognitoGroupConfiguration	224
Contents	224
See Also	224
CognitoGroupConfigurationDetail	225
Contents	225
See Also	225
CognitoGroupConfigurationItem	226
Contents	226
See Also	226
CognitoUserPoolConfiguration	227
Contents	227
See Also	228
CognitoUserPoolConfigurationDetail	229
Contents	229
See Also	230
CognitoUserPoolConfigurationItem	232
Contents	232
See Also	233
Configuration	235
Contents	235
See Also	236
ConfigurationDetail	237
Contents	237
See Also	238
ConfigurationItem	239

Contents	239
See Also	240
ContextDefinition	241
Contents	241
See Also	241
DeterminingPolicyItem	243
Contents	243
See Also	243
EntitiesDefinition	244
Contents	244
See Also	244
EntityIdentifier	246
Contents	246
See Also	247
EntityItem	248
Contents	248
See Also	249
EntityReference	250
Contents	250
See Also	251
EvaluationErrorItem	252
Contents	252
See Also	252
IdentitySourceDetails	253
Contents	253
See Also	254
IdentitySourceFilter	256
Contents	256
See Also	256
IdentitySourceItem	257
Contents	257
See Also	258
IdentitySourceItemDetails	260
Contents	260
See Also	261
PolicyDefinition	263

Contents	263
See Also	264
PolicyDefinitionDetail	265
Contents	265
See Also	265
PolicyDefinitionItem	267
Contents	267
See Also	267
PolicyFilter	269
Contents	269
See Also	270
PolicyItem	271
Contents	271
See Also	273
PolicyStoreItem	274
Contents	274
See Also	275
PolicyTemplateItem	276
Contents	276
See Also	277
ResourceConflict	278
Contents	278
See Also	278
SchemaDefinition	279
Contents	279
See Also	279
StaticPolicyDefinition	281
Contents	281
See Also	281
StaticPolicyDefinitionDetail	283
Contents	283
See Also	283
StaticPolicyDefinitionItem	285
Contents	285
See Also	285
TemplateLinkedPolicyDefinition	286

Contents	286
See Also	287
TemplateLinkedPolicyDefinitionDetail	288
Contents	288
See Also	289
TemplateLinkedPolicyDefinitionItem	290
Contents	290
See Also	291
UpdateCognitoGroupConfiguration	292
Contents	292
See Also	292
UpdateCognitoUserPoolConfiguration	293
Contents	293
See Also	294
UpdateConfiguration	295
Contents	295
See Also	295
UpdatePolicyDefinition	297
Contents	297
See Also	297
UpdateStaticPolicyDefinition	298
Contents	298
See Also	299
ValidationExceptionField	300
Contents	300
See Also	300
ValidationSettings	301
Contents	301
See Also	302
Making API requests	303
Verified Permissions endpoints	303
Query parameters	303
Request identifiers	303
Query API authentication	304
Available libraries	304
Making API requests using the POST method	304

Common Parameters	307
Common Errors	310
Document history	312
AWS Glossary	313

Welcome

Amazon Verified Permissions is a permissions management service from AWS. You can use Verified Permissions to manage permissions for your application, and authorize user access based on those permissions. Using Verified Permissions, application developers can grant access based on information about the users, resources, and requested actions. You can also evaluate additional information like group membership, attributes of the resources, and session context, such as time of request and IP addresses. Verified Permissions manages these permissions by letting you create and store authorization policies for your applications, such as consumer-facing web sites and enterprise business systems.

Verified Permissions uses Cedar as the policy language to express your permission requirements. Cedar supports both role-based access control (RBAC) and attribute-based access control (ABAC) authorization models.

For more information about configuring, administering, and using Amazon Verified Permissions in your applications, see the [Amazon Verified Permissions User Guide](#).

For more information about the Cedar policy language, see the [Cedar Policy Language Guide](#).

Important

When you write Cedar policies that reference principals, resources and actions, you can define the unique identifiers used for each of those elements. We strongly recommend that you follow these best practices:

- **Use values like universally unique identifiers (UUIDs) for all principal and resource identifiers.**

For example, if user `jane` leaves the company, and you later let someone else use the name `jane`, then that new user automatically gets access to everything granted by policies that still reference `User1 : "jane"`. Cedar can't distinguish between the new user and the old. This applies to both principal and resource identifiers. Always use identifiers that are guaranteed unique and never reused to ensure that you don't unintentionally grant access because of the presence of an old identifier in a policy.

Where you use a UUID for an entity, we recommend that you follow it with the `//` comment specifier and the 'friendly' name of your entity. This helps to make your policies

easier to understand. For example: `principal == User::"a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111", // alice`

- **Do not include personally identifying, confidential, or sensitive information as part of the unique identifier for your principals or resources.** These identifiers are included in log entries shared in AWS CloudTrail trails.

Several operations return structures that appear similar, but have different purposes. As new functionality is added to the product, the structure used in a parameter of one operation might need to change in a way that wouldn't make sense for the same parameter in a different operation. To help you understand the purpose of each, the following naming convention is used for the structures:

- Parameter type structures that end in `Detail` are used in `Get` operations.
- Parameter type structures that end in `Item` are used in `List` operations.
- Parameter type structures that use neither suffix are used in the mutating (create and update) operations.

Note

The example HTTP query requests and responses in this guide are displayed with the [JSON](#) formatted across multiple lines for readability. The actual query responses from the Amazon Verified Permissions service do not include this extra whitespace.

We want your feedback about this documentation

Our goal is to help you get everything you can from Amazon Verified Permissions. If this guide helps you to do that, then let us know. If the guide isn't helping you, then we want to hear from you so we can address the issue. Use the **Feedback** link that's in the upper-right corner of every page. That sends your comments directly to the writers of this guide. We review every submission, looking for opportunities to improve the documentation. Thank you in advance for your help!

This document was last published on May 10, 2024.

Actions

The following actions are supported:

- [BatchIsAuthorized](#)
- [BatchIsAuthorizedWithToken](#)
- [CreateIdentitySource](#)
- [CreatePolicy](#)
- [CreatePolicyStore](#)
- [CreatePolicyTemplate](#)
- [DeleteIdentitySource](#)
- [DeletePolicy](#)
- [DeletePolicyStore](#)
- [DeletePolicyTemplate](#)
- [GetIdentitySource](#)
- [GetPolicy](#)
- [GetPolicyStore](#)
- [GetPolicyTemplate](#)
- [GetSchema](#)
- [IsAuthorized](#)
- [IsAuthorizedWithToken](#)
- [ListIdentitySources](#)
- [ListPolicies](#)
- [ListPolicyStores](#)
- [ListPolicyTemplates](#)
- [PutSchema](#)
- [UpdateIdentitySource](#)
- [UpdatePolicy](#)
- [UpdatePolicyStore](#)
- [UpdatePolicyTemplate](#)

BatchIsAuthorized

Makes a series of decisions about multiple authorization requests for one principal or resource. Each request contains the equivalent content of an `IsAuthorized` request: `principal`, `action`, `resource`, and `context`. Either the `principal` or the `resource` parameter must be identical across all requests. For example, Verified Permissions won't evaluate a pair of requests where bob views photo1 and alice views photo2. Authorization of bob to view photo1 and photo2, or bob and alice to view photo1, are valid batches.

The request is evaluated against all policies in the specified policy store that match the entities that you declare. The result of the decisions is a series of `Allow` or `Deny` responses, along with the IDs of the policies that produced each decision.

The entities of a `BatchIsAuthorized` API request can contain up to 100 principals and up to 100 resources. The requests of a `BatchIsAuthorized` API request can contain up to 30 requests.

Note

The `BatchIsAuthorized` operation doesn't have its own IAM permission. To authorize this operation for AWS principals, include the permission `verifiedpermissions:IsAuthorized` in their IAM policies.

Request Syntax

```
{
  "entities": { ... },
  "policyStoreId": "string",
  "requests": [
    {
      "action": {
        "actionId": "string",
        "actionType": "string"
      },
      "context": { ... },
      "principal": {
        "entityId": "string",
        "entityType": "string"
      }
    }
  ]
}
```

```
    },
    "resource": {
      "entityId": "string",
      "entityType": "string"
    }
  ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

policyStoreId

Specifies the ID of the policy store. Policies in this policy store will be used to make the authorization decisions for the input.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

requests

An array of up to 30 requests that you want Verified Permissions to evaluate.

Type: Array of [BatchIsAuthorizedInputItem](#) objects

Array Members: Minimum number of 1 item.

Required: Yes

entities

Specifies the list of resources and principals and their associated attributes that Verified Permissions can examine when evaluating the policies.

Note

You can include only principal and resource entities in this parameter; you can't include actions. You must specify actions in the schema.

Type: [EntitiesDefinition](#) object

Note: This object is a Union. Only one member of this object can be specified or returned.

Required: No

Response Syntax

```
{
  "results": [
    {
      "decision": "string",
      "determiningPolicies": [
        {
          "policyId": "string"
        }
      ],
      "errors": [
        {
          "errorDescription": "string"
        }
      ],
      "request": {
        "action": {
          "actionId": "string",
          "actionType": "string"
        },
        "context": { ... },
        "principal": {
          "entityId": "string",
```

```
    "entityType": "string"
  },
  "resource": {
    "entityId": "string",
    "entityType": "string"
  }
}
]
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

results

A series of Allow or Deny decisions for each request, and the policies that produced them.

Type: Array of [BatchIsAuthorizedOutputItem](#) objects

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerError

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its

value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example requests for multiple principals and actions with one resource

The following example requests two authorization decisions for two principals of type `User` named Alice and Annalisa. Alice wants to perform the `ViewPhoto` operation on a resource of type `Photo` named `VacationPhoto94.jpg`. The photo is in Alice's account. Annalisa wants to delete `VacationPhoto94.jpg`.

The response shows that Alice's request was allowed by one policy and Annalisa's request was denied because the photo is in someone else's account.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.BatchIsAuthorized
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>
```

```
{
  "requests": [
    {
      "principal": {
        "entityType": "PhotoFlash::User",
        "entityId": "Alice"
      },
      "action": {
        "actionType": "PhotoFlash::Action",
        "actionId": "ViewPhoto"
      },
      "resource": {
        "entityType": "PhotoFlash::Photo",
        "entityId": "VacationPhoto94.jpg"
      }
    },
    {
      "principal": {
        "entityType": "PhotoFlash::User",
        "entityId": "Annalisa"
      },
      "action": {
        "actionType": "PhotoFlash::Action",
        "actionId": "DeletePhoto"
      },
      "resource": {
        "entityType": "PhotoFlash::Photo",
        "entityId": "VacationPhoto94.jpg"
      }
    }
  ],
  "entities": {
    "entityList": [
      {
        "identifier": {
          "entityType": "PhotoFlash::User",
          "entityId": "Alice"
        },
        "attributes": {
          "Account": {
            "entityIdentifier": {
              "entityType": "PhotoFlash::Account",
              "entityId": "1234"
            }
          }
        }
      }
    ]
  }
}
```

```
    },
    "Email": {
      "string": ""
    }
  },
  "parents": []
},
{
  "identifier": {
    "entityType": "PhotoFlash::User",
    "entityId": "Annalisa"
  },
  "attributes": {
    "Account": {
      "entityIdentifier": {
        "entityType": "PhotoFlash::Account",
        "entityId": "5678"
      }
    },
    "Email": {
      "string": ""
    }
  },
  "parents": []
},
{
  "identifier": {
    "entityType": "PhotoFlash::Photo",
    "entityId": "VacationPhoto94.jpg"
  },
  "attributes": {
    "IsPrivate": {
      "boolean": false
    },
    "Name": {
      "string": ""
    }
  },
  "parents": [
    {
      "entityType": "PhotoFlash::Account",
      "entityId": "1234"
    }
  ]
}
```



```
    },
    {
      "identifier": {
        "entityType": "PhotoFlash::Account",
        "entityId": "1234"
      },
      "attributes": {
        "Name": {
          "string": ""
        }
      },
      "parents": []
    }
  ]
},
"policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE111111
Connection: keep-alive
```

```
{
  "results": [
    {
      "request": {
        "principal": {
          "entityType": "PhotoFlash::User",
          "entityId": "alice"
        },
        "action": {
          "actionType": "PhotoFlash::Action",
          "actionId": "ViewPhoto"
        },
        "resource": {
```

```
        "entityType": "PhotoFlash::Photo",
        "entityId": "VacationPhoto94.jpg"
    },
    "decision": "ALLOW",
    "determiningPolicies": [
        {
            "policyId": "SPEXAMPLEEabcdefg111111"
        }
    ],
    "errors": []
},
{
    "request": {
        "principal": {
            "entityType": "PhotoFlash::User",
            "entityId": "annalisa"
        },
        "action": {
            "actionType": "PhotoFlash::Action",
            "actionId": "DeletePhoto"
        },
        "resource": {
            "entityType": "PhotoFlash::Photo",
            "entityId": "VacationPhoto94.jpg"
        }
    },
    "decision": "DENY",
    "determiningPolicies": [],
    "errors": []
}
]
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)

- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

BatchIsAuthorizedWithToken

Makes a series of decisions about multiple authorization requests for one token. The principal in this request comes from an external identity source in the form of an identity or access token, formatted as a [JSON web token \(JWT\)](#). The information in the parameters can also define additional context that Verified Permissions can include in the evaluations.

The request is evaluated against all policies in the specified policy store that match the entities that you provide in the entities declaration and in the token. The result of the decisions is a series of Allow or Deny responses, along with the IDs of the policies that produced each decision.

The entities of a BatchIsAuthorizedWithToken API request can contain up to 100 resources and up to 99 user groups. The requests of a BatchIsAuthorizedWithToken API request can contain up to 30 requests.

Note

The BatchIsAuthorizedWithToken operation doesn't have its own IAM permission. To authorize this operation for AWS principals, include the permission `verifiedpermissions:IsAuthorizedWithToken` in their IAM policies.

Request Syntax

```
{
  "accessToken": "string",
  "entities": { ... },
  "identityToken": "string",
  "policyStoreId": "string",
  "requests": [
    {
      "action": {
        "actionId": "string",
        "actionType": "string"
      },
      "context": { ... },
      "resource": {
        "entityId": "string",
        "entityType": "string"
      }
    }
  ]
}
```

```
]
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

policyStoreId

Specifies the ID of the policy store. Policies in this policy store will be used to make an authorization decision for the input.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

requests

An array of up to 30 requests that you want Verified Permissions to evaluate.

Type: Array of [BatchIsAuthorizedWithTokenInputItem](#) objects

Array Members: Minimum number of 1 item.

Required: Yes

accessToken

Specifies an access token for the principal that you want to authorize in each request. This token is provided to you by the identity provider (IdP) associated with the specified identity source. You must specify either an `accessToken`, an `identityToken`, or both.

Must be an access token. Verified Permissions returns an error if the `token_use` claim in the submitted token isn't `access`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 131072.

Pattern: `[A-Za-z0-9-_=]+.[A-Za-z0-9-_=]+.[A-Za-z0-9-_=]+`

Required: No

entities

Specifies the list of resources and their associated attributes that Verified Permissions can examine when evaluating the policies.

Important

You can't include principals in this parameter, only resource and action entities. This parameter can't include any entities of a type that matches the user or group entity types that you defined in your identity source.

- The `BatchIsAuthorizedWithToken` operation takes principal attributes from **only** the `identityToken` or `accessToken` passed to the operation.
- For action entities, you can include only their `Identifier` and `EntityType`.

Type: [EntitiesDefinition](#) object

Note: This object is a Union. Only one member of this object can be specified or returned.

Required: No

identityToken

Specifies an identity (ID) token for the principal that you want to authorize in each request. This token is provided to you by the identity provider (IdP) associated with the specified identity source. You must specify either an `accessToken`, an `identityToken`, or both.

Must be an ID token. Verified Permissions returns an error if the `token_use` claim in the submitted token isn't `id`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 131072.

Pattern: `[A-Za-z0-9-_=]+.[A-Za-z0-9-_=]+.[A-Za-z0-9-_=]+`

Required: No

Response Syntax

```
{
  "principal": {
    "entityId": "string",
    "entityType": "string"
  },
  "results": [
    {
      "decision": "string",
      "determiningPolicies": [
        {
          "policyId": "string"
        }
      ],
      "errors": [
        {
          "errorDescription": "string"
        }
      ],
      "request": {
        "action": {
          "actionId": "string",
          "actionType": "string"
        },
        "context": { ... },
        "resource": {
          "entityId": "string",
          "entityType": "string"
        }
      }
    }
  ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

results

A series of Allow or Deny decisions for each request, and the policies that produced them.

Type: Array of [BatchIsAuthorizedWithTokenOutputItem](#) objects

principal

The identifier of the principal in the ID or access token.

Type: [EntityIdentifier](#) object

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerError

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example requests for multiple actions and resource

The following example requests two authorization decisions for a principal from a user pool token. This variation on the PhotoFlash sample policy store has the following policy:

- `principal in PhotoFlash::FriendGroup::"us-east-1_EXAMPLE|MyExampleGroup"`

The user's token contains a `cognito:groups` claim that includes `MyExampleGroup`.

- `action in [PhotoFlash::Action::"FullPhotoAccess"]`

The actions `ViewPhoto` and `SharePhoto` have `FullPhotoAccess` as a parent in the policy store schema.

- `resource in PhotoFlash::Album::"MyExampleAlbum1"`

The album membership of the photos is declared in `entities`.

The result of this batch of requests is that the user can view and share a photo in an album that is authorized for their friend group, but not a photo in a different album.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.BatchIsAuthorizedWithToken
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
```

```
"identityToken": "eyJra12345EXAMPLE",
"requests": [
  {
    "action": {
      "actionType": "PhotoFlash::Action",
      "actionId": "ViewPhoto"
    },
    "resource": {
      "entityType": "PhotoFlash::Photo",
      "entityId": "VacationPhoto94.jpg"
    }
  },
  {
    "action": {
      "actionType": "PhotoFlash::Action",
      "actionId": "SharePhoto"
    },
    "resource": {
      "entityType": "PhotoFlash::Photo",
      "entityId": "VacationPhoto94.jpg"
    }
  },
  {
    "action": {
      "actionType": "PhotoFlash::Action",
      "actionId": "ViewPhoto"
    },
    "resource": {
      "entityType": "PhotoFlash::Photo",
      "entityId": "OfficePhoto94.jpg"
    }
  }
],
"entities": {
  "entityList": [
    {
      "identifier": {
        "entityType": "PhotoFlash::Photo",
        "entityId": "VacationPhoto94.jpg"
      },
      "parents": [
        {
          "entityType": "PhotoFlash::Album",
          "entityId": "MyExampleAlbum1"
        }
      ]
    }
  ]
}
```

```
    }
  ]
},
{
  "identifier": {
    "entityType": "PhotoFlash::Photo",
    "entityId": "OfficePhoto94.jpg"
  },
  "parents": [
    {
      "entityType": "PhotoFlash::Album",
      "entityId": "MyExampleAlbum2"
    }
  ]
}
]
},
"policyStoreId": "PSEXAMPLEEabcdefg111111"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "principal": {
    "entityType": "PhotoFlash::User",
    "entityId": "us-east-1_EXAMPLE|a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  },
  "results": [
    {
      "request": {
        "action": {
          "actionType": "PhotoFlash::Action",
          "actionId": "ViewPhoto"
        }
      }
    }
  ]
}
```

```
    },
    "resource": {
      "entityType": "PhotoFlash::Photo",
      "entityId": "VacationPhoto94.jpg"
    }
  },
  "decision": "ALLOW",
  "determiningPolicies": [
    {
      "policyId": "SPEXAMPLEabcdefgh111111"
    }
  ],
  "errors": []
},
{
  "request": {
    "action": {
      "actionType": "PhotoFlash::Action",
      "actionId": "SharePhoto"
    },
    "resource": {
      "entityType": "PhotoFlash::Photo",
      "entityId": "VacationPhoto94.jpg"
    }
  },
  "decision": "ALLOW",
  "determiningPolicies": [
    {
      "policyId": "SPEXAMPLEabcdefgh111111"
    }
  ],
  "errors": []
},
{
  "request": {
    "action": {
      "actionType": "PhotoFlash::Action",
      "actionId": "ViewPhoto"
    },
    "resource": {
      "entityType": "PhotoFlash::Photo",
      "entityId": "OfficePhoto94.jpg"
    }
  },
  "decision": "ALLOW",
  "determiningPolicies": [
    {
      "policyId": "SPEXAMPLEabcdefgh111111"
    }
  ],
  "errors": []
}
```

```
    "decision": "DENY",
    "determiningPolicies": [],
    "errors": []
  }
]
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

CreateIdentitySource

Creates a reference to an Amazon Cognito user pool as an external identity provider (IdP).

After you create an identity source, you can use the identities provided by the IdP as proxies for the principal in authorization queries that use the [IsAuthorizedWithToken](#) operation. These identities take the form of tokens that contain claims about the user, such as IDs, attributes and group memberships. Amazon Cognito provides both identity tokens and access tokens, and Verified Permissions can use either or both. Any combination of identity and access tokens results in the same Cedar principal. Verified Permissions automatically translates the information about the identities into the standard Cedar attributes that can be evaluated by your policies. Because the Amazon Cognito identity and access tokens can contain different information, the tokens you choose to use determine which principal attributes are available to access when evaluating Cedar policies.

Important

If you delete a Amazon Cognito user pool or user, tokens from that deleted pool or that deleted user continue to be usable until they expire.

Note

To reference a user from this identity source in your Cedar policies, use the following syntax.

```
IdentityType::"<CognitoUserPoolIdentifier>|<CognitoClientId>
```

Where `IdentityType` is the string that you provide to the `PrincipalEntityType` parameter for this operation. The `CognitoUserPoolId` and `CognitoClientId` are defined by the Amazon Cognito user pool.

Note

Verified Permissions is [eventually consistent](#). It can take a few seconds for a new or changed element to propagate through the service and be visible in the results of other Verified Permissions operations.

Request Syntax

```
{  
  "clientToken": "string",  
  "configuration": { ... },  
  "policyStoreId": "string",  
  "principalEntityType": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

configuration

Specifies the details required to communicate with the identity provider (IdP) associated with this identity source.

Note

At this time, the only valid member of this structure is a Amazon Cognito user pool configuration.

You must specify a `UserPoolArn`, and optionally, a `ClientId`.

Type: [Configuration](#) object

Note: This object is a Union. Only one member of this object can be specified or returned.

Required: Yes

policyStoreId

Specifies the ID of the policy store in which you want to store this identity source. Only policies and requests made using this policy store can reference identities from the identity provider configured in the new identity source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

clientToken

Specifies a unique, case-sensitive ID that you provide to ensure the idempotency of the request. This lets you safely retry the request without accidentally performing the same operation a second time. Passing the same value to a later call to an operation requires that you also pass the same value for all other parameters. We recommend that you use a [UUID type of value](#).

If you don't provide this value, then AWS generates a random one for you.

If you retry the operation with the same `ClientToken`, but with different parameters, the retry fails with an `ConflictException` error.

Verified Permissions recognizes a `ClientToken` for eight hours. After eight hours, the next request with the same parameters performs the operation again regardless of the value of `ClientToken`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: [a-zA-Z0-9-]*

Required: No

principalEntityType

Specifies the namespace and data type of the principals generated for identities authenticated by the new identity source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: .*

Required: No

Response Syntax

```
{
  "createdDate": "string",
  "identitySourceId": "string",
  "lastUpdatedDate": "string",
  "policyStoreId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

createdDate

The date and time the identity source was originally created.

Type: Timestamp

identitySourceId

The unique ID of the new identity source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

lastUpdatedDate

The date and time the identity source was most recently updated.

Type: Timestamp

policyStoreId

The ID of the policy store that contains the identity source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

ConflictException

The request failed because another request to modify a resource occurred at the same.

HTTP Status Code: 400

InternalServerError

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ServiceQuotaExceededException

The request failed because it would cause a service quota to be exceeded.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example

The following example request creates an identity source that Verified Permissions can use to retrieve authenticated identities to use in evaluations. The specified identity provider (IdP) is a Amazon Cognito user pool.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.CreateIdentitySource
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "configuration": {
    "cognitoUserPoolConfiguration": {
      "userPoolArn": "arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-
east-1_1a2b3c4d5",
      "clientIds": ["a1b2c3d4e5f6g7h8i9j0kalbmc"],
      "groupConfiguration": {
        "groupEntityType": "MyCorp::Group"
```

```
    }  
  }  
},  
"policyStoreId": "PSEXAMPLEEabcdefg111111",  
"principalEntityType": "MyCorp::User",  
"clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN11111111"  
}
```

Sample Response

```
HTTP/1.1 200 OK  
Date: Tue, 13 Jun 2023 20:00:59 GMT  
Content-Type: application/x-amz-json-1.0  
Content-Length: <PayloadSizeBytes>  
vary: origin  
vary: access-control-request-method  
vary: access-control-request-headers  
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111  
Connection: keep-alive  
  
{  
  "createdDate": "2023-05-19T20:30:28.214829Z",  
  "identitySourceId": "ISEXAMPLEEabcdefg111111",  
  "lastUpdatedDate": "2023-05-19T20:30:28.214829Z",  
  "policyStoreId": "PSEXAMPLEEabcdefg111111"  
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)

- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

CreatePolicy

Creates a Cedar policy and saves it in the specified policy store. You can create either a static policy or a policy linked to a policy template.

- To create a static policy, provide the Cedar policy text in the `StaticPolicy` section of the `PolicyDefinition`.
- To create a policy that is dynamically linked to a policy template, specify the policy template ID and the principal and resource to associate with this policy in the `templateLinked` section of the `PolicyDefinition`. If the policy template is ever updated, any policies linked to the policy template automatically use the updated template.

Note

Creating a policy causes it to be validated against the schema in the policy store. If the policy doesn't pass validation, the operation fails and the policy isn't stored.

Note

Verified Permissions is [eventually consistent](#). It can take a few seconds for a new or changed element to propagate through the service and be visible in the results of other Verified Permissions operations.

Request Syntax

```
{
  "clientToken": "string",
  "definition": { ... },
  "policyStoreId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

definition

A structure that specifies the policy type and content to use for the new policy. You must include either a static or a templateLinked element. The policy content must be written in the Cedar policy language.

Type: [PolicyDefinition](#) object

Note: This object is a Union. Only one member of this object can be specified or returned.

Required: Yes

policyStoreId

Specifies the `PolicyStoreId` of the policy store you want to store the policy in.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `[a-zA-Z0-9-]*`

Required: Yes

clientToken

Specifies a unique, case-sensitive ID that you provide to ensure the idempotency of the request. This lets you safely retry the request without accidentally performing the same operation a second time. Passing the same value to a later call to an operation requires that you also pass the same value for all other parameters. We recommend that you use a [UUID type of value](#).

If you don't provide this value, then AWS generates a random one for you.

If you retry the operation with the same `ClientToken`, but with different parameters, the retry fails with an `ConflictException` error.

Verified Permissions recognizes a `ClientToken` for eight hours. After eight hours, the next request with the same parameters performs the operation again regardless of the value of `ClientToken`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: [a-zA-Z0-9-]*

Required: No

Response Syntax

```
{
  "createdDate": "string",
  "lastUpdatedDate": "string",
  "policyId": "string",
  "policyStoreId": "string",
  "policyType": "string",
  "principal": {
    "entityId": "string",
    "entityType": "string"
  },
  "resource": {
    "entityId": "string",
    "entityType": "string"
  }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

createdDate

The date and time the policy was originally created.

Type: Timestamp

lastUpdatedDate

The date and time the policy was last updated.

Type: Timestamp

policyId

The unique ID of the new policy.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

policyStoreId

The ID of the policy store that contains the new policy.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

policyType

The policy type of the new policy.

Type: String

Valid Values: STATIC | TEMPLATE_LINKED

principal

The principal specified in the new policy's scope. This response element isn't present when `principal` isn't specified in the policy content.

Type: [EntityIdentifier](#) object

resource

The resource specified in the new policy's scope. This response element isn't present when the `resource` isn't specified in the policy content.

Type: [EntityIdentifier](#) object

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

ConflictException

The request failed because another request to modify a resource occurred at the same.

HTTP Status Code: 400

InternalServerErrorException

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ServiceQuotaExceededException

The request failed because it would cause a service quota to be exceeded.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example 1

The following example request creates a static policy with a policy scope that specifies both a principal and a resource. The response includes both the `Principal` and `Resource` elements because both were specified in the request policy scope.

Note

The JSON in the parameters of this operation are strings that can contain embedded quotation marks (") within the outermost quotation mark pair. This requires that you *stringify* the JSON object by preceding all embedded quotation marks with a backslash character (\") and combining all lines into a single text line with no line breaks. Example strings are displayed wrapped across multiple lines here for readability, but the operation requires the parameters be submitted as single line strings.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.CreatePolicy
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "definition": {
    "static": {
      "description": "Grant members of janeFriends UserGroup access to the
vacationFolder Album",
      "statement": "permit( principal in UserGroup::\"janeFriends\", action,
resource in Album::\"vacationFolder\" );"
    }
  },
  "policyStoreId": "PSEXAMPLEabcdefgh111111",
  "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN11111111"
```

```
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "policyId": "SPEXAMPLEabcdefg111111",
  "arn": "arn:aws:verifiedpermissions::123456789012:policy/PSEXAMPLEabcdefg111111/SPEXAMPLEabcdefg111111",
  "createdDate": "2023-05-16T20:33:01.730817Z",
  "lastUpdatedDate": "2023-05-16T20:33:01.730817Z",
  "policyType": "STATIC",
  "principal": {
    "entityId": "janeFriends",
    "entityType": "UserGroup"
  },
  "resource": {
    "entityId": "vacationFolder",
    "entityType": "Album"
  }
}
```

Example 2

The following example creates a static policy with a policy scope that identifies a specific resource but does not specify a principal. Therefore, the response does not include a `Principal` element.

Note

The JSON in the parameters of this operation are strings that can contain embedded quotation marks (") within the outermost quotation mark pair. This requires that you

stringify the JSON object by preceding all embedded quotation marks with a backslash character (\ ") and combining all lines into a single text line with no line breaks. Example strings are displayed wrapped across multiple lines here for readability, but the operation requires the parameters be submitted as single line strings.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.CreatePolicy
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "policyDefinition": {
    "staticPolicy": {
      "description": "Grant everyone of janeFriends UserGroup access to the
vacationFolder Album",
      "statement": "permit(principal, action, resource in Album::\"publicFolder
\");"
    }
  },
  "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive
```



```
{
  "arn": "arn:aws:verifiedpermissions::123456789012:policy/PSEXAMPLEEabcdefg111111/SPEXAMPLEEabcdefg222222",
  "createdDate": "2023-05-16T21:19:44.528576+00:00",
  "lastUpdatedDate": "2023-05-16T21:19:44.528576+00:00",
  "policyId": "SPEXAMPLEEabcdefg222222",
  "policyStoreId": "PSEXAMPLEEabcdefg111111",
  "policyType": "STATIC",
  "resource": {
    "entityId": "publicFolder",
    "entityType": "Album"
  }
}
```

Example 3

The following example creates a template-linked policy using the specified policy template and associates the specified principal to use with the new template-linked policy.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.CreatePolicy
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "definition": {
    "templateLinked": {
      "policyTemplateId": "PTEXAMPLEEabcdefg111111",
      "principal": {
        "entityType": "User",
        "entityId": "alice"
      }
    }
  },
  "policyStoreId": "PSEXAMPLEEabcdefg111111",
  "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN11111111"
```

```
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "policyStoreId": "PSEXAMPLEabcdefgh111111",
  "policyId": "TPEXAMPLEabcdefgh111111",
  "policyType": "TEMPLATELINKED",
  "principal": {
    "entityType": "User",
    "entityId": "alice"
  },
  "resource": {
    "entityType": "Photo",
    "entityId": "VacationPhoto94.jpg"
  },
  "createdDate": "2023-05-22T18:57:53.298278Z",
  "lastUpdatedDate": "2023-05-22T18:57:53.298278Z"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)

- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

CreatePolicyStore

Creates a policy store. A policy store is a container for policy resources.

Note

Although [Cedar supports multiple namespaces](#), Verified Permissions currently supports only one namespace per policy store.

Note

Verified Permissions is [eventually consistent](#). It can take a few seconds for a new or changed element to propagate through the service and be visible in the results of other Verified Permissions operations.

Request Syntax

```
{
  "clientToken": "string",
  "description": "string",
  "validationSettings": {
    "mode": "string"
  }
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

validationSettings

Specifies the validation setting for this policy store.

Currently, the only valid and required value is Mode.

Important

We recommend that you turn on STRICT mode only after you define a schema. If a schema doesn't exist, then STRICT mode causes any policy to fail validation, and Verified Permissions rejects the policy. You can turn off validation by using the [UpdatePolicyStore](#). Then, when you have a schema defined, use [UpdatePolicyStore](#) again to turn validation back on.

Type: [ValidationSettings](#) object

Required: Yes

clientToken

Specifies a unique, case-sensitive ID that you provide to ensure the idempotency of the request. This lets you safely retry the request without accidentally performing the same operation a second time. Passing the same value to a later call to an operation requires that you also pass the same value for all other parameters. We recommend that you use a [UUID type of value](#).

If you don't provide this value, then AWS generates a random one for you.

If you retry the operation with the same ClientToken, but with different parameters, the retry fails with an `ConflictException` error.

Verified Permissions recognizes a ClientToken for eight hours. After eight hours, the next request with the same parameters performs the operation again regardless of the value of ClientToken.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9-]*`

Required: No

description

Descriptive text that you can provide to help with identification of the current policy store.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 150.

Required: No

Response Syntax

```
{
  "arn": "string",
  "createdDate": "string",
  "lastUpdatedDate": "string",
  "policyStoreId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

arn

The Amazon Resource Name (ARN) of the new policy store.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2500.

Pattern: `arn:[^:]*:[^:]*:[^:]*:[^:]*:.*`

createdDate

The date and time the policy store was originally created.

Type: Timestamp

lastUpdatedDate

The date and time the policy store was last updated.

Type: Timestamp

policyStoreId

The unique ID of the new policy store.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

ConflictException

The request failed because another request to modify a resource occurred at the same.

HTTP Status Code: 400

InternalServerError

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ServiceQuotaExceededException

The request failed because it would cause a service quota to be exceeded.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example

The following example creates a new policy store with strict validation turned on.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.CreatePolicyStore
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "validationSettings": {"mode": "STRICT"},
  "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
```

```
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "policyStoreId":"PSEXAMPLEabcdefg111111",
  "arn":"arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111",
  "createdDate":"2023-05-16T17:41:29.103459Z",
  "lastUpdatedDate":"2023-05-16T17:41:29.103459Z"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

CreatePolicyTemplate

Creates a policy template. A template can use placeholders for the principal and resource. A template must be instantiated into a policy by associating it with specific principals and resources to use for the placeholders. That instantiated policy can then be considered in authorization decisions. The instantiated policy works identically to any other policy, except that it is dynamically linked to the template. If the template changes, then any policies that are linked to that template are immediately updated as well.

Note

Verified Permissions is *eventually consistent*. It can take a few seconds for a new or changed element to propagate through the service and be visible in the results of other Verified Permissions operations.

Request Syntax

```
{
  "clientToken": "string",
  "description": "string",
  "policyStoreId": "string",
  "statement": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

[policyStoreId](#)

The ID of the policy store in which to create the policy template.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

statement

Specifies the content that you want to use for the new policy template, written in the Cedar policy language.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 10000.

Required: Yes

clientToken

Specifies a unique, case-sensitive ID that you provide to ensure the idempotency of the request. This lets you safely retry the request without accidentally performing the same operation a second time. Passing the same value to a later call to an operation requires that you also pass the same value for all other parameters. We recommend that you use a [UUID type of value](#).

If you don't provide this value, then AWS generates a random one for you.

If you retry the operation with the same `ClientToken`, but with different parameters, the retry fails with an `ConflictException` error.

Verified Permissions recognizes a `ClientToken` for eight hours. After eight hours, the next request with the same parameters performs the operation again regardless of the value of `ClientToken`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: [a-zA-Z0-9-]*

Required: No

description

Specifies a description for the policy template.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 150.

Required: No

Response Syntax

```
{  
  "createdDate": "string",  
  "lastUpdatedDate": "string",  
  "policyStoreId": "string",  
  "policyTemplateId": "string"  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

createdDate

The date and time the policy template was originally created.

Type: Timestamp

lastUpdatedDate

The date and time the policy template was most recently updated.

Type: Timestamp

policyStoreId

The ID of the policy store that contains the policy template.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

policyTemplateId

The unique ID of the new policy template.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

ConflictException

The request failed because another request to modify a resource occurred at the same.

HTTP Status Code: 400

InternalServerError

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ServiceQuotaExceededException

The request failed because it would cause a service quota to be exceeded.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example

The following example creates a policy template that has a placeholder for the principal.

Note

The JSON in the parameters of this operation are strings that can contain embedded quotation marks (") within the outermost quotation mark pair. This requires that you *stringify* the JSON object by preceding all embedded quotation marks with a backslash character (\ ") and combining all lines into a single text line with no line breaks. Example strings are displayed wrapped across multiple lines here for readability, but the operation requires the parameters be submitted as single line strings.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.CreatePolicyTemplate
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
Signature=<Signature>
```



```
Content-Length: <PayloadSizeBytes>
```

```
{
  "description": "Template for research dept",
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "statement": "\"AccessVacation\"\\npermit(\\n  principal in ?principal,\\n  action == Action::\\\"view\\\",\\n  resource == Photo::\\\"VacationPhoto94.jpg\\\"\\n)\\nwhen {\\n  principal has department && principal.department == \\\"research\\\"\\n};",
  "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN11111111"}
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE111111
Connection: keep-alive
```

```
{
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "policyTemplateId": "PTEXAMPLEabcdefg111111",
  "createdDate": "2023-05-17T18:58:48.795411Z",
  "lastUpdatedDate": "2023-05-17T18:58:48.795411Z"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)

- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DeleteIdentitySource

Deletes an identity source that references an identity provider (IdP) such as Amazon Cognito. After you delete the identity source, you can no longer use tokens for identities from that identity source to represent principals in authorization queries made using [IsAuthorizedWithToken](#) operations.

Request Syntax

```
{
  "identitySourceId": "string",
  "policyStoreId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

[identitySourceId](#)

Specifies the ID of the identity source that you want to delete.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

[policyStoreId](#)

Specifies the ID of the policy store that contains the identity source that you want to delete.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

ConflictException

The request failed because another request to modify a resource occurred at the same.

HTTP Status Code: 400

InternalServerErrorException

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example

The following example request deletes the specified identity source.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.DeleteIdentitySource
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "identitySourceId": "ISEXAMPLEabcdefgh111111",
  "policyStoreId": "PSEXAMPLEabcdefgh111111"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
```

```
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{ }
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DeletePolicy

Deletes the specified policy from the policy store.

This operation is idempotent; if you specify a policy that doesn't exist, the request response returns a successful HTTP 200 status code.

Request Syntax

```
{
  "policyId": "string",
  "policyStoreId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

policyId

Specifies the ID of the policy that you want to delete.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

policyStoreId

Specifies the ID of the policy store that contains the policy that you want to delete.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

ConflictException

The request failed because another request to modify a resource occurred at the same.

HTTP Status Code: 400

InternalServerErrorException

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example

The following example deletes the specified policy from its policy store.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.DeletePolicy
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "policyId": "SPEXAMPLEabcdefghijklmnop111111",
  "policyStoreId": "PSEXAMPLEabcdefghijklmnop111111"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
```

```
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{ }
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DeletePolicyStore

Deletes the specified policy store.

This operation is idempotent. If you specify a policy store that does not exist, the request response will still return a successful HTTP 200 status code.

Request Syntax

```
{  
  "policyStoreId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

policyStoreId

Specifies the ID of the policy store that you want to delete.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerError

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example

The following example deletes the specified policy store.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
```

```
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.DeletePolicyStore
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "policyStoreId": "PSEXAMPLEabcdefghijklmnop111111"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)

- [AWS SDK for Ruby V3](#)

DeletePolicyTemplate

Deletes the specified policy template from the policy store.

Important

This operation also deletes any policies that were created from the specified policy template. Those policies are immediately removed from all future API responses, and are asynchronously deleted from the policy store.

Request Syntax

```
{
  "policyStoreId": "string",
  "policyTemplateId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

policyStoreId

Specifies the ID of the policy store that contains the policy template that you want to delete.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

policyTemplateId

Specifies the ID of the policy template that you want to delete.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

ConflictException

The request failed because another request to modify a resource occurred at the same.

HTTP Status Code: 400

InternalServerError

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its

value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example

The following example deletes a policy template. Before you can perform this operation, you must first delete any template-linked policies that were instantiated from this policy template. To delete them, use [DeletePolicy](#).

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.DeletePolicyTemplate
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "policyStoreId": "PSEXAMPLEabcdefgh111111",
  "policyTemplateId": "PTEXAMPLEabcdefgh111111"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive
```

```
{}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

GetIdentitySource

Retrieves the details about the specified identity source.

Request Syntax

```
{
  "identitySourceId": "string",
  "policyStoreId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

identitySourceId

Specifies the ID of the identity source you want information about.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

policyStoreId

Specifies the ID of the policy store that contains the identity source you want information about.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

Response Syntax

```
{
  "configuration": { ... },
  "createdDate": "string",
  "details": {
    "clientIds": [ "string" ],
    "discoveryUrl": "string",
    "openIdIssuer": "string",
    "userPoolArn": "string"
  },
  "identitySourceId": "string",
  "lastUpdatedDate": "string",
  "policyStoreId": "string",
  "principalEntityType": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

createdDate

The date and time that the identity source was originally created.

Type: Timestamp

identitySourceId

The ID of the identity source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

lastUpdatedDate

The date and time that the identity source was most recently updated.

Type: Timestamp

policyStoreId

The ID of the policy store that contains the identity source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

principalEntityType

The data type of principals generated for identities authenticated by this identity source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: .*

configuration

Contains configuration information about an identity source.

Type: [ConfigurationDetail](#) object

Note: This object is a Union. Only one member of this object can be specified or returned.

details

This parameter has been deprecated.

A structure that describes the configuration of the identity source.

Type: [IdentitySourceDetails](#) object

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerErrorException

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example

The following example retrieves the details for the specified identity source.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.GetIdentitySource
User-Agent: <UserAgentString>
```

```
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "identitySourceId": "ISEXAMPLEEabcdefg111111",
  "policyStoreId": "PSEXAMPLEEabcdefg111111"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "createdDate": "2023-05-19T20:30:28.173926Z",
  "details": {
    "clientIds": [ "a1b2c3d4e5f6g7h8i9j0kalbmc" ],
    "userPoolArn": "arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-
east-1_1a2b3c4d5",
    "discoveryUrl": "https://cognito-idp.us-east-1.amazonaws.com/us-
east-1_1a2b3c4d5",
    "openIdIssuer": "COGNITO"
  },
  "identitySourceId": "ISEXAMPLEEabcdefg111111",
  "lastUpdatedDate": "2023-05-22T20:45:59.962216Z",
  "policyStoreId": "PSEXAMPLEEabcdefg111111",
  "principalEntityType": "MyCorp::User",
  "configuration": {
    "cognitoUserPoolConfiguration": {
      "userPoolArn": "arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-
east-1_1a2b3c4d5",
      "clientIds": [],
      "issuer": "https://cognito-idp.us-east-1.amazonaws.com/us-
east-1_1a2b3c4d5",
      "groupConfiguration": {
```

```
        "groupEntityType": "MyCorp::Group"  
      }  
    }  
  }  
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

GetPolicy

Retrieves information about the specified policy.

Request Syntax

```
{  
  "policyId": "string",  
  "policyStoreId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

policyId

Specifies the ID of the policy you want information about.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

policyStoreId

Specifies the ID of the policy store that contains the policy that you want information about.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

Response Syntax

```
{
  "createdDate": "string",
  "definition": { ... },
  "lastUpdatedDate": "string",
  "policyId": "string",
  "policyStoreId": "string",
  "policyType": "string",
  "principal": {
    "entityId": "string",
    "entityType": "string"
  },
  "resource": {
    "entityId": "string",
    "entityType": "string"
  }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

createdDate

The date and time that the policy was originally created.

Type: Timestamp

definition

The definition of the requested policy.

Type: [PolicyDefinitionDetail](#) object

Note: This object is a Union. Only one member of this object can be specified or returned.

lastUpdatedDate

The date and time that the policy was last updated.

Type: Timestamp

[policyId](#)

The unique ID of the policy that you want information about.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

[policyStoreId](#)

The ID of the policy store that contains the policy that you want information about.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

[policyType](#)

The type of the policy.

Type: String

Valid Values: STATIC | TEMPLATE_LINKED

[principal](#)

The principal specified in the policy's scope. This element isn't included in the response when `Principal` isn't present in the policy content.

Type: [EntityIdentifier](#) object

[resource](#)

The resource specified in the policy's scope. This element isn't included in the response when `Resource` isn't present in the policy content.

Type: [EntityIdentifier](#) object

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerErrorException

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example

The following example retrieves information about the specified policy contained in the specified policy store. In this example, the requested policy is a template-linked policy, so it returns the ID of the policy template, and the specific principal and resource used by this policy.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.GetPolicy
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "policyId": "SPEXAMPLEabcdefg111111",
  "arn": "arn:aws:verifiedpermissions::123456789012:policy/PSEXAMPLEabcdefg111111/SPEXAMPLEabcdefg111111",
  "policyType": "STATIC",
  "resource": {
    "entityId": "publicFolder",
    "entityType": "Album"
  },
  "definition": {
    "static": {
      "description": "Grant everyone of janeFriends UserGroup access to the vacationFolder Album",
      "statement": "permit(principal, action, resource in Album::\"publicFolder\");"
    }
  },
  "createdDate": "2023-05-16T20:33:01.730817Z",
```

```
"lastUpdatedDate": "2023-05-16T20:33:01.730817Z"  
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

GetPolicyStore

Retrieves details about a policy store.

Request Syntax

```
{  
  "policyStoreId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

policyStoreId

Specifies the ID of the policy store that you want information about.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

Response Syntax

```
{  
  "arn": "string",  
  "createdDate": "string",  
  "description": "string",  
  "lastUpdatedDate": "string",  
}
```

```
"policyStoreId": "string",
"validationSettings": {
  "mode": "string"
}
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

arn

The Amazon Resource Name (ARN) of the policy store.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2500.

Pattern: `arn:[^:]*:[^:]*:[^:]*:[^:]*:.*`

createdDate

The date and time that the policy store was originally created.

Type: Timestamp

lastUpdatedDate

The date and time that the policy store was last updated.

Type: Timestamp

policyStoreId

The ID of the policy store;

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `[a-zA-Z0-9-]*`

validationSettings

The current validation settings for the policy store.

Type: [ValidationSettings](#) object

description

Descriptive text that you can provide to help with identification of the current policy store.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 150.

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerError

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example

The following example retrieves details about the specified policy store.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.GetPolicyStore
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "policyStoreId": "PSEXAMPLEabcdefgh111111"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "policyStoreId": "PSEXAMPLEabcdefgh111111",
}
```

```
"arn": "arn:aws:verifiedpermissions::123456789012:policy-store/  
PSEXAMPLEabcdefgh111111",  
  "validationSettings": {"mode":"STRICT"},  
  "createdDate": "2023-05-17T16:20:22.75472Z",  
  "lastUpdatedDate": "2023-05-17T16:20:22.75472Z"  
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

GetPolicyTemplate

Retrieve the details for the specified policy template in the specified policy store.

Request Syntax

```
{  
  "policyStoreId": "string",  
  "policyTemplateId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

policyStoreId

Specifies the ID of the policy store that contains the policy template that you want information about.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

policyTemplateId

Specifies the ID of the policy template that you want information about.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

Response Syntax

```
{
  "createdDate": "string",
  "description": "string",
  "lastUpdatedDate": "string",
  "policyStoreId": "string",
  "policyTemplateId": "string",
  "statement": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

createdDate

The date and time that the policy template was originally created.

Type: Timestamp

lastUpdatedDate

The date and time that the policy template was most recently updated.

Type: Timestamp

policyStoreId

The ID of the policy store that contains the policy template.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

policyTemplateId

The ID of the policy template.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

statement

The content of the body of the policy template written in the Cedar policy language.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 10000.

description

The description of the policy template.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 150.

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerErrorException

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more

information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example

The following example displays the details of the specified policy template.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.GetPolicyTemplate
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>
```

```
{
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "policyTemplateId": "PTEXAMPLEabcdefg111111"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "policyTemplateId": "PTEXAMPLEabcdefg111111",
  "description": "Template for research dept",
  "statement": "\"ResearchAccess\"\n\npermit(\n  principal in ?principal,\n  action == Action::\"view\", \n  resource in ?resource)\n\nwhen {\n  principal has\n  department && principal.department == \"research\"\n};",
  "createdDate": "2023-05-17T18:58:48.795411Z",
  "lastUpdatedDate": "2023-05-17T18:58:48.795411Z"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)

- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

GetSchema

Retrieve the details for the specified schema in the specified policy store.

Request Syntax

```
{  
  "policyStoreId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

policyStoreId

Specifies the ID of the policy store that contains the schema.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

Response Syntax

```
{  
  "createdDate": "string",  
  "lastUpdatedDate": "string",  
  "namespaces": [ "string" ],  
}
```

```
"policyStoreId": "string",  
"schema": "string"  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

createdDate

The date and time that the schema was originally created.

Type: Timestamp

lastUpdatedDate

The date and time that the schema was most recently updated.

Type: Timestamp

policyStoreId

The ID of the policy store that contains the schema.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

schema

The body of the schema, written in Cedar schema JSON.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100000.

namespaces

The namespaces of the entities referenced by this schema.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: . *

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerError

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example

The following example retrieves the current schema stored in the specified policy store.

Note

The JSON in the parameters of this operation are strings that can contain embedded quotation marks (") within the outermost quotation mark pair. This requires that you *stringify* the JSON object by preceding all embedded quotation marks with a backslash character (\ ") and combining all lines into a single text line with no line breaks. Example strings are displayed wrapped across multiple lines here for readability, but the operation requires the parameters be submitted as single line strings.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.GetSchema
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "policyStoreId": "PSEXAMPLEabcdefghijklmnop111111"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
```

```
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "schema": "{
    \"My::Application\": {
      \"actions\": {
        \"remoteAccess\": {
          \"appliesTo\": {
            \"principalTypes\": [\"Employee\"]
          }
        }
      },
      \"entityTypes\": {
        \"Employee\": {
          \"shape\": {
            \"attributes\": {
              \"jobLevel\": { \"type\": \"Long\" },
              \"name\": { \"type\": \"String\" }
            },
            \"type\": \"Record\"
          }
        }
      }
    }
  }",
  "createdDate": "2023-05-18T14:46:35.020571Z",
  "lastUpdatedDate": "2023-05-23T16:48:20.95041Z"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)

- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

IsAuthorized

Makes an authorization decision about a service request described in the parameters. The information in the parameters can also define additional context that Verified Permissions can include in the evaluation. The request is evaluated against all matching policies in the specified policy store. The result of the decision is either Allow or Deny, along with a list of the policies that resulted in the decision.

Request Syntax

```
{
  "action": {
    "actionId": "string",
    "actionType": "string"
  },
  "context": { ... },
  "entities": { ... },
  "policyStoreId": "string",
  "principal": {
    "entityId": "string",
    "entityType": "string"
  },
  "resource": {
    "entityId": "string",
    "entityType": "string"
  }
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

policyStoreId

Specifies the ID of the policy store. Policies in this policy store will be used to make an authorization decision for the input.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

action

Specifies the requested action to be authorized. For example, is the principal authorized to perform this action on the resource?

Type: [ActionIdentifier](#) object

Required: No

context

Specifies additional context that can be used to make more granular authorization decisions.

Type: [ContextDefinition](#) object

Note: This object is a Union. Only one member of this object can be specified or returned.

Required: No

entities

Specifies the list of resources and principals and their associated attributes that Verified Permissions can examine when evaluating the policies.

Note

You can include only principal and resource entities in this parameter; you can't include actions. You must specify actions in the schema.

Type: [EntitiesDefinition](#) object

Note: This object is a Union. Only one member of this object can be specified or returned.

Required: No

principal

Specifies the principal for which the authorization decision is to be made.

Type: [EntityIdentifier](#) object

Required: No

resource

Specifies the resource for which the authorization decision is to be made.

Type: [EntityIdentifier](#) object

Required: No

Response Syntax

```
{
  "decision": "string",
  "determiningPolicies": [
    {
      "policyId": "string"
    }
  ],
  "errors": [
    {
      "errorDescription": "string"
    }
  ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

decision

An authorization decision that indicates if the authorization request should be allowed or denied.

Type: String

Valid Values: ALLOW | DENY

determiningPolicies

The list of determining policies used to make the authorization decision. For example, if there are two matching policies, where one is a forbid and the other is a permit, then the forbid policy will be the determining policy. In the case of multiple matching permit policies then there would be multiple determining policies. In the case that no policies match, and hence the response is DENY, there would be no determining policies.

Type: Array of [DeterminingPolicyItem](#) objects

errors

Errors that occurred while making an authorization decision, for example, a policy references an Entity or entity Attribute that does not exist in the slice.

Type: Array of [EvaluationErrorItem](#) objects

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerError

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its

value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example 1

The following example requests an authorization decision for a principal of type `User` named Alice, who wants to perform the `updatePhoto` operation, on a resource of type `Photo` named `VacationPhoto94.jpg`.

The response shows that the request was allowed by one policy.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.IsAuthorized
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
```

```
"principal": {
  "entityType": "PhotoFlash::User",
  "entityId": "alice"
},
"action": {
  "actionType": "Action",
  "actionId": "view"
},
"resource": {
  "entityType": "PhotoFlash::Photo",
  "entityId": "VacationPhoto94.jpg"
},
"policyStoreId": "PSEXAMPLEEabcdefg111111"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "decision": "ALLOW",
  "determiningPolicies": [
    {
      "policyId": "SPEXAMPLEEabcdefg111111"
    }
  ],
  "errors": []
}
```

Example 2

The following example is a more extensive request for an authorization decision for a principal of type User named alice, who wants to perform the updatePhoto operation, on a resource of type Photo named VacationPhoto94.jpg. The request includes the optional entities

element, that specifies that the photo is contained in an Album named `alice_folder`. Those additional entities and their attributes can be referenced and checked by conditional elements in the policies in the specified policy store.

In this example, the policy that permits access specifies that `User::"alice"` is allowed to update photos in the folder `Album::"alice_folder"`

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.IsAuthorized
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>
```

```
{
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "principal": {
    "entityType": "PhotoFlash::User",
    "entityId": "alice"
  },
  "action": {
    "actionType": "Action",
    "actionId": "updatePhoto"
  },
  "resource": {
    "entityType": "PhotoFlash::Photo",
    "entityId": "VacationPhoto94.jpg"
  },
  "entities": {
    "entityList": [
      {
        "identifier": {
          "entityType": "PhotoFlash::Photo",
          "entityId": "VacationPhoto94.jpg"
        },
        "attributes": {},
        "parents": [
          {
```



```
        "entityType": "PhotoFlash::Album",
        "entityId": "alice_folder"
      }
    ],
    {
      "identifier": {
        "entityType": "PhotoFlash::Album",
        "entityId": "alice_folder"
      }
    }
  ]
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "determiningPolicies": [
    {
      "determiningPolicyId": "SPEXAMPLEabcdefg111111"
    }
  ],
  "decision": "ALLOW",
  "errors": []
}
```

Example 3

The following example is the same as the previous example, except that the principal is `User::"bob"`, and the policy store doesn't contain any policy that allows that user access to

Album::"alice_folder". The output infers that the Deny was implicit because the list of `DeterminingPolicies` is empty.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.IsAuthorized
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "principal": {
    "entityType": "PhotoFlash::User",
    "entityId": "bob"
  },
  "action": {
    "actionType": "Action",
    "actionId": "view"
  },
  "resource": {
    "entityType": "PhotoFlash::Photo",
    "entityId": "VacationPhoto94.jpg"
  },
  "policyStoreId": "PSEXAMPLEabcdefghijklmnop111111"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive
```

```
{
  "decision": "DENY",
  "determiningPolicies": [],
  "errors": []
}
```

Example 4

The following example relies on the `DigitalPetStore` sample app and the policy `Customer Role - Get Order`. To satisfy this policy, Alice must be in the `Customer` role and the `Order` must have Alice in the `owner` attribute. To define Alice and the order with these properties, we must dive deeper into the `entities` element and declare that Alice is a customer, and the order is owned by Alice. Because Alice is the customer who placed the order, Verified Permissions returns an `ALLOW` decision to their request for the action `DigitalPetStore::GetOrder`.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.IsAuthorized
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>
```

```
    {
      "principal": {
        "entityType": "DigitalPetStore::User",
        "entityId": "Alice"
      },
      "action": {
        "actionType": "DigitalPetStore::Action",
        "actionId": "GetOrder"
      },
      "resource": {
        "entityType": "DigitalPetStore::Order",
        "entityId": "1234"
      },
      "entities": {
        "entityList": [
```

```
{
  "identifier": {
    "entityType": "DigitalPetStore::User",
    "entityId": "Alice"
  },
  "attributes": {
    "memberId": {
      "string": "5cad60b9-209c-46d6-bfb7-536c341634ca"
    }
  },
  "parents": [
    {
      "entityType": "DigitalPetStore::Role",
      "entityId": "Customer"
    }
  ]
},
{
  "identifier": {
    "entityType": "DigitalPetStore::Order",
    "entityId": "1234"
  },
  "attributes": {
    "owner": {
      "entityIdentifier": {
        "entityType": "DigitalPetStore::User",
        "entityId": "Alice"
      }
    }
  },
  "parents": []
}
],
"policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
```

```
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "decision": "ALLOW",
  "determiningPolicies": [
    {
      "policyId": "SPEXAMPLEabcdefg111111"
    }
  ],
  "errors": []
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

IsAuthorizedWithToken

Makes an authorization decision about a service request described in the parameters. The principal in this request comes from an external identity source in the form of an identity token formatted as a [JSON web token \(JWT\)](#). The information in the parameters can also define additional context that Verified Permissions can include in the evaluation. The request is evaluated against all matching policies in the specified policy store. The result of the decision is either Allow or Deny, along with a list of the policies that resulted in the decision.

At this time, Verified Permissions accepts tokens from only Amazon Cognito.

Verified Permissions validates each token that is specified in a request by checking its expiration date and its signature.

Important

If you delete a Amazon Cognito user pool or user, tokens from that deleted pool or that deleted user continue to be usable until they expire.

Request Syntax

```
{
  "accessToken": "string",
  "action": {
    "actionId": "string",
    "actionType": "string"
  },
  "context": { ... },
  "entities": { ... },
  "identityToken": "string",
  "policyStoreId": "string",
  "resource": {
    "entityId": "string",
    "entityType": "string"
  }
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

policyStoreId

Specifies the ID of the policy store. Policies in this policy store will be used to make an authorization decision for the input.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `[a-zA-Z0-9-]*`

Required: Yes

accessToken

Specifies an access token for the principal to be authorized. This token is provided to you by the identity provider (IdP) associated with the specified identity source. You must specify either an `accessToken`, an `identityToken`, or both.

Must be an access token. Verified Permissions returns an error if the `token_use` claim in the submitted token isn't `access`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 131072.

Pattern: `[A-Za-z0-9-_=]+.[A-Za-z0-9-_=]+.[A-Za-z0-9-_=]+`

Required: No

action

Specifies the requested action to be authorized. Is the specified principal authorized to perform this action on the specified resource.

Type: [ActionIdentifier](#) object

Required: No

[context](#)

Specifies additional context that can be used to make more granular authorization decisions.

Type: [ContextDefinition](#) object

Note: This object is a Union. Only one member of this object can be specified or returned.

Required: No

[entities](#)

Specifies the list of resources and their associated attributes that Verified Permissions can examine when evaluating the policies.

Important

You can't include principals in this parameter, only resource and action entities. This parameter can't include any entities of a type that matches the user or group entity types that you defined in your identity source.

- The `IsAuthorizedWithToken` operation takes principal attributes from **only** the `identityToken` or `accessToken` passed to the operation.
- For action entities, you can include only their `Identifier` and `EntityType`.

Type: [EntitiesDefinition](#) object

Note: This object is a Union. Only one member of this object can be specified or returned.

Required: No

[identityToken](#)

Specifies an identity token for the principal to be authorized. This token is provided to you by the identity provider (IdP) associated with the specified identity source. You must specify either an `accessToken`, an `identityToken`, or `both`.

Must be an ID token. Verified Permissions returns an error if the `token_use` claim in the submitted token isn't `id`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 131072.

Pattern: `[A-Za-z0-9-_=]+.[A-Za-z0-9-_=]+.[A-Za-z0-9-_=]+`

Required: No

resource

Specifies the resource for which the authorization decision is made. For example, is the principal allowed to perform the action on the resource?

Type: [EntityIdentifier](#) object

Required: No

Response Syntax

```
{
  "decision": "string",
  "determiningPolicies": [
    {
      "policyId": "string"
    }
  ],
  "errors": [
    {
      "errorDescription": "string"
    }
  ],
  "principal": {
    "entityId": "string",
    "entityType": "string"
  }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

decision

An authorization decision that indicates if the authorization request should be allowed or denied.

Type: String

Valid Values: ALLOW | DENY

determiningPolicies

The list of determining policies used to make the authorization decision. For example, if there are multiple matching policies, where at least one is a forbid policy, then because forbid always overrides permit the forbid policies are the determining policies. If all matching policies are permit policies, then those policies are the determining policies. When no policies match and the response is the default DENY, there are no determining policies.

Type: Array of [DeterminingPolicyItem](#) objects

errors

Errors that occurred while making an authorization decision. For example, a policy references an entity or entity attribute that does not exist in the slice.

Type: Array of [EvaluationErrorItem](#) objects

principal

The identifier of the principal in the ID or access token.

Type: [EntityIdentifier](#) object

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerError

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if . . . then . . . else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more

information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example

The following example requests an authorization decision for a user who was authenticated by Amazon Cognito. The request uses the identity token provided by Amazon Cognito instead of the access token. In this example, the specified information store is configured to return principals as entities of type `CognitoUser`. The policy store contains a policy with the following statement.

```
permit(  
  principal == CognitoUser::"us-east-1_1a2b3c4d5|a1b2c3d4e5f6g7h8i9j0ka1bmc",  
  action,  
  resource == PhotoFlash::Photo::"VacationPhoto94.jpg"  
);
```

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.IsAuthorizedWithToken
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "action": {
    "actionId": "View",
    "actionType": "Action"
  },
  "resource": {
    "entityId": "vacationPhoto94.jpg",
    "entityType": "PhotoFlash::Photo"
  }
  "identityToken": "AbCdE12345...long.string...54321EdCbA",
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "decision": "Allow",
  "determiningPolicies": [
    {
      "determiningPolicyId": "SPEXAMPLEabcdefg111111"
    }
  ],
}
```

```
"errors": []  
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListIdentitySources

Returns a paginated list of all of the identity sources defined in the specified policy store.

Request Syntax

```
{
  "filters": [
    {
      "principalEntityType": "string"
    }
  ],
  "maxResults": number,
  "nextToken": "string",
  "policyStoreId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

policyStoreId

Specifies the ID of the policy store that contains the identity sources that you want to list.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

filters

Specifies characteristics of an identity source that you can use to limit the output to matching identity sources.

Type: Array of [IdentitySourceFilter](#) objects

Array Members: Minimum number of 0 items. Maximum number of 10 items.

Required: No

maxResults

Specifies the total number of results that you want included in each response. If additional items exist beyond the number you specify, the `NextToken` response element is returned with a value (not null). Include the specified value as the `NextToken` request parameter in the next call to the operation to get the next set of results. Note that the service might return fewer results than the maximum even when there are more results available. You should check `NextToken` after every operation to ensure that you receive all of the results.

If you do not specify this parameter, the operation defaults to 10 identity sources per response. You can specify a maximum of 50 identity sources per response.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

nextToken

Specifies that you want to receive the next page of results. Valid only if you received a `NextToken` response in the previous request. If you did, it indicates that more output is available. Set this parameter to the value provided by the previous call's `NextToken` response to request the next page of results.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 8000.

Pattern: `[A-Za-z0-9-_/\.]*`

Required: No

Response Syntax

```
{
  "identitySources": [
    {
      "configuration": { ... },
      "createdDate": "string",
      "details": {
        "clientIds": [ "string" ],
        "discoveryUrl": "string",
        "openIdIssuer": "string",
        "userPoolArn": "string"
      },
      "identitySourceId": "string",
      "lastUpdatedDate": "string",
      "policyStoreId": "string",
      "principalEntityType": "string"
    }
  ],
  "nextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[identitySources](#)

The list of identity sources stored in the specified policy store.

Type: Array of [IdentitySourceItem](#) objects

[nextToken](#)

If present, this value indicates that more output is available than is included in the current response. Use this value in the NextToken request parameter in a subsequent call to the operation to get the next part of the output. You should repeat this until the NextToken response element comes back as null. This indicates that this is the last page of results.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 8000.

Pattern: `[A-Za-z0-9-_/+.]*`

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerError

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example

The following example request creates lists the identity sources currently defined in the specified policy store.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.ListIdentitySources
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "identitySources": [
    {
      "createdDate": "2023-05-19T20:29:23.66812Z",
      "details": {
        "clientIds": ["a1b2c3d4e5f6g7h8i9j0kalbmc"],
        "userPoolArn": "arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-
east-1_1a2b3c4d5",

```

```
        "discoveryUrl": "https://cognito-idp.us-east-1.amazonaws.com/us-
east-1_1a2b3c4d5",
        "openIdIssuer": "COGNITO"
    },
    "identitySourceId": "ISEXAMPLEEabcdefg111111",
    "lastUpdatedDate": "2023-05-19T20:29:23.66812Z",
    "policyStoreId": "PSEXAMPLEEabcdefg111111",
    "principalEntityType": "MyCorp::User",
    "configuration": {
        "cognitoUserPoolConfiguration": {
            "userPoolArn": "arn:aws:cognito-idp:us-east-1:123456789012:userpool/
us-east-1_1a2b3c4d5",
            "clientIds": [],
            "issuer": "https://cognito-idp.us-east-1.amazonaws.com/us-
east-1_1a2b3c4d5",
            "groupConfiguration": {
                "groupEntityType": "MyCorp::Group"
            }
        }
    }
}
]
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListPolicies

Returns a paginated list of all policies stored in the specified policy store.

Request Syntax

```
{
  "filter": {
    "policyTemplateId": "string",
    "policyType": "string",
    "principal": { ... },
    "resource": { ... }
  },
  "maxResults": number,
  "nextToken": "string",
  "policyStoreId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

policyStoreId

Specifies the ID of the policy store you want to list policies from.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

filter

Specifies a filter that limits the response to only policies that match the specified criteria. For example, you list only the policies that reference a specified principal.

Type: [PolicyFilter](#) object

Required: No

maxResults

Specifies the total number of results that you want included in each response. If additional items exist beyond the number you specify, the `NextToken` response element is returned with a value (not null). Include the specified value as the `NextToken` request parameter in the next call to the operation to get the next set of results. Note that the service might return fewer results than the maximum even when there are more results available. You should check `NextToken` after every operation to ensure that you receive all of the results.

If you do not specify this parameter, the operation defaults to 10 policies per response. You can specify a maximum of 50 policies per response.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

nextToken

Specifies that you want to receive the next page of results. Valid only if you received a `NextToken` response in the previous request. If you did, it indicates that more output is available. Set this parameter to the value provided by the previous call's `NextToken` response to request the next page of results.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 8000.

Pattern: `[A-Za-z0-9-_=+/\.]*`

Required: No

Response Syntax

```
{
  "nextToken": "string",
  "policies": [
    {
      "createdDate": "string",
      "definition": { ... },
      "lastUpdatedDate": "string",
      "policyId": "string",
      "policyStoreId": "string",
      "policyType": "string",
      "principal": {
        "entityId": "string",
        "entityType": "string"
      },
      "resource": {
        "entityId": "string",
        "entityType": "string"
      }
    }
  ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

policies

Lists all policies that are available in the specified policy store.

Type: Array of [PolicyItem](#) objects

nextToken

If present, this value indicates that more output is available than is included in the current response. Use this value in the NextToken request parameter in a subsequent call to the operation to get the next part of the output. You should repeat this until the NextToken response element comes back as null. This indicates that this is the last page of results.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 8000.

Pattern: `[A-Za-z0-9-_=+/\.]*`

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerError

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example 1

The following example lists all policies in the policy store.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.ListPolicies
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "policyStoreId": "PSEXAMPLEabcdefgh111111"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "policies": [
    {
      "arn": "arn:aws:verifiedpermissions::123456789012:policy/
PSEXAMPLEabcdefgh111111/SPEXAMPLEabcdefgh111111",
      "createdDate": "2023-05-16T21:12:52.882422+00:00",
      "definition": {
        "static": {
```

```

        "description": "Grant members of janeFriends UserGroup access to
the vacationFolder Album"
    }
},
"lastUpdatedDate": "2023-05-16T21:12:52.882422+00:00",
"policyId": "SPEXAMPLEabcdefg111111",
"policyStoreId": "PSEXAMPLEabcdefg111111",
"policyType": "STATIC",
"principal": {
    "entityId": "janeFriends",
    "entityType": "UserGroup"
},
"resource": {
    "entityId": "vacationFolder",
    "entityType": "Album"
}
},
{
    "arn": "arn:aws:verifiedpermissions::123456789012:policy/
PSEXAMPLEabcdefg111111/SPEXAMPLEabcdefg222222",
    "createdDate": "2023-05-16T21:19:44.528576+00:00",
    "definition": {
        "static": {
            "description": "Grant everyone access to the publicFolder Album"
        }
    },
    "lastUpdatedDate": "2023-05-16T21:19:44.528576+00:00",
    "policyId": "SPEXAMPLEabcdefg222222",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyType": "STATIC",
    "resource": {
        "entityId": "publicFolder",
        "entityType": "Album"
    }
}
]
}

```

Example 2

The following example lists all policies for a specified principal.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.ListPolicies
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "filter": {
    "principal": {
      "id": {
        "entityType": "User",
        "entityId": "alice"
      }
    }
  }
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "policies": [
    {
      "policyStoreId": "ps-f0ff7596-a721-4df2-8c04-45bcb8e12ccd",
      "policyId": "ip-376c8292-968e-48fb-8b77-5f695e8be789",
      "arn": "arn:aws:verifiedpermissions:123456789012::policy/ps-f0ff7596-
a721-4df2-8c04-45bcb8e12ccd/ip-376c8292-968e-48fb-8b77-5f695e8be789",
      "policyType": "STATIC",
```

```

    "principal": {
      "entityType": "User",
      "entityId": "alice"
    },
    "resource": {
      "entityType": "Album",
      "entityId": "bob_folder"
    },
    "policyDefinition": {
      "static": {
        "description": "An example policy"
      }
    },
    "createdDate": "2022-12-09T22:55:16.067533Z",
    "lastUpdatedDate": "2022-12-09T22:55:16.067533Z"
  },
  {
    "policyStoreId": "ps-f0ff7596-a721-4df2-8c04-45bcb8e12ccd",
    "policyId": "ip-9faa0844-24a0-4d81-a937-0ad7b155750a",
    "arn": "arn:aws:verifiedpermissions:123456789012::policy/ps-f0ff7596-
a721-4df2-8c04-45bcb8e12ccd/ip-9faa0844-24a0-4d81-a937-0ad7b155750a",
    "policyType": "STATIC",
    "principal": {
      "entityType": "User",
      "entityId": "alice"
    },
    "resource": {
      "entityType": "Album",
      "entityId": "alice_folder"
    },
    "policyDefinition": {
      "static": {}
    },
    "createdDate": "2022-12-09T23:00:24.66266Z",
    "lastUpdatedDate": "2022-12-09T23:00:24.66266Z"
  }
]
}

```

Example 3

The following example uses the `Filter` parameter to list only the template-linked policies in the specified policy store.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.ListPolicies
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "filter": {
    "policyType": "TEMPLATELINKED"
  }
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "policies": [{
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyId": "TPEXAMPLEabcdefg111111",
    "arn": "arn:aws:verifiedpermissions:us-east-1:123456789012:policy/
PSEXAMPLEabcdefg111111/TPEXAMPLEabcdefg111111",
    "policyType": "TEMPLATELINKED",
    "principal": {
      "entityType": "User",
      "entityId": "alice"
    },
    "resource": {
      "entityType": "Photo",
```

```
    "entityId": "pic.jpg"
  },
  "policyDefinition": {
    "templateLinked": {
      "policyTemplateId": "PTEXAMPLEEabcdefg111111",
      "principal": {
        "entityType": "User",
        "entityId": "alice"
      },
      "resource": {
        "entityType": "Photo",
        "entityId": "pic.jpg"
      }
    }
  },
  "createdDate": "2023-06-13T16:03:07.620867Z",
  "lastUpdatedDate": "2023-06-13T16:03:07.620867Z"
}]
}
```

Example 4

The following example uses the `Filter` parameter to list only those policies that were instantiated from the specified policy template.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.ListPolicies
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "filter": {
    "policyTemplateId": "PTEXAMPLEEabcdefg111111"
  }
}
```


Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "policies": [{
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyId": "TPEXAMPLEabcdefg111111",
    "arn": "arn:aws:verifiedpermissions::128716708097:policy/
PSEXAMPLEabcdefg111111/TPEXAMPLEabcdefg111111",
    "policyType": "TEMPLATELINKED",
    "principal": {
      "entityType": "User",
      "entityId": "alice"
    },
    "resource": {
      "entityType": "Photo",
      "entityId": "pic.jpg"
    },
    "policyDefinition": {
      "templateLinked": {
        "policyTemplateId": "pt-e42e3eee-8cbc-4af6-a187-4c94773ec89b",
        "principal": {
          "entityType": "User",
          "entityId": "alice"
        },
        "resource": {
          "entityType": "Photo",
          "entityId": "pic.jpg"
        }
      }
    },
    "createdDate": "2023-03-15T16:03:07.620867Z",
    "lastUpdatedDate": "2023-03-15T16:03:07.620867Z"
  }]
}
```

```
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListPolicyStores

Returns a paginated list of all policy stores in the calling AWS account.

Request Syntax

```
{
  "maxResults": number,
  "nextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

maxResults

Specifies the total number of results that you want included in each response. If additional items exist beyond the number you specify, the `NextToken` response element is returned with a value (not null). Include the specified value as the `NextToken` request parameter in the next call to the operation to get the next set of results. Note that the service might return fewer results than the maximum even when there are more results available. You should check `NextToken` after every operation to ensure that you receive all of the results.

If you do not specify this parameter, the operation defaults to 10 policy stores per response. You can specify a maximum of 50 policy stores per response.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

[nextToken](#)

Specifies that you want to receive the next page of results. Valid only if you received a NextToken response in the previous request. If you did, it indicates that more output is available. Set this parameter to the value provided by the previous call's NextToken response to request the next page of results.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 8000.

Pattern: [A-Za-z0-9-_/+=+\.\.]*

Required: No

Response Syntax

```
{
  "nextToken": "string",
  "policyStores": [
    {
      "arn": "string",
      "createdDate": "string",
      "description": "string",
      "lastUpdatedDate": "string",
      "policyStoreId": "string"
    }
  ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[policyStores](#)

The list of policy stores in the account.

Type: Array of [PolicyStoreItem](#) objects

nextToken

If present, this value indicates that more output is available than is included in the current response. Use this value in the `NextToken` request parameter in a subsequent call to the operation to get the next part of the output. You should repeat this until the `NextToken` response element comes back as `null`. This indicates that this is the last page of results.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 8000.

Pattern: `[A-Za-z0-9-_/+. \.]*`

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerError

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example

The following example lists all policy stores in the AWS account in the AWS Region in which you call the operation.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.ListPolicyStores
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "policyStores": [
    {
```

```
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "arn":"arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111",
    "createdDate":"2023-05-16T17:41:29.103459Z"
  },
  {
    "policyStoreId":"PSEXAMPLEabcdefg222222",
    "arn":"arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg222222",
    "createdDate":"2023-05-16T18:23:04.985521Z"
  }
]
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListPolicyTemplates

Returns a paginated list of all policy templates in the specified policy store.

Request Syntax

```
{  
  "maxResults": number,  
  "nextToken": "string",  
  "policyStoreId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

policyStoreId

Specifies the ID of the policy store that contains the policy templates you want to list.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

maxResults

Specifies the total number of results that you want included in each response. If additional items exist beyond the number you specify, the NextToken response element is returned with a value (not null). Include the specified value as the NextToken request parameter in

the next call to the operation to get the next set of results. Note that the service might return fewer results than the maximum even when there are more results available. You should check `NextToken` after every operation to ensure that you receive all of the results.

If you do not specify this parameter, the operation defaults to 10 policy templates per response. You can specify a maximum of 50 policy templates per response.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

nextToken

Specifies that you want to receive the next page of results. Valid only if you received a `NextToken` response in the previous request. If you did, it indicates that more output is available. Set this parameter to the value provided by the previous call's `NextToken` response to request the next page of results.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 8000.

Pattern: `[A-Za-z0-9-_/+=\.\]*`

Required: No

Response Syntax

```
{
  "nextToken": "string",
  "policyTemplates": [
    {
      "createdDate": "string",
      "description": "string",
      "lastUpdatedDate": "string",
      "policyStoreId": "string",
      "policyTemplateId": "string"
    }
  ]
}
```

```
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[policyTemplates](#)

The list of the policy templates in the specified policy store.

Type: Array of [PolicyTemplateItem](#) objects

[nextToken](#)

If present, this value indicates that more output is available than is included in the current response. Use this value in the NextToken request parameter in a subsequent call to the operation to get the next part of the output. You should repeat this until the NextToken response element comes back as null. This indicates that this is the last page of results.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 8000.

Pattern: `[A-Za-z0-9-_=+/\.]*`

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerError

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more

information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example

The following example retrieves a list of all of the policy templates in the specified policy store.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.ListPolicyTemplates
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>
```

```
{
  "policyStoreId": "PSEXAMPLEabcdefgh111111"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE111111
Connection: keep-alive

{
  "policyTemplates": [
    {
      "createdDate": "2023-05-17T18:55:20.888033+00:00",
      "description": "Generic template",
      "lastUpdatedDate": "2023-05-17T18:55:20.888033+00:00",
      "policyStoreId": "PSEXAMPLEabcdefgh111111",
      "policyTemplateId": "PTEXAMPLEabcdefgh111111"
    },
    {
      "createdDate": "2023-05-17T18:58:48.795411+00:00",
      "description": "Template for research dept",
      "lastUpdatedDate": "2023-05-17T18:58:48.795411+00:00",
      "policyStoreId": "PSEXAMPLEabcdefgh111111",
      "policyTemplateId": "PTEXAMPLEabcdefgh222222"
    }
  ]
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)

- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

PutSchema

Creates or updates the policy schema in the specified policy store. The schema is used to validate any Cedar policies and policy templates submitted to the policy store. Any changes to the schema validate only policies and templates submitted after the schema change. Existing policies and templates are not re-evaluated against the changed schema. If you later update a policy, then it is evaluated against the new schema at that time.

Note

Verified Permissions is [eventually consistent](#). It can take a few seconds for a new or changed element to propagate through the service and be visible in the results of other Verified Permissions operations.

Request Syntax

```
{
  "definition": { ... },
  "policyStoreId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

definition

Specifies the definition of the schema to be stored. The schema definition must be written in Cedar schema JSON.

Type: [SchemaDefinition](#) object

Note: This object is a Union. Only one member of this object can be specified or returned.

Required: Yes

policyStoreId

Specifies the ID of the policy store in which to place the schema.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

Response Syntax

```
{
  "createdDate": "string",
  "lastUpdatedDate": "string",
  "namespaces": [ "string" ],
  "policyStoreId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

createdDate

The date and time that the schema was originally created.

Type: Timestamp

lastUpdatedDate

The date and time that the schema was last updated.

Type: Timestamp

namespaces

Identifies the namespaces of the entities referenced by this schema.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: .*

policyStoreId

The unique ID of the policy store that contains the schema.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

ConflictException

The request failed because another request to modify a resource occurred at the same.

HTTP Status Code: 400

InternalServerError

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ServiceQuotaExceededException

The request failed because it would cause a service quota to be exceeded.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example

The following example creates a new schema, or updates an existing schema, in the specified policy store. Note that the schema text is shown line wrapped for readability. You should submit the entire schema text as a single line of text.

Note

The JSON in the parameters of this operation are strings that can contain embedded quotation marks (") within the outermost quotation mark pair. This requires that you *stringify* the JSON object by preceding all embedded quotation marks with a backslash character (\ ") and combining all lines into a single text line with no line breaks. Example strings are displayed wrapped across multiple lines here for readability, but the operation requires the parameters be submitted as single line strings.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.PutSchema
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "definition": {"cedarJson": "{\"MySampleNamespace\": {\"actions\": {\"remoteAccess\": {
    \"appliesTo\": {\"principalTypes\": [\"Employee\"]}}},\"entityTypes\": {
    \"Employee\": {
      \"shape\": {\"attributes\": {\"jobLevel\": {\"type\": \"Long\"},\"name\": {
        \"type\": \"String\"}},\"type\": \"Record\"}}}}}"}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "createdDate": "2023-06-13T19:28:06.003726Z",
  "lastUpdatedDate": "2023-06-13T19:28:06.003726Z",
  "Namespaces": [
    "My::Sample::Namespace"
  ],
  "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

UpdateIdentitySource

Updates the specified identity source to use a new identity provider (IdP) source, or to change the mapping of identities from the IdP to a different principal entity type.

Note

Verified Permissions is [eventually consistent](#). It can take a few seconds for a new or changed element to propagate through the service and be visible in the results of other Verified Permissions operations.

Request Syntax

```
{
  "identitySourceId": "string",
  "policyStoreId": "string",
  "principalEntityType": "string",
  "updateConfiguration": { ... }
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

identitySourceId

Specifies the ID of the identity source that you want to update.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

policyStoreId

Specifies the ID of the policy store that contains the identity source that you want to update.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

updateConfiguration

Specifies the details required to communicate with the identity provider (IdP) associated with this identity source.

Note

At this time, the only valid member of this structure is a Amazon Cognito user pool configuration.

You must specify a `userPoolArn`, and optionally, a `ClientId`.

Type: [UpdateConfiguration](#) object

Note: This object is a Union. Only one member of this object can be specified or returned.

Required: Yes

principalEntityType

Specifies the data type of principals generated for identities authenticated by the identity source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: .*

Required: No

Response Syntax

```
{  
  "createdDate": "string",  
  "identitySourceId": "string",  
  "lastUpdatedDate": "string",  
  "policyStoreId": "string"  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

createdDate

The date and time that the updated identity source was originally created.

Type: Timestamp

identitySourceId

The ID of the updated identity source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

lastUpdatedDate

The date and time that the identity source was most recently updated.

Type: Timestamp

policyStoreId

The ID of the policy store that contains the updated identity source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

ConflictException

The request failed because another request to modify a resource occurred at the same.

HTTP Status Code: 400

InternalServerErrorException

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example

The following example updates the configuration of the specified identity source with a new configuration.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.UpdateIdentitySource
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "identitySourceId": "ISEXAMPLEabcdefg111111",
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "updateConfiguration": {
    "cognitoUserPoolConfiguration": {
      "userPoolArn": "arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-
east-1_1a2b3c4d5",
      "clientId": ["a1b2c3d4e5f6g7h8i9j0kalbmc"],
      "groupConfiguration": {
        "groupEntityType": "MyCorp::Group"
      }
    }
  },
  "principalEntityType": "MyCorp::User",
  "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN11111111"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
```

```
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "createdDate": "2023-05-19T20:30:28.173926Z",
  "identitySourceId": "ISEXAMPLEEabcdefg111111",
  "lastUpdatedDate": "2023-05-22T20:45:59.962216Z",
  "policyStoreId": "PSEXAMPLEEabcdefg111111"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

UpdatePolicy

Modifies a Cedar static policy in the specified policy store. You can change only certain elements of the [UpdatePolicyDefinition](#) parameter. You can directly update only static policies. To change a template-linked policy, you must update the template instead, using [UpdatePolicyTemplate](#).

Note

- If policy validation is enabled in the policy store, then updating a static policy causes Verified Permissions to validate the policy against the schema in the policy store. If the updated static policy doesn't pass validation, the operation fails and the update isn't stored.
- When you edit a static policy, you can change only certain elements of a static policy:
 - The action referenced by the policy.
 - A condition clause, such as when and unless.

You can't change these elements of a static policy:

- Changing a policy from a static policy to a template-linked policy.
- Changing the effect of a static policy from permit or forbid.
- The principal referenced by a static policy.
- The resource referenced by a static policy.
- To update a template-linked policy, you must update the template instead.

Note

Verified Permissions is *eventually consistent*. It can take a few seconds for a new or changed element to propagate through the service and be visible in the results of other Verified Permissions operations.

Request Syntax

```
{  
  "definition": { ... },
```

```
"policyId": "string",  
"policyStoreId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

definition

Specifies the updated policy content that you want to replace on the specified policy. The content must be valid Cedar policy language text.

You can change only the following elements from the policy definition:

- The `action` referenced by the policy.
- Any conditional clauses, such as `when` or `unless` clauses.

You **can't** change the following elements:

- Changing from `static` to `templateLinked`.
- Changing the effect of the policy from `permit` or `forbid`.
- The `principal` referenced by the policy.
- The `resource` referenced by the policy.

Type: [UpdatePolicyDefinition](#) object

Note: This object is a Union. Only one member of this object can be specified or returned.

Required: Yes

policyId

Specifies the ID of the policy that you want to update. To find this value, you can use [ListPolicies](#).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

policyStoreId

Specifies the ID of the policy store that contains the policy that you want to update.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

Response Syntax

```
{
  "createdDate": "string",
  "lastUpdatedDate": "string",
  "policyId": "string",
  "policyStoreId": "string",
  "policyType": "string",
  "principal": {
    "entityId": "string",
    "entityType": "string"
  },
  "resource": {
    "entityId": "string",
    "entityType": "string"
  }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

createdDate

The date and time that the policy was originally created.

Type: Timestamp

lastUpdatedDate

The date and time that the policy was most recently updated.

Type: Timestamp

policyId

The ID of the policy that was updated.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

policyStoreId

The ID of the policy store that contains the policy that was updated.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

policyType

The type of the policy that was updated.

Type: String

Valid Values: STATIC | TEMPLATE_LINKED

principal

The principal specified in the policy's scope. This element isn't included in the response when `Principal` isn't present in the policy content.

Type: [EntityIdentifier](#) object

resource

The resource specified in the policy's scope. This element isn't included in the response when `Resource` isn't present in the policy content.

Type: [EntityIdentifier](#) object

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

ConflictException

The request failed because another request to modify a resource occurred at the same.

HTTP Status Code: 400

InternalServerErrorException

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ServiceQuotaExceededException

The request failed because it would cause a service quota to be exceeded.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example

The following example replaces the definition of the specified static policy with a new one.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.UpdatePolicy
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "policyId": "SPEXAMPLEabcdefg111111",
  "definition": {
    "static": {
      "statement": "permit(principal, action, resource in Album:\""public_folder
\");"
    }
  }
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "arn": "arn:aws:verifiedpermissions:us-east-1:123456789012:policy/
PSEXAMPLEEabcdefg111111/SPEXAMPLEEabcdefg111111",
  "createdDate": "20230613T22:56:48.020321Z",
  "lastUpdatedDate": "20230613T23:26:09.764859Z",
  "policyId": "SPEXAMPLEEabcdefg111111",
  "policyStoreId": "PSEXAMPLEEabcdefg111111",
  "policyType": "STATIC",
  "resource": {
    "entityType": "Album",
    "entityId": "public_folder"
  }
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)

- [AWS SDK for Ruby V3](#)

UpdatePolicyStore

Modifies the validation setting for a policy store.

Note

Verified Permissions is *eventually consistent*. It can take a few seconds for a new or changed element to propagate through the service and be visible in the results of other Verified Permissions operations.

Request Syntax

```
{
  "description": "string",
  "policyStoreId": "string",
  "validationSettings": {
    "mode": "string"
  }
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

policyStoreId

Specifies the ID of the policy store that you want to update

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `[a-zA-Z0-9-]*`

Required: Yes

validationSettings

A structure that defines the validation settings that want to enable for the policy store.

Type: [ValidationSettings](#) object

Required: Yes

description

Descriptive text that you can provide to help with identification of the current policy store.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 150.

Required: No

Response Syntax

```
{
  "arn": "string",
  "createdDate": "string",
  "lastUpdatedDate": "string",
  "policyStoreId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

arn

The [Amazon Resource Name \(ARN\)](#) of the updated policy store.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2500.

Pattern: `arn:[^:]*:[^:]*:[^:]*:[^:]*:.*`

createdDate

The date and time that the policy store was originally created.

Type: Timestamp

lastUpdatedDate

The date and time that the policy store was most recently updated.

Type: Timestamp

policyStoreId

The ID of the updated policy store.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `[a-zA-Z0-9-]*`

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

ConflictException

The request failed because another request to modify a resource occurred at the same.

HTTP Status Code: 400

InternalServerErrorException

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if . . . then . . . else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more

information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example

The following example turns off the validation settings for a policy store.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.UpdatePolicyStore
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>
```

```
{
  "policyStoreId": "PSEXAMPLEabcdefgh111111",
  "validationSettings": { "mode": "OFF" }
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "policyStoreId": "PSEXAMPLEabcdefgh111111",
  "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefgh111111",
  "createdDate": "2023-05-17T18:36:10.134448Z",
  "lastUpdatedDate": "2023-05-23T18:18:12.443083Z"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)

- [AWS SDK for Ruby V3](#)

UpdatePolicyTemplate

Updates the specified policy template. You can update only the description and the some elements of the [policyBody](#).

Important

Changes you make to the policy template content are immediately (within the constraints of eventual consistency) reflected in authorization decisions that involve all template-linked policies instantiated from this template.

Note

Verified Permissions is [eventually consistent](#). It can take a few seconds for a new or changed element to propagate through the service and be visible in the results of other Verified Permissions operations.

Request Syntax

```
{
  "description": "string",
  "policyStoreId": "string",
  "policyTemplateId": "string",
  "statement": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

Note

In the following list, the required parameters are described first.

policyStoreId

Specifies the ID of the policy store that contains the policy template that you want to update.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

policyTemplateId

Specifies the ID of the policy template that you want to update.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

statement

Specifies new statement content written in Cedar policy language to replace the current body of the policy template.

You can change only the following elements of the policy body:

- The `action` referenced by the policy template.
- Any conditional clauses, such as `when` or `unless` clauses.

You **can't** change the following elements:

- The `effect` (`permit` or `forbid`) of the policy template.
- The `principal` referenced by the policy template.
- The `resource` referenced by the policy template.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 10000.

Required: Yes

description

Specifies a new description to apply to the policy template.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 150.

Required: No

Response Syntax

```
{
  "createdDate": "string",
  "lastUpdatedDate": "string",
  "policyStoreId": "string",
  "policyTemplateId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

createdDate

The date and time that the policy template was originally created.

Type: Timestamp

lastUpdatedDate

The date and time that the policy template was most recently updated.

Type: Timestamp

policyStoreId

The ID of the policy store that contains the updated policy template.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

policyTemplateId

The ID of the updated policy template.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

You don't have sufficient access to perform this action.

HTTP Status Code: 400

ConflictException

The request failed because another request to modify a resource occurred at the same.

HTTP Status Code: 400

InternalServerError

The request failed because of an internal error. Try your request again later

HTTP Status Code: 500

ResourceNotFoundException

The request failed because it references a resource that doesn't exist.

HTTP Status Code: 400

ThrottlingException

The request failed because it exceeded a throttling quota.

HTTP Status Code: 400

ValidationException

The request failed because one or more input parameters don't satisfy their constraint requirements. The output is provided as a list of fields and a reason for each field that isn't valid.

The possible reasons include the following:

- **UnrecognizedEntityType**

The policy includes an entity type that isn't found in the schema.

- **UnrecognizedActionId**

The policy includes an action id that isn't found in the schema.

- **InvalidActionApplication**

The policy includes an action that, according to the schema, doesn't support the specified principal and resource.

- **UnexpectedType**

The policy included an operand that isn't a valid type for the specified operation.

- **IncompatibleTypes**

The types of elements included in a set, or the types of expressions used in an `if...then...else` clause aren't compatible in this context.

- **MissingAttribute**

The policy attempts to access a record or entity attribute that isn't specified in the schema. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **UnsafeOptionalAttributeAccess**

The policy attempts to access a record or entity attribute that is optional and isn't guaranteed to be present. Test for the existence of the attribute first before attempting to access its value. For more information, see the [has \(presence of attribute test\) operator](#) in the *Cedar Policy Language Guide*.

- **ImpossiblePolicy**

Cedar has determined that a policy condition always evaluates to false. If the policy is always false, it can never apply to any query, and so it can never affect an authorization decision.

- **WrongNumberArguments**

The policy references an extension type with the wrong number of arguments.

- **FunctionArgumentValidationError**

Cedar couldn't parse the argument passed to an extension type. For example, a string that is to be parsed as an IPv4 address can contain only digits and the period character.

HTTP Status Code: 400

Examples

Example

The following example updates a policy template with both a new description and a new policy body. The effect, principal, and resource are the same as the original policy template. Only the action in the head, and the when and unless clauses can be different.

Note

The JSON in the parameters of this operation are strings that can contain embedded quotation marks (") within the outermost quotation mark pair. This requires that you *stringify* the JSON object by preceding all embedded quotation marks with a backslash character (\ ") and combining all lines into a single text line with no line breaks. Example strings are displayed wrapped across multiple lines here for readability, but the operation requires the parameters be submitted as single line strings.

Sample Request

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.UpdatePolicyTemplate
User-Agent: <UserAgentString>
```

```
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "description": "My updated template description",
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "policyTemplateId": "PTEXAMPLEabcdefg111111",
  "statement": "\"ResearchAccess\"\n\npermit(\n  principal in ?principal,\n  action
== Action::\"view\", \n  resource in ?resource)\n\nwhen {\n  principal has
department && principal.department == \"research\"\n};",
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Jun 2023 20:00:59 GMT
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
vary: origin
vary: access-control-request-method
vary: access-control-request-headers
x-amzn-requestid: a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111
Connection: keep-alive

{
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "policyTemplateId": "PTEXAMPLEabcdefg111111",
  "createdDate": "2023-05-17T18:58:48.795411Z",
  "lastUpdatedDate": "2023-05-17T19:18:48.870209Z"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)

- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

Data Types

The Amazon Verified Permissions API contains several data types that various actions use. This section describes each data type in detail.

Note

The order of each element in a data type structure is not guaranteed. Applications should not assume a particular order.

The following data types are supported:

- [ActionIdentifier](#)
- [AttributeValue](#)
- [BatchIsAuthorizedInputItem](#)
- [BatchIsAuthorizedOutputItem](#)
- [BatchIsAuthorizedWithTokenInputItem](#)
- [BatchIsAuthorizedWithTokenOutputItem](#)
- [CognitoGroupConfiguration](#)
- [CognitoGroupConfigurationDetail](#)
- [CognitoGroupConfigurationItem](#)
- [CognitoUserPoolConfiguration](#)
- [CognitoUserPoolConfigurationDetail](#)
- [CognitoUserPoolConfigurationItem](#)
- [Configuration](#)
- [ConfigurationDetail](#)
- [ConfigurationItem](#)
- [ContextDefinition](#)
- [DeterminingPolicyItem](#)
- [EntitiesDefinition](#)
- [EntityIdentifier](#)
- [EntityItem](#)

- [EntityReference](#)
- [EvaluationErrorItem](#)
- [IdentitySourceDetails](#)
- [IdentitySourceFilter](#)
- [IdentitySourceItem](#)
- [IdentitySourceItemDetails](#)
- [PolicyDefinition](#)
- [PolicyDefinitionDetail](#)
- [PolicyDefinitionItem](#)
- [PolicyFilter](#)
- [PolicyItem](#)
- [PolicyStoreItem](#)
- [PolicyTemplateItem](#)
- [ResourceConflict](#)
- [SchemaDefinition](#)
- [StaticPolicyDefinition](#)
- [StaticPolicyDefinitionDetail](#)
- [StaticPolicyDefinitionItem](#)
- [TemplateLinkedPolicyDefinition](#)
- [TemplateLinkedPolicyDefinitionDetail](#)
- [TemplateLinkedPolicyDefinitionItem](#)
- [UpdateCognitoGroupConfiguration](#)
- [UpdateCognitoUserPoolConfiguration](#)
- [UpdateConfiguration](#)
- [UpdatePolicyDefinition](#)
- [UpdateStaticPolicyDefinition](#)
- [ValidationExceptionField](#)
- [ValidationSettings](#)

ActionIdentifier

Contains information about an action for a request for which an authorization decision is made.

This data type is used as a request parameter to the [IsAuthorized](#), [BatchIsAuthorized](#), and [IsAuthorizedWithToken](#) operations.

```
Example: { "actionId": "<action name>", "actionType": "Action" }
```

Contents

Note

In the following list, the required parameters are described first.

actionId

The ID of an action.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: .*

Required: Yes

actionType

The type of an action.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: Action\$|^\.+::Action

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

AttributeValue

The value of an attribute.

Contains information about the runtime context for a request for which an authorization decision is made.

This data type is used as a member of the [ContextDefinition](#) structure which is used as a request parameter for the [IsAuthorized](#), [BatchIsAuthorized](#), and [IsAuthorizedWithToken](#) operations.

Contents

Note

In the following list, the required parameters are described first.

Important

This data type is a UNION, so only one of the following members can be specified when used or returned.

boolean

An attribute value of [Boolean](#) type.

Example: {"boolean": true}

Type: Boolean

Required: No

entityIdentifier

An attribute value of type [EntityIdentifier](#).

Example: "entityIdentifier": { "entityId": "<id>", "entityType": "<entity type>"}

Type: [EntityIdentifier](#) object

Required: No

long

An attribute value of [Long](#) type.

Example: {"long": 0}

Type: Long

Required: No

record

An attribute value of [Record](#) type.

Example: {"record": { "keyName": {} } }

Type: String to [AttributeValue](#) object map

Required: No

set

An attribute value of [Set](#) type.

Example: {"set": [{}] }

Type: Array of [AttributeValue](#) objects

Required: No

string

An attribute value of [String](#) type.

Example: {"string": "abc"}

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

BatchIsAuthorizedInputItem

An authorization request that you include in a BatchIsAuthorized API request.

Contents

Note

In the following list, the required parameters are described first.

action

Specifies the requested action to be authorized. For example, PhotoFlash::ReadPhoto.

Type: [ActionIdentifier](#) object

Required: No

context

Specifies additional context that can be used to make more granular authorization decisions.

Type: [ContextDefinition](#) object

Note: This object is a Union. Only one member of this object can be specified or returned.

Required: No

principal

Specifies the principal for which the authorization decision is to be made.

Type: [EntityIdentifier](#) object

Required: No

resource

Specifies the resource that you want an authorization decision for. For example, PhotoFlash::Photo.

Type: [EntityIdentifier](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

BatchIsAuthorizedOutputItem

The decision, based on policy evaluation, from an individual authorization request in a BatchIsAuthorized API request.

Contents

Note

In the following list, the required parameters are described first.

decision

An authorization decision that indicates if the authorization request should be allowed or denied.

Type: String

Valid Values: ALLOW | DENY

Required: Yes

determiningPolicies

The list of determining policies used to make the authorization decision. For example, if there are two matching policies, where one is a forbid and the other is a permit, then the forbid policy will be the determining policy. In the case of multiple matching permit policies then there would be multiple determining policies. In the case that no policies match, and hence the response is DENY, there would be no determining policies.

Type: Array of [DeterminingPolicyItem](#) objects

Required: Yes

errors

Errors that occurred while making an authorization decision. For example, a policy might reference an entity or attribute that doesn't exist in the request.

Type: Array of [EvaluationErrorItem](#) objects

Required: Yes

request

The authorization request that initiated the decision.

Type: [BatchIsAuthorizedInputItem](#) object

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

BatchIsAuthorizedWithTokenInputItem

An authorization request that you include in a BatchIsAuthorizedWithToken API request.

Contents

Note

In the following list, the required parameters are described first.

action

Specifies the requested action to be authorized. For example, PhotoFlash::ReadPhoto.

Type: [ActionIdentifier](#) object

Required: No

context

Specifies additional context that can be used to make more granular authorization decisions.

Type: [ContextDefinition](#) object

Note: This object is a Union. Only one member of this object can be specified or returned.

Required: No

resource

Specifies the resource that you want an authorization decision for. For example, PhotoFlash::Photo.

Type: [EntityIdentifier](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

BatchIsAuthorizedWithTokenOutputItem

The decision, based on policy evaluation, from an individual authorization request in a `BatchIsAuthorizedWithToken` API request.

Contents

Note

In the following list, the required parameters are described first.

decision

An authorization decision that indicates if the authorization request should be allowed or denied.

Type: String

Valid Values: ALLOW | DENY

Required: Yes

determiningPolicies

The list of determining policies used to make the authorization decision. For example, if there are two matching policies, where one is a forbid and the other is a permit, then the forbid policy will be the determining policy. In the case of multiple matching permit policies then there would be multiple determining policies. In the case that no policies match, and hence the response is DENY, there would be no determining policies.

Type: Array of [DeterminingPolicyItem](#) objects

Required: Yes

errors

Errors that occurred while making an authorization decision. For example, a policy might reference an entity or attribute that doesn't exist in the request.

Type: Array of [EvaluationErrorItem](#) objects

Required: Yes

request

The authorization request that initiated the decision.

Type: [BatchIsAuthorizedWithTokenInputItem](#) object

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

CognitoGroupConfiguration

The type of entity that a policy store maps to groups from an Amazon Cognito user pool identity source.

This data type is part of a [CognitoUserPoolConfiguration](#) structure and is a request parameter in [CreateIdentitySource](#).

Contents

Note

In the following list, the required parameters are described first.

groupEntityType

The name of the schema entity type that's mapped to the user pool group. Defaults to `AWS::CognitoGroup`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: (`[_a-zA-Z][_a-zA-Z0-9]*`: :)*`[_a-zA-Z][_a-zA-Z0-9]*`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

CognitoGroupConfigurationDetail

The type of entity that a policy store maps to groups from an Amazon Cognito user pool identity source.

This data type is part of an [CognitoUserPoolConfigurationDetail](#) structure and is a response parameter to [GetIdentitySource](#).

Contents

Note

In the following list, the required parameters are described first.

groupEntityType

The name of the schema entity type that's mapped to the user pool group. Defaults to `AWS::CognitoGroup`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: (`[_a-zA-Z][_a-zA-Z0-9]*`: :)*`[_a-zA-Z][_a-zA-Z0-9]*`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

CognitoGroupConfigurationItem

The type of entity that a policy store maps to groups from an Amazon Cognito user pool identity source.

This data type is part of an [CognitoUserPoolConfigurationItem](#) structure and is a response parameter to [ListIdentitySources](#).

Contents

Note

In the following list, the required parameters are described first.

groupEntityType

The name of the schema entity type that's mapped to the user pool group. Defaults to `AWS::CognitoGroup`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: (`[_a-zA-Z][_a-zA-Z0-9]*`: :)*`[_a-zA-Z][_a-zA-Z0-9]*`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

CognitoUserPoolConfiguration

The configuration for an identity source that represents a connection to an Amazon Cognito user pool used as an identity provider for Verified Permissions.

This data type is used as a field that is part of an [Configuration](#) structure that is used as a parameter to [CreateIdentitySource](#).

```
Example:"CognitoUserPoolConfiguration":{"UserPoolArn":"arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-east-1_1a2b3c4d5","ClientIds":["a1b2c3d4e5f6g7h8i9j0kalbmc"],"groupConfiguration":{"groupEntityType":"MyCorp::Group"}}
```

Contents

Note

In the following list, the required parameters are described first.

userPoolArn

The [Amazon Resource Name \(ARN\)](#) of the Amazon Cognito user pool that contains the identities to be authorized.

```
Example: "UserPoolArn": "arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-east-1_1a2b3c4d5"
```

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `arn:[a-zA-Z0-9-]+:cognito-idp:(([a-zA-Z0-9-]+:\d{12}:userpool/[\w-]+_[0-9a-zA-Z]+))`

Required: Yes

clientIds

The unique application client IDs that are associated with the specified Amazon Cognito user pool.

Example: "ClientIds": ["&ExampleCogClientId;"]

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 1000 items.

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: .*

Required: No

groupConfiguration

The type of entity that a policy store maps to groups from an Amazon Cognito user pool identity source.

Type: [CognitoGroupConfiguration](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

CognitoUserPoolConfigurationDetail

The configuration for an identity source that represents a connection to an Amazon Cognito user pool used as an identity provider for Verified Permissions.

This data type is used as a field that is part of an [ConfigurationDetail](#) structure that is part of the response to [GetIdentitySource](#).

```
Example:"CognitoUserPoolConfiguration":{"UserPoolArn":"arn:aws:cognito-  
idp:us-east-1:123456789012:userpool/us-east-1_1a2b3c4d5","ClientIds":  
["a1b2c3d4e5f6g7h8i9j0kalbmc"],"groupConfiguration":{"groupEntityType":  
"MyCorp::Group"}}
```

Contents

Note

In the following list, the required parameters are described first.

clientIds

The unique application client IDs that are associated with the specified Amazon Cognito user pool.

Example: "clientIds": ["&ExampleCogClientId;"]

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 1000 items.

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: .*

Required: Yes

issuer

The OpenID Connect (OIDC) issuer ID of the Amazon Cognito user pool that contains the identities to be authorized.

Example: "issuer": "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_1a2b3c4d5"

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: https://.*

Required: Yes

userPoolArn

The [Amazon Resource Name \(ARN\)](#) of the Amazon Cognito user pool that contains the identities to be authorized.

Example: "userPoolArn": "arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-east-1_1a2b3c4d5"

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: arn:[a-zA-Z0-9-]+:cognito-idp:(([a-zA-Z0-9-]+\d{12}:userpool/[\w-]+_[0-9a-zA-Z]+))

Required: Yes

groupConfiguration

The type of entity that a policy store maps to groups from an Amazon Cognito user pool identity source.

Type: [CognitoGroupConfigurationDetail](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)

- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

CognitoUserPoolConfigurationItem

The configuration for an identity source that represents a connection to an Amazon Cognito user pool used as an identity provider for Verified Permissions.

This data type is used as a field that is part of the [ConfigurationItem](#) structure that is part of the response to [ListIdentitySources](#).

```
Example:"CognitoUserPoolConfiguration":{"UserPoolArn":"arn:aws:cognito-  
idp:us-east-1:123456789012:userpool/us-east-1_1a2b3c4d5","ClientIds":  
["a1b2c3d4e5f6g7h8i9j0kalbmc"],"groupConfiguration":{"groupEntityType":  
"MyCorp::Group"}}
```

Contents

Note

In the following list, the required parameters are described first.

clientIds

The unique application client IDs that are associated with the specified Amazon Cognito user pool.

Example: "clientIds": ["&ExampleCogClientId;"]

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 1000 items.

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: .*

Required: Yes

issuer

The OpenID Connect (OIDC) issuer ID of the Amazon Cognito user pool that contains the identities to be authorized.

Example: "issuer": "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_1a2b3c4d5"

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: https://.*

Required: Yes

userPoolArn

The [Amazon Resource Name \(ARN\)](#) of the Amazon Cognito user pool that contains the identities to be authorized.

Example: "userPoolArn": "arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-east-1_1a2b3c4d5"

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: arn:[a-zA-Z0-9-]+:cognito-idp:(([a-zA-Z0-9-]+\d{12}:userpool/[\w-]+_[0-9a-zA-Z]+))

Required: Yes

groupConfiguration

The type of entity that a policy store maps to groups from an Amazon Cognito user pool identity source.

Type: [CognitoGroupConfigurationItem](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)

- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Configuration

Contains configuration information used when creating a new identity source.

Note

At this time, the only valid member of this structure is a Amazon Cognito user pool configuration.
Specifies a `userPoolArn`, a `groupConfiguration`, and a `ClientId`.

This data type is used as a request parameter for the [CreateIdentitySource](#) operation.

Contents

Note

In the following list, the required parameters are described first.

Important

This data type is a UNION, so only one of the following members can be specified when used or returned.

cognitoUserPoolConfiguration

Contains configuration details of a Amazon Cognito user pool that Verified Permissions can use as a source of authenticated identities as entities. It specifies the [Amazon Resource Name \(ARN\)](#) of a Amazon Cognito user pool and one or more application client IDs.

```
Example: "configuration":{"cognitoUserPoolConfiguration":
{"userPoolArn":"arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-
east-1_1a2b3c4d5","clientId":
["a1b2c3d4e5f6g7h8i9j0kalbmc"],"groupConfiguration":{"groupEntityType":
"MyCorp::Group"}}}}
```

Type: [CognitoUserPoolConfiguration](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ConfigurationDetail

Contains configuration information about an identity source.

This data type is a response parameter to the [GetIdentitySource](#) operation.

Contents

Note

In the following list, the required parameters are described first.

Important

This data type is a UNION, so only one of the following members can be specified when used or returned.

cognitoUserPoolConfiguration

Contains configuration details of a Amazon Cognito user pool that Verified Permissions can use as a source of authenticated identities as entities. It specifies the [Amazon Resource Name \(ARN\)](#) of a Amazon Cognito user pool, the policy store entity that you want to assign to user groups, and one or more application client IDs.

```
Example: "configuration":{"cognitoUserPoolConfiguration":
{"userPoolArn":"arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-
east-1_1a2b3c4d5","clientIds":
["a1b2c3d4e5f6g7h8i9j0kalbmc"],"groupConfiguration":{"groupEntityType":
"MyCorp::Group"}}}
```

Type: [CognitoUserPoolConfigurationDetail](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ConfigurationItem

Contains configuration information about an identity source.

This data type is a response parameter to the [ListIdentitySources](#) operation.

Contents

Note

In the following list, the required parameters are described first.

Important

This data type is a UNION, so only one of the following members can be specified when used or returned.

cognitoUserPoolConfiguration

Contains configuration details of a Amazon Cognito user pool that Verified Permissions can use as a source of authenticated identities as entities. It specifies the [Amazon Resource Name \(ARN\)](#) of a Amazon Cognito user pool, the policy store entity that you want to assign to user groups, and one or more application client IDs.

```
Example: "configuration":{"cognitoUserPoolConfiguration":
{"userPoolArn":"arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-
east-1_1a2b3c4d5","clientIds":
["a1b2c3d4e5f6g7h8i9j0kalbmc"],"groupConfiguration":{"groupEntityType":
"MyCorp::Group"}}}
```

Type: [CognitoUserPoolConfigurationItem](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ContextDefinition

Contains additional details about the context of the request. Verified Permissions evaluates this information in an authorization request as part of the `when` and `unless` clauses in a policy.

This data type is used as a request parameter for the [IsAuthorized](#), [BatchIsAuthorized](#), and [IsAuthorizedWithToken](#) operations.

```
Example: "context": {"contextMap": {"<KeyName1>": {"boolean": true}, "<KeyName2>": {"long": 1234}}}
```

Contents

Note

In the following list, the required parameters are described first.

Important

This data type is a UNION, so only one of the following members can be specified when used or returned.

contextMap

An list of attributes that are needed to successfully evaluate an authorization request. Each attribute in this array must include a map of a data type and its value.

```
Example: "contextMap": {"<KeyName1>": {"boolean": true}, "<KeyName2>": {"long": 1234}}
```

Type: String to [AttributeValue](#) object map

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DeterminingPolicyItem

Contains information about one of the policies that determined an authorization decision.

This data type is used as an element in a response parameter for the [IsAuthorized](#), [BatchIsAuthorized](#), and [IsAuthorizedWithToken](#) operations.

Example: "determiningPolicies": [{"policyId": "SPEXAMPLEabcdefg111111"}]

Contents

Note

In the following list, the required parameters are described first.

policyId

The Id of a policy that determined to an authorization decision.

Example: "policyId": "SPEXAMPLEabcdefg111111"

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

EntitiesDefinition

Contains the list of entities to be considered during an authorization request. This includes all principals, resources, and actions required to successfully evaluate the request.

This data type is used as a field in the response parameter for the [IsAuthorized](#) and [IsAuthorizedWithToken](#) operations.

Contents

Note

In the following list, the required parameters are described first.

Important

This data type is a UNION, so only one of the following members can be specified when used or returned.

entityList

An array of entities that are needed to successfully evaluate an authorization request. Each entity in this array must include an identifier for the entity, the attributes of the entity, and a list of any parent entities.

Type: Array of [EntityItem](#) objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)

- [AWS SDK for Ruby V3](#)

EntityIdentifier

Contains the identifier of an entity, including its ID and type.

This data type is used as a request parameter for [IsAuthorized](#) operation, and as a response parameter for the [CreatePolicy](#), [GetPolicy](#), and [UpdatePolicy](#) operations.

Example: `{"entityId":"string","entityType":"string"}`

Contents

Note

In the following list, the required parameters are described first.

entityId

The identifier of an entity.

`"entityId":"identifier"`

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `.*`

Required: Yes

entityType

The type of an entity.

Example: `"entityType":"typeName"`

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `.*`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

EntityItem

Contains information about an entity that can be referenced in a Cedar policy.

This data type is used as one of the fields in the [EntitiesDefinition](#) structure.

```
{ "identifier": { "entityType": "Photo", "entityId":  
"VacationPhoto94.jpg" }, "attributes": {}, "parents": [ { "entityType":  
"Album", "entityId": "alice_folder" } ] }
```

Contents

Note

In the following list, the required parameters are described first.

identifier

The identifier of the entity.

Type: [EntityIdentifier](#) object

Required: Yes

attributes

A list of attributes for the entity.

Type: String to [AttributeValue](#) object map

Required: No

parents

The parents in the hierarchy that contains the entity.

Type: Array of [EntityIdentifier](#) objects

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

EntityReference

Contains information about a principal or resource that can be referenced in a Cedar policy.

This data type is used as part of the [PolicyFilter](#) structure that is used as a request parameter for the [ListPolicies](#) operation..

Contents

Note

In the following list, the required parameters are described first.

Important

This data type is a UNION, so only one of the following members can be specified when used or returned.

identifier

The identifier of the entity. It can consist of either an EntityType and EntityId, a principal, or a resource.

Type: [EntityIdentifier](#) object

Required: No

unspecified

Used to indicate that a principal or resource is not specified. This can be used to search for policies that are not associated with a specific principal or resource.

Type: Boolean

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

EvaluationErrorItem

Contains a description of an evaluation error.

This data type is a response parameter of the [IsAuthorized](#), [BatchIsAuthorized](#), and [IsAuthorizedWithToken](#) operations.

Contents

Note

In the following list, the required parameters are described first.

errorDescription

The error description.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

IdentitySourceDetails

This data type has been deprecated.

A structure that contains configuration of the identity source.

This data type was a response parameter for the [GetIdentitySource](#) operation. Replaced by [ConfigurationDetail](#).

Contents

Note

In the following list, the required parameters are described first.

clientIds

This member has been deprecated.

The application client IDs associated with the specified Amazon Cognito user pool that are enabled for this identity source.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 1000 items.

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: . *

Required: No

discoveryUrl

This member has been deprecated.

The well-known URL that points to this user pool's OIDC discovery endpoint. This is a URL string in the following format. This URL replaces the placeholders for both the AWS Region and the user pool identifier with those appropriate for this user pool.

```
https://cognito-idp.<region>.amazonaws.com/<user-pool-id>/well-known/openid-configuration
```

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `https://.*`

Required: No

openIdIssuer

This member has been deprecated.

A string that identifies the type of OIDC service represented by this identity source.

At this time, the only valid value is `cognito`.

Type: String

Valid Values: `COGNITO`

Required: No

userPoolArn

This member has been deprecated.

The [Amazon Resource Name \(ARN\)](#) of the Amazon Cognito user pool whose identities are accessible to this Verified Permissions policy store.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `arn:[a-zA-Z0-9-]+:cognito-idp:(([a-zA-Z0-9-]+\d{12}:userpool/[\w-]+\[_0-9a-zA-Z\]+))`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)

- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

IdentitySourceFilter

A structure that defines characteristics of an identity source that you can use to filter.

This data type is a request parameter for the [ListIdentityStores](#) operation.

Contents

Note

In the following list, the required parameters are described first.

principalEntityType

The Cedar entity type of the principals returned by the identity provider (IdP) associated with this identity source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: .*

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

IdentitySourceItem

A structure that defines an identity source.

This data type is a response parameter to the [ListIdentitySources](#) operation.

Contents

Note

In the following list, the required parameters are described first.

createdDate

The date and time the identity source was originally created.

Type: Timestamp

Required: Yes

identitySourceId

The unique identifier of the identity source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

lastUpdatedDate

The date and time the identity source was most recently updated.

Type: Timestamp

Required: Yes

policyStoreId

The identifier of the policy store that contains the identity source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

principalEntityType

The Cedar entity type of the principals returned from the IdP associated with this identity source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: .*

Required: Yes

configuration

Contains configuration information about an identity source.

Type: [ConfigurationItem](#) object

Note: This object is a Union. Only one member of this object can be specified or returned.

Required: No

details

This member has been deprecated.

A structure that contains the details of the associated identity provider (IdP).

Type: [IdentitySourceItemDetails](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

IdentitySourceItemDetails

This data type has been deprecated.

A structure that contains configuration of the identity source.

This data type was a response parameter for the [ListIdentitySources](#) operation. Replaced by [ConfigurationItem](#).

Contents

Note

In the following list, the required parameters are described first.

clientIds

This member has been deprecated.

The application client IDs associated with the specified Amazon Cognito user pool that are enabled for this identity source.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 1000 items.

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: . *

Required: No

discoveryUrl

This member has been deprecated.

The well-known URL that points to this user pool's OIDC discovery endpoint. This is a URL string in the following format. This URL replaces the placeholders for both the AWS Region and the user pool identifier with those appropriate for this user pool.

```
https://cognito-idp.<region>.amazonaws.com/<user-pool-id>/well-known/openid-configuration
```

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `https://.*`

Required: No

openIdIssuer

This member has been deprecated.

A string that identifies the type of OIDC service represented by this identity source.

At this time, the only valid value is `cognito`.

Type: String

Valid Values: `COGNITO`

Required: No

userPoolArn

This member has been deprecated.

The Amazon Cognito user pool whose identities are accessible to this Verified Permissions policy store.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `arn:[a-zA-Z0-9-]+:cognito-idp:([a-zA-Z0-9-]+:\d{12}:userpool/[\w-]+_[0-9a-zA-Z]+)`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)

- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

PolicyDefinition

A structure that contains the details for a Cedar policy definition. It includes the policy type, a description, and a policy body. This is a top level data type used to create a policy.

This data type is used as a request parameter for the [CreatePolicy](#) operation. This structure must always have either a `static` or a `templateLinked` element.

Contents

Note

In the following list, the required parameters are described first.

Important

This data type is a UNION, so only one of the following members can be specified when used or returned.

static

A structure that describes a static policy. An static policy doesn't use a template or allow placeholders for entities.

Type: [StaticPolicyDefinition](#) object

Required: No

templateLinked

A structure that describes a policy that was instantiated from a template. The template can specify placeholders for `principal` and `resource`. When you use [CreatePolicy](#) to create a policy from a template, you specify the exact principal and resource to use for the instantiated policy.

Type: [TemplateLinkedPolicyDefinition](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

PolicyDefinitionDetail

A structure that describes a policy definition. It must always have either an `static` or a `templateLinked` element.

This data type is used as a response parameter for the [GetPolicy](#) operation.

Contents

Note

In the following list, the required parameters are described first.

Important

This data type is a UNION, so only one of the following members can be specified when used or returned.

static

Information about a static policy that wasn't created with a policy template.

Type: [StaticPolicyDefinitionDetail](#) object

Required: No

templateLinked

Information about a template-linked policy that was created by instantiating a policy template.

Type: [TemplateLinkedPolicyDefinitionDetail](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

PolicyDefinitionItem

A structure that describes a [PolicyDefinitionItem](#). It will always have either a `StaticPolicy` or a `TemplateLinkedPolicy` element.

This data type is used as a response parameter for the [CreatePolicy](#) and [ListPolicies](#) operations.

Contents

Note

In the following list, the required parameters are described first.

Important

This data type is a UNION, so only one of the following members can be specified when used or returned.

static

Information about a static policy that wasn't created with a policy template.

Type: [StaticPolicyDefinitionItem](#) object

Required: No

templateLinked

Information about a template-linked policy that was created by instantiating a policy template.

Type: [TemplateLinkedPolicyDefinitionItem](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

PolicyFilter

Contains information about a filter to refine policies returned in a query.

This data type is used as a response parameter for the [ListPolicies](#) operation.

Contents

Note

In the following list, the required parameters are described first.

policyTemplateId

Filters the output to only template-linked policies that were instantiated from the specified policy template.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: No

policyType

Filters the output to only policies of the specified type.

Type: String

Valid Values: STATIC | TEMPLATE_LINKED

Required: No

principal

Filters the output to only policies that reference the specified principal.

Type: [EntityReference](#) object

Note: This object is a Union. Only one member of this object can be specified or returned.

Required: No

resource

Filters the output to only policies that reference the specified resource.

Type: [EntityReference](#) object

Note: This object is a Union. Only one member of this object can be specified or returned.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

PolicyItem

Contains information about a policy.

This data type is used as a response parameter for the [ListPolicies](#) operation.

Contents

Note

In the following list, the required parameters are described first.

createdDate

The date and time the policy was created.

Type: Timestamp

Required: Yes

definition

The policy definition of an item in the list of policies returned.

Type: [PolicyDefinitionItem](#) object

Note: This object is a Union. Only one member of this object can be specified or returned.

Required: Yes

lastUpdatedDate

The date and time the policy was most recently updated.

Type: Timestamp

Required: Yes

policyId

The identifier of the policy you want information about.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

policyStoreId

The identifier of the PolicyStore where the policy you want information about is stored.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

policyType

The type of the policy. This is one of the following values:

- `static`
- `templateLinked`

Type: String

Valid Values: `STATIC` | `TEMPLATE_LINKED`

Required: Yes

principal

The principal associated with the policy.

Type: [EntityIdentifier](#) object

Required: No

resource

The resource associated with the policy.

Type: [EntityIdentifier](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

PolicyStoreItem

Contains information about a policy store.

This data type is used as a response parameter for the [ListPolicyStores](#) operation.

Contents

Note

In the following list, the required parameters are described first.

arn

The Amazon Resource Name (ARN) of the policy store.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2500.

Pattern: `arn:[^:]*:[^:]*:[^:]*:[^:]*:.*`

Required: Yes

createdDate

The date and time the policy was created.

Type: Timestamp

Required: Yes

policyStoreId

The unique identifier of the policy store.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `[a-zA-Z0-9-]*`

Required: Yes

description

Descriptive text that you can provide to help with identification of the current policy store.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 150.

Required: No

lastUpdatedDate

The date and time the policy store was most recently updated.

Type: Timestamp

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

PolicyTemplateItem

Contains details about a policy template

This data type is used as a response parameter for the [ListPolicyTemplates](#) operation.

Contents

Note

In the following list, the required parameters are described first.

createdDate

The date and time that the policy template was created.

Type: Timestamp

Required: Yes

lastUpdatedDate

The date and time that the policy template was most recently updated.

Type: Timestamp

Required: Yes

policyStoreId

The unique identifier of the policy store that contains the template.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

policyTemplateId

The unique identifier of the policy template.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

description

The description attached to the policy template.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 150.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ResourceConflict

Contains information about a resource conflict.

Contents

Note

In the following list, the required parameters are described first.

resourceId

The unique identifier of the resource involved in a conflict.

Type: String

Required: Yes

resourceType

The type of the resource involved in a conflict.

Type: String

Valid Values: IDENTITY_SOURCE | POLICY_STORE | POLICY | POLICY_TEMPLATE | SCHEMA

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

SchemaDefinition

Contains a list of principal types, resource types, and actions that can be specified in policies stored in the same policy store. If the validation mode for the policy store is set to `STRICT`, then policies that can't be validated by this schema are rejected by Verified Permissions and can't be stored in the policy store.

Contents

Note

In the following list, the required parameters are described first.

Important

This data type is a UNION, so only one of the following members can be specified when used or returned.

cedarJson

A JSON string representation of the schema supported by applications that use this policy store. For more information, see [Policy store schema](#) in the *Amazon Verified Permissions User Guide*.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100000.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)

- [AWS SDK for Ruby V3](#)

StaticPolicyDefinition

Contains information about a static policy.

This data type is used as a field that is part of the [PolicyDefinitionDetail](#) type.

Contents

Note

In the following list, the required parameters are described first.

statement

The policy content of the static policy, written in the Cedar policy language.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 10000.

Required: Yes

description

The description of the static policy.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 150.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

StaticPolicyDefinitionDetail

A structure that contains details about a static policy. It includes the description and policy body.

This data type is used within a [PolicyDefinition](#) structure as part of a request parameter for the [CreatePolicy](#) operation.

Contents

Note

In the following list, the required parameters are described first.

statement

The content of the static policy written in the Cedar policy language.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 10000.

Required: Yes

description

A description of the static policy.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 150.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)

- [AWS SDK for Ruby V3](#)

StaticPolicyDefinitionItem

A structure that contains details about a static policy. It includes the description and policy statement.

This data type is used within a [PolicyDefinition](#) structure as part of a request parameter for the [CreatePolicy](#) operation.

Contents

Note

In the following list, the required parameters are described first.

description

A description of the static policy.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 150.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

TemplateLinkedPolicyDefinition

Contains information about a policy created by instantiating a policy template.

Contents

Note

In the following list, the required parameters are described first.

policyTemplateId

The unique identifier of the policy template used to create this policy.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

principal

The principal associated with this template-linked policy. Verified Permissions substitutes this principal for the `?principal` placeholder in the policy template when it evaluates an authorization request.

Type: [EntityIdentifier](#) object

Required: No

resource

The resource associated with this template-linked policy. Verified Permissions substitutes this resource for the `?resource` placeholder in the policy template when it evaluates an authorization request.

Type: [EntityIdentifier](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

TemplateLinkedPolicyDefinitionDetail

Contains information about a policy that was created by instantiating a policy template.

Contents

Note

In the following list, the required parameters are described first.

policyTemplateId

The unique identifier of the policy template used to create this policy.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

principal

The principal associated with this template-linked policy. Verified Permissions substitutes this principal for the `?principal` placeholder in the policy template when it evaluates an authorization request.

Type: [EntityIdentifier](#) object

Required: No

resource

The resource associated with this template-linked policy. Verified Permissions substitutes this resource for the `?resource` placeholder in the policy template when it evaluates an authorization request.

Type: [EntityIdentifier](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

TemplateLinkedPolicyDefinitionItem

Contains information about a policy created by instantiating a policy template.

This

Contents

Note

In the following list, the required parameters are described first.

policyTemplateId

The unique identifier of the policy template used to create this policy.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9-]*

Required: Yes

principal

The principal associated with this template-linked policy. Verified Permissions substitutes this principal for the `?principal` placeholder in the policy template when it evaluates an authorization request.

Type: [EntityIdentifier](#) object

Required: No

resource

The resource associated with this template-linked policy. Verified Permissions substitutes this resource for the `?resource` placeholder in the policy template when it evaluates an authorization request.

Type: [EntityIdentifier](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

UpdateCognitoGroupConfiguration

The user group entities from an Amazon Cognito user pool identity source.

Contents

Note

In the following list, the required parameters are described first.

groupEntityType

The name of the schema entity type that's mapped to the user pool group. Defaults to `AWS::CognitoGroup`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `([_a-zA-Z][_a-zA-Z0-9]*:)*[_a-zA-Z][_a-zA-Z0-9]*`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

UpdateCognitoUserPoolConfiguration

Contains configuration details of a Amazon Cognito user pool for use with an identity source.

Contents

Note

In the following list, the required parameters are described first.

userPoolArn

The [Amazon Resource Name \(ARN\)](#) of the Amazon Cognito user pool associated with this identity source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `arn:[a-zA-Z0-9-]+:cognito-idp:([a-zA-Z0-9-]+:\d{12}:userpool/[\w-]+_[0-9a-zA-Z]+)`

Required: Yes

clientIds

The client ID of an app client that is configured for the specified Amazon Cognito user pool.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 1000 items.

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `.*`

Required: No

groupConfiguration

The configuration of the user groups from an Amazon Cognito user pool identity source.

Type: [UpdateCognitoGroupConfiguration](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

UpdateConfiguration

Contains an updated configuration to replace the configuration in an existing identity source.

Note

At this time, the only valid member of this structure is a Amazon Cognito user pool configuration.

You must specify a `userPoolArn`, and optionally, a `ClientId`.

Contents

Note

In the following list, the required parameters are described first.

Important

This data type is a UNION, so only one of the following members can be specified when used or returned.

cognitoUserPoolConfiguration

Contains configuration details of a Amazon Cognito user pool.

Type: [UpdateCognitoUserPoolConfiguration](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)

- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

UpdatePolicyDefinition

Contains information about updates to be applied to a policy.

This data type is used as a request parameter in the [UpdatePolicy](#) operation.

Contents

Note

In the following list, the required parameters are described first.

Important

This data type is a UNION, so only one of the following members can be specified when used or returned.

static

Contains details about the updates to be applied to a static policy.

Type: [UpdateStaticPolicyDefinition](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

UpdateStaticPolicyDefinition

Contains information about an update to a static policy.

Contents

Note

In the following list, the required parameters are described first.

statement

Specifies the Cedar policy language text to be added to or replaced on the static policy.

Important

You can change only the following elements from the original content:

- The action referenced by the policy.
- Any conditional clauses, such as when or unless clauses.

You **can't** change the following elements:

- Changing from `StaticPolicy` to `TemplateLinkedPolicy`.
- The effect (permit or forbid) of the policy.
- The principal referenced by the policy.
- The resource referenced by the policy.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 10000.

Required: Yes

description

Specifies the description to be added to or replaced on the static policy.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 150.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ValidationExceptionField

Details about a field that failed policy validation.

Contents

Note

In the following list, the required parameters are described first.

message

Describes the policy validation error.

Type: String

Required: Yes

path

The path to the specific element that Verified Permissions found to be not valid.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ValidationSettings

A structure that contains Cedar policy validation settings for the policy store. The validation mode determines which validation failures that Cedar considers serious enough to block acceptance of a new or edited static policy or policy template.

This data type is used as a request parameter in the [CreatePolicyStore](#) and [UpdatePolicyStore](#) operations.

Contents

Note

In the following list, the required parameters are described first.

mode

The validation mode currently configured for this policy store. The valid values are:

- **OFF** – Neither Verified Permissions nor Cedar perform any validation on policies. No validation errors are reported by either service.
- **STRICT** – Requires a schema to be present in the policy store. Cedar performs validation on all submitted new or updated static policies and policy templates. Any that fail validation are rejected and Cedar doesn't store them in the policy store.

Important

If Mode=STRICT and the policy store doesn't contain a schema, Verified Permissions rejects all static policies and policy templates because there is no schema to validate against.

To submit a static policy or policy template without a schema, you must turn off validation.

Type: String

Valid Values: OFF | STRICT

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Making API requests

Query requests for the Amazon Verified Permissions are HTTP or HTTPS requests that use an HTTP verb such as GET or POST.

Verified Permissions endpoints

An *endpoint* is a URL that serves as an entry point for a web service. You can select an appropriate AWS Region endpoint when you make your requests to reduce latency. For information about the endpoints used by Verified Permissions, see [Amazon Verified Permissions](#) in the *Amazon Web Services General Reference*.

Query parameters

Each query request must include some common parameters to handle authentication and selection of an action. For more information, see [Common Parameters](#).

Some API operations take lists of parameters. These lists are specified using the following notation:

```
param.member.n
```

Values of *n* are integers starting from 1. All lists of parameters must follow this notation, including lists that contain only one parameter. A query parameter list looks like the following example.

```
&attribute.member.1=this  
&attribute.member.2=that
```

Request identifiers

In every response from an AWS Query API, there is a `ResponseMetadata` element, which contains a `RequestId` element. This string is a unique identifier that AWS assigns to provide tracking information. Although `RequestId` is included as part of every response, it isn't listed on the individual API documentation pages to improve readability and to reduce redundancy.

Query API authentication

You send query requests over HTTPS. You must include a signature in every query request. For more information about creating and including a signature, see [Signing AWS API Requests](#) in the *Amazon Web Services General Reference*.

Available libraries

AWS provides libraries, sample code, tutorials, and other resources for software developers who prefer to build applications using language-specific APIs instead of the command-line tools and Query API. These libraries provide basic functions (not included in the APIs), such as request authentication, request retries, and error handling so that it's easier to get started. Verified Permissions libraries and resources are available for the following languages and platforms:

- [AWS SDK for Go](#)
- [AWS SDK for Java 2.x](#)
- [AWS SDK for Java 1.x](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for .NET](#)
- [AWS SDK for PHP](#)
- [AWS SDK for Python \(Boto\)](#)
- [AWS SDK for Ruby](#)

For more information about libraries and sample code in all languages, see [Sample Code & Libraries](#).

Making API requests using the POST method

If you don't use one of the AWS SDKs, you can make Verified Permissions requests over HTTPS using the POST request method. The POST method requires that you specify the operation in the header of the request and provide the data for the operation in JSON format in the body of the request.

Header name	Header value
Host	The Amazon Verified Permissions endpoint. For example: <code>verifiedpermissions.us-east-1.amazonaws.com</code>
X-Amz-Date	<p>You must provide the timestamp in either the HTTP Date header or the AWS <code>x-amz-date</code> header. Some HTTP client libraries don't let you set the Date header. When an <code>x-amz-date</code> header is present, the system ignores any Date header during the request authentication.</p> <p>The <code>x-amz-date</code> header must be specified in ISO 8601 basic format. For example: <code>20130315T092054Z</code></p>
Authorization	The set of authorization parameters that AWS uses to ensure the validity and authenticity of the request. For more information about constructing this header, see Signature Version 4 Signing Process in the <i>Amazon Web Services General Reference</i> .
X-Amz-Target	<p>Specifies the Verified Permissions operation that you want to perform.</p> <p><code>VerifiedPermissions.</code> <i>API_Name</i></p> <p>For example, to call the <code>CreatePolicy</code> operation, use the following target value.</p> <p><code>VerifiedPermissions.CreatePolicy</code></p>
Content-Type	<p>Specifies the input format. Use the following value.</p> <p><code>application/x-amz-json-1.0</code></p>
Accept	<p>Specifies the response format. Use the following value.</p> <p><code>application/x-amz-json-1.0</code></p>
Content-Length	Size of the payload in bytes.
Content-Encoding	Specifies the encoding format of the input and output. Use the following value.

Header name	Header value
	amz-1.0

The following is an example header for an HTTP request to return a list of all policies in the specified policy store in the AWS account where the `Principal` references a User named `alice`. In this example, the `Authorization` line is word-wrapped here for easier reading. Don't word wrap it in your actual request.

```
POST HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: identity
X-Amz-Target: VerifiedPermissions.ListPolicies
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
  "Filter": {
    "Principal": {
      "Id": {
        "EntityType": "User",
        "EntityId": "alice"
      }
    }
  }
}
```

Common Parameters

The following list contains the parameters that all actions use for signing Signature Version 4 requests with a query string. Any action-specific parameters are listed in the topic for that action. For more information about Signature Version 4, see [Signing AWS API requests](#) in the *IAM User Guide*.

Action

The action to be performed.

Type: string

Required: Yes

Version

The API version that the request is written for, expressed in the format YYYY-MM-DD.

Type: string

Required: Yes

X-Amz-Algorithm

The hash algorithm that you used to create the request signature.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Valid Values: AWS4-HMAC-SHA256

Required: Conditional

X-Amz-Credential

The credential scope value, which is a string that includes your access key, the date, the region you are targeting, the service you are requesting, and a termination string ("aws4_request"). The value is expressed in the following format: *access_key/YYYYMMDD/region/service/aws4_request*.

For more information, see [Create a signed AWS API request](#) in the *IAM User Guide*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-Date

The date that is used to create the signature. The format must be ISO 8601 basic format (YYYYMMDD'T'HHMMSS'Z'). For example, the following date time is a valid X-Amz-Date value: 20120325T120000Z.

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see [Elements of an AWS API request signature](#) in the *IAM User Guide*.

Type: string

Required: Conditional

X-Amz-Security-Token

The temporary security token that was obtained through a call to AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS STS, see [AWS services that work with IAM](#) in the *IAM User Guide*.

Condition: If you're using temporary security credentials from AWS STS, you must include the security token.

Type: string

Required: Conditional

X-Amz-Signature

Specifies the hex-encoded signature that was calculated from the string to sign and the derived signing key.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-SignedHeaders

Specifies all the HTTP headers that were included as part of the canonical request. For more information about specifying signed headers, see [Create a signed AWS API request](#) in the *IAM User Guide*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 400

InternalFailure

The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

InvalidAction

The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 400

InvalidClientTokenId

The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

NotAuthorized

You do not have permission to perform this action.

HTTP Status Code: 400

OptInRequired

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

RequestExpired

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

ServiceUnavailable

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 400

ValidationError

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400

Document history for the Amazon Verified Permissions API Reference Guide

The following table describes the documentation releases for Verified Permissions.

Change	Description	Date
Initial public release	Initial public release of the Amazon Verified Permissions API Reference Guide	June 13, 2023

AWS Glossary

For the latest AWS terminology, see the [AWS glossary](#) in the *AWS Glossary Reference*.