



API Reference, Version 2

AWS IoT Greengrass



AWS IoT Greengrass: API Reference, Version 2

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	2
AssociateServiceRoleToAccount	4
Request Syntax	4
URI Request Parameters	4
Request Body	4
Response Syntax	4
Response Elements	5
Errors	5
See Also	5
BatchAssociateClientDeviceWithCoreDevice	7
Request Syntax	7
URI Request Parameters	7
Request Body	8
Response Syntax	8
Response Elements	8
Errors	9
See Also	9
BatchDisassociateClientDeviceFromCoreDevice	11
Request Syntax	11
URI Request Parameters	11
Request Body	11
Response Syntax	12
Response Elements	12
Errors	12
See Also	13
CancelDeployment	15
Request Syntax	15
URI Request Parameters	15
Request Body	15
Response Syntax	15
Response Elements	15
Errors	16
See Also	17

CreateComponentVersion	18
Request Syntax	18
URI Request Parameters	20
Request Body	20
Response Syntax	21
Response Elements	22
Errors	23
See Also	24
CreateDeployment	25
Request Syntax	25
URI Request Parameters	27
Request Body	27
Response Syntax	29
Response Elements	29
Errors	30
See Also	31
DeleteComponent	32
Request Syntax	32
URI Request Parameters	32
Request Body	32
Response Syntax	32
Response Elements	33
Errors	33
See Also	34
DeleteCoreDevice	35
Request Syntax	35
URI Request Parameters	35
Request Body	35
Response Syntax	35
Response Elements	35
Errors	35
See Also	36
DeleteDeployment	38
Request Syntax	38
URI Request Parameters	38
Request Body	38

Response Syntax	38
Response Elements	38
Errors	38
See Also	39
DescribeComponent	41
Request Syntax	41
URI Request Parameters	41
Request Body	41
Response Syntax	41
Response Elements	42
Errors	44
See Also	44
DisassociateServiceRoleFromAccount	46
Request Syntax	46
URI Request Parameters	46
Request Body	46
Response Syntax	46
Response Elements	46
Errors	47
See Also	47
GetComponent	48
Request Syntax	48
URI Request Parameters	48
Request Body	48
Response Syntax	48
Response Elements	49
Errors	49
See Also	50
GetComponentVersionArtifact	52
Request Syntax	52
URI Request Parameters	52
Request Body	53
Response Syntax	53
Response Elements	53
Errors	53
See Also	54

GetConnectivityInfo	56
Request Syntax	56
URI Request Parameters	56
Request Body	56
Response Syntax	56
Response Elements	57
Errors	57
See Also	58
GetCoreDevice	59
Request Syntax	59
URI Request Parameters	59
Request Body	60
Response Syntax	60
Response Elements	60
Errors	62
See Also	62
GetDeployment	64
Request Syntax	64
URI Request Parameters	64
Request Body	64
Response Syntax	64
Response Elements	66
Errors	68
See Also	69
GetServiceRoleForAccount	70
Request Syntax	70
URI Request Parameters	70
Request Body	70
Response Syntax	70
Response Elements	70
Errors	71
See Also	71
ListClientDevicesAssociatedWithCoreDevice	72
Request Syntax	72
URI Request Parameters	72
Request Body	72

Response Syntax	72
Response Elements	73
Errors	73
See Also	74
ListComponents	75
Request Syntax	75
URI Request Parameters	75
Request Body	75
Response Syntax	75
Response Elements	76
Errors	77
See Also	77
ListComponentVersions	79
Request Syntax	79
URI Request Parameters	79
Request Body	79
Response Syntax	79
Response Elements	80
Errors	80
See Also	81
ListCoreDevices	82
Request Syntax	82
URI Request Parameters	82
Request Body	83
Response Syntax	83
Response Elements	84
Errors	84
See Also	85
ListDeployments	86
Request Syntax	86
URI Request Parameters	86
Request Body	87
Response Syntax	87
Response Elements	87
Errors	88
See Also	88

ListEffectiveDeployments	90
Request Syntax	90
URI Request Parameters	90
Request Body	90
Response Syntax	90
Response Elements	91
Errors	91
See Also	92
ListInstalledComponents	94
Request Syntax	94
URI Request Parameters	94
Request Body	95
Response Syntax	95
Response Elements	96
Errors	96
See Also	97
ListTagsForResource	99
Request Syntax	99
URI Request Parameters	99
Request Body	99
Response Syntax	99
Response Elements	99
Errors	100
See Also	100
ResolveComponentCandidates	102
Request Syntax	102
URI Request Parameters	103
Request Body	103
Response Syntax	103
Response Elements	104
Errors	104
See Also	105
TagResource	106
Request Syntax	106
URI Request Parameters	106
Request Body	106

Response Syntax	107
Response Elements	107
Errors	107
See Also	107
UntagResource	109
Request Syntax	109
URI Request Parameters	109
Request Body	109
Response Syntax	109
Response Elements	110
Errors	110
See Also	110
UpdateConnectivityInfo	112
Request Syntax	112
URI Request Parameters	112
Request Body	112
Response Syntax	113
Response Elements	113
Errors	113
See Also	114
Data Types	115
AssociateClientDeviceWithCoreDeviceEntry	117
Contents	117
See Also	117
AssociateClientDeviceWithCoreDeviceErrorEntry	118
Contents	118
See Also	118
AssociatedClientDevice	120
Contents	120
See Also	120
CloudComponentStatus	121
Contents	121
See Also	122
Component	123
Contents	123
See Also	123

ComponentCandidate	125
Contents	125
See Also	126
ComponentConfigurationUpdate	127
Contents	127
See Also	127
ComponentDependencyRequirement	129
Contents	129
See Also	129
ComponentDeploymentSpecification	131
Contents	131
See Also	132
ComponentLatestVersion	133
Contents	133
See Also	134
ComponentPlatform	135
Contents	135
See Also	135
ComponentRunWith	137
Contents	137
See Also	138
ComponentVersionListItem	139
Contents	139
See Also	139
ConnectivityInfo	141
Contents	141
See Also	142
CoreDevice	143
Contents	143
See Also	143
Deployment	145
Contents	145
See Also	146
DeploymentComponentUpdatePolicy	148
Contents	148
See Also	149

DeploymentConfigurationValidationPolicy	150
Contents	150
See Also	150
DeploymentIoTJobConfiguration	151
Contents	151
See Also	151
DeploymentPolicies	153
Contents	153
See Also	153
DisassociateClientDeviceFromCoreDeviceEntry	155
Contents	155
See Also	155
DisassociateClientDeviceFromCoreDeviceErrorEntry	156
Contents	156
See Also	156
EffectiveDeployment	158
Contents	158
See Also	160
EffectiveDeploymentStatusDetails	161
Contents	161
See Also	162
InstalledComponent	163
Contents	163
See Also	165
IoTJobAbortConfig	166
Contents	166
See Also	166
IoTJobAbortCriteria	167
Contents	167
See Also	168
IoTJobExecutionsRolloutConfig	169
Contents	169
See Also	169
IoTJobExponentialRolloutRate	170
Contents	170
See Also	170

IoTJobRateIncreaseCriteria	172
Contents	172
See Also	172
IoTJobTimeoutConfig	173
Contents	173
See Also	173
LambdaContainerParams	174
Contents	174
See Also	175
LambdaDeviceMount	176
Contents	176
See Also	176
LambdaEventSource	178
Contents	178
See Also	178
LambdaExecutionParameters	179
Contents	179
See Also	181
LambdaFunctionRecipeSource	182
Contents	182
See Also	183
LambdaLinuxProcessParams	184
Contents	184
See Also	184
LambdaVolumeMount	185
Contents	185
See Also	186
ResolvedComponentVersion	187
Contents	187
See Also	188
SystemResourceLimits	189
Contents	189
See Also	189
ValidationExceptionField	191
Contents	191
See Also	191

Common Parameters	192
Common Errors	195

Welcome

AWS IoT Greengrass brings local compute, messaging, data management, sync, and ML inference capabilities to edge devices. This enables devices to collect and analyze data closer to the source of information, react autonomously to local events, and communicate securely with each other on local networks. Local devices can also communicate securely with AWS IoT Core and export IoT data to the AWS Cloud. AWS IoT Greengrass developers can use AWS Lambda functions and components to create and deploy applications to fleets of edge devices for local operation.

AWS IoT Greengrass Version 2 provides a new major version of the AWS IoT Greengrass Core software, new APIs, and a new console. Use this API reference to learn how to use the AWS IoT Greengrass V2 API operations to manage components, manage deployments, and core devices.

For more information, see [What is AWS IoT Greengrass?](#) in the *AWS IoT Greengrass V2 Developer Guide*.

This document was last published on July 3, 2024.

Actions

The following actions are supported:

- [AssociateServiceRoleToAccount](#)
- [BatchAssociateClientDeviceWithCoreDevice](#)
- [BatchDisassociateClientDeviceFromCoreDevice](#)
- [CancelDeployment](#)
- [CreateComponentVersion](#)
- [CreateDeployment](#)
- [DeleteComponent](#)
- [DeleteCoreDevice](#)
- [DeleteDeployment](#)
- [DescribeComponent](#)
- [DisassociateServiceRoleFromAccount](#)
- [GetComponent](#)
- [GetComponentVersionArtifact](#)
- [GetConnectivityInfo](#)
- [GetCoreDevice](#)
- [GetDeployment](#)
- [GetServiceRoleForAccount](#)
- [ListClientDevicesAssociatedWithCoreDevice](#)
- [ListComponents](#)
- [ListComponentVersions](#)
- [ListCoreDevices](#)
- [ListDeployments](#)
- [ListEffectiveDeployments](#)
- [ListInstalledComponents](#)
- [ListTagsForResource](#)
- [ResolveComponentCandidates](#)
- [TagResource](#)

- [UntagResource](#)
- [UpdateConnectivityInfo](#)

AssociateServiceRoleToAccount

Associates a Greengrass service role with AWS IoT Greengrass for your AWS account in this AWS Region. AWS IoT Greengrass uses this role to verify the identity of client devices and manage core device connectivity information. The role must include the [AWSGreengrassResourceAccessRolePolicy](#) managed policy or a custom policy that defines equivalent permissions for the AWS IoT Greengrass features that you use. For more information, see [Greengrass service role](#) in the *AWS IoT Greengrass Version 2 Developer Guide*.

Request Syntax

```
PUT /greengrass/servicerole HTTP/1.1
Content-type: application/json

{
  "RoleArn": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

RoleArn

The Amazon Resource Name (ARN) of the service role to associate with AWS IoT Greengrass for your AWS account in this AWS Region.

Type: String

Required: Yes

Response Syntax

```
HTTP/1.1 200
Content-type: application/json
```

```
{  
  "AssociatedAt": "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.

AssociatedAt

The time when the service role was associated with AWS IoT Greengrass for your AWS account in this AWS Region.

Type: String

Errors

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

InternalServerError

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

BatchAssociateClientDeviceWithCoreDevice

Associates a list of client devices with a core device. Use this API operation to specify which client devices can discover a core device through cloud discovery. With cloud discovery, client devices connect to AWS IoT Greengrass to retrieve associated core devices' connectivity information and certificates. For more information, see [Configure cloud discovery](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Note

Client devices are local IoT devices that connect to and communicate with an AWS IoT Greengrass core device over MQTT. You can connect client devices to a core device to sync MQTT messages and data to AWS IoT Core and interact with client devices in Greengrass components. For more information, see [Interact with local IoT devices](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Request Syntax

```
POST /greengrass/v2/coreDevices/coreDeviceThingName/associateClientDevices HTTP/1.1
Content-type: application/json
```

```
{
  "entries": [
    {
      "thingName": "string"
    }
  ]
}
```

URI Request Parameters

The request uses the following URI parameters.

[coreDeviceThingName](#)

The name of the core device. This is also the name of the AWS IoT thing.

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

Request Body

The request accepts the following data in JSON format.

entries

The list of client devices to associate.

Type: Array of [AssociateClientDeviceWithCoreDeviceEntry](#) objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "errorEntries": [
    {
      "code": "string",
      "message": "string",
      "thingName": "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.

errorEntries

The list of any errors for the entries in the request. Each error entry contains the name of the AWS IoT thing that failed to associate.

Type: Array of [AssociateClientDeviceWithCoreDeviceErrorEntry](#) objects

Array Members: Maximum number of 100 items.

Errors

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

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

BatchDisassociateClientDeviceFromCoreDevice

Disassociates a list of client devices from a core device. After you disassociate a client device from a core device, the client device won't be able to use cloud discovery to retrieve the core device's connectivity information and certificates.

Request Syntax

```
POST /greengrass/v2/coreDevices/coreDeviceThingName/disassociateClientDevices HTTP/1.1
Content-type: application/json
```

```
{
  "entries": [
    {
      "thingName": "string"
    }
  ]
}
```

URI Request Parameters

The request uses the following URI parameters.

coreDeviceThingName

The name of the core device. This is also the name of the AWS IoT thing.

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

Request Body

The request accepts the following data in JSON format.

entries

The list of client devices to disassociate.

Type: Array of [DisassociateClientDeviceFromCoreDeviceEntry](#) objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "errorEntries": [
    {
      "code": "string",
      "message": "string",
      "thingName": "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.

errorEntries

The list of any errors for the entries in the request. Each error entry contains the name of the AWS IoT thing that failed to disassociate.

Type: Array of [DisassociateClientDeviceFromCoreDeviceErrorEntry](#) objects

Array Members: Maximum number of 100 items.

Errors

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

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

CancelDeployment

Cancel a deployment. This operation cancels the deployment for devices that haven't yet received it. If a device already received the deployment, this operation doesn't change anything for that device.

Request Syntax

```
POST /greengrass/v2/deployments/deploymentId/cancel HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

deploymentId

The ID of the deployment.

Length Constraints: Minimum length of 1.

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "message": "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.

message

A message that communicates if the cancel was successful.

Type: String

Length Constraints: Minimum length of 1.

Errors

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

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

ConflictException

Your request has conflicting operations. This can occur if you're trying to perform more than one operation on the same resource at the same time.

HTTP Status Code: 409

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

CreateComponentVersion

Creates a component. Components are software that run on Greengrass core devices. After you develop and test a component on your core device, you can use this operation to upload your component to AWS IoT Greengrass. Then, you can deploy the component to other core devices.

You can use this operation to do the following:

- **Create components from recipes**

Create a component from a recipe, which is a file that defines the component's metadata, parameters, dependencies, lifecycle, artifacts, and platform capability. For more information, see [AWS IoT Greengrass component recipe reference](#) in the *AWS IoT Greengrass V2 Developer Guide*.


To create a component from a recipe, specify `inlineRecipe` when you call this operation.

- **Create components from Lambda functions**

Create a component from an AWS Lambda function that runs on AWS IoT Greengrass. This creates a recipe and artifacts from the Lambda function's deployment package. You can use this operation to migrate Lambda functions from AWS IoT Greengrass V1 to AWS IoT Greengrass V2.

This function accepts Lambda functions in all supported versions of Python, Node.js, and Java runtimes. AWS IoT Greengrass doesn't apply any additional restrictions on deprecated Lambda runtime versions.

To create a component from a Lambda function, specify `lambdaFunction` when you call this operation.

 **Note**

AWS IoT Greengrass currently supports Lambda functions on only Linux core devices.

Request Syntax

```
POST /greengrass/v2/createComponentVersion HTTP/1.1
Content-type: application/json

{
  "clientToken": "string",
```

```

"inlineRecipe": blob,
"lambdaFunction": {
  "componentDependencies": {
    "string" : {
      "dependencyType": "string",
      "versionRequirement": "string"
    }
  },
  "componentLambdaParameters": {
    "environmentVariables": {
      "string" : "string"
    },
    "eventSources": [
      {
        "topic": "string",
        "type": "string"
      }
    ],
    "execArgs": [ "string" ],
    "inputPayloadEncodingType": "string",
    "linuxProcessParams": {
      "containerParams": {
        "devices": [
          {
            "addGroupOwner": boolean,
            "path": "string",
            "permission": "string"
          }
        ],
        "memorySizeInKB": number,
        "mountROSysfs": boolean,
        "volumes": [
          {
            "addGroupOwner": boolean,
            "destinationPath": "string",
            "permission": "string",
            "sourcePath": "string"
          }
        ]
      }
    ],
    "isolationMode": "string"
  },
  "maxIdleTimeInSeconds": number,
  "maxInstancesCount": number,

```



```
    "maxQueueSize": number,
    "pinned": boolean,
    "statusTimeoutInSeconds": number,
    "timeoutInSeconds": number
  },
  "componentName": "string",
  "componentPlatforms": [
    {
      "attributes": {
        "string" : "string"
      },
      "name": "string"
    }
  ],
  "componentVersion": "string",
  "lambdaArn": "string"
},
"tags": {
  "string" : "string"
}
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

clientToken

A unique, case-sensitive identifier that you can provide to ensure that the request is idempotent. Idempotency means that the request is successfully processed only once, even if you send the request multiple times. When a request succeeds, and you specify the same client token for subsequent successful requests, the AWS IoT Greengrass V2 service returns the successful response that it caches from the previous request. AWS IoT Greengrass V2 caches successful responses for idempotent requests for up to 8 hours.

Type: String

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

Pattern: `^[a-zA-Z0-9-]+$`

Required: No

inlineRecipe

The recipe to use to create the component. The recipe defines the component's metadata, parameters, dependencies, lifecycle, artifacts, and platform compatibility.

You must specify either `inlineRecipe` or `lambdaFunction`.

Type: Base64-encoded binary data object

Required: No

lambdaFunction

The parameters to create a component from a Lambda function.

You must specify either `inlineRecipe` or `lambdaFunction`.

Type: [LambdaFunctionRecipeSource](#) object

Required: No

tags

A list of key-value pairs that contain metadata for the resource. For more information, see [Tag your resources](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String to string map

Map Entries: Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Value Length Constraints: Maximum length of 256.

Required: No

Response Syntax

```
HTTP/1.1 201
```

```
Content-type: application/json

{
  "arn": "string",
  "componentName": "string",
  "componentVersion": "string",
  "creationTimestamp": number,
  "status": {
    "componentState": "string",
    "errors": {
      "string" : "string"
    },
    "message": "string",
    "vendorGuidance": "string",
    "vendorGuidanceMessage": "string"
  }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 201 response.

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

arn

The [ARN](#) of the component version.

Type: String

Pattern: `arn:[^:]*:greengrass:[^:]*:(aws|[0-9]+):components:[^:]+:versions:[^:]+`

componentName

The name of the component.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

componentVersion

The version of the component.

Type: String

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

creationTimestamp

The time at which the component was created, expressed in ISO 8601 format.

Type: Timestamp

status

The status of the component version in AWS IoT Greengrass V2. This status is different from the status of the component on a core device.

Type: [CloudComponentStatus](#) object

Errors

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

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

ConflictException

Your request has conflicting operations. This can occur if you're trying to perform more than one operation on the same resource at the same time.

HTTP Status Code: 409

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

RequestAlreadyInProgressException

The request is already in progress. This exception occurs when you use a client token for multiple requests while AWS IoT Greengrass is still processing an earlier request that uses the same client token.

HTTP Status Code: 400

ServiceQuotaExceededException

Your request exceeds a service quota. For example, you might have the maximum number of components that you can create.

HTTP Status Code: 402

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

CreateDeployment

Creates a continuous deployment for a target, which is a Greengrass core device or group of core devices. When you add a new core device to a group of core devices that has a deployment, AWS IoT Greengrass deploys that group's deployment to the new device.

You can define one deployment for each target. When you create a new deployment for a target that has an existing deployment, you replace the previous deployment. AWS IoT Greengrass applies the new deployment to the target devices.

Every deployment has a revision number that indicates how many deployment revisions you define for a target. Use this operation to create a new revision of an existing deployment.

For more information, see the [Create deployments](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Request Syntax

```
POST /greengrass/v2/deployments HTTP/1.1
```

```
Content-type: application/json
```

```
{
  "clientToken": "string",
  "components": {
    "string" : {
      "componentVersion": "string",
      "configurationUpdate": {
        "merge": "string",
        "reset": [ "string" ]
      },
      "runWith": {
        "posixUser": "string",
        "systemResourceLimits": {
          "cpus": number,
          "memory": number
        },
        "windowsUser": "string"
      }
    }
  },
  "deploymentName": "string",
  "deploymentPolicies": {
    "componentUpdatePolicy": {
```

```
    "action": "string",
    "timeoutInSeconds": number
  },
  "configurationValidationPolicy": {
    "timeoutInSeconds": number
  },
  "failureHandlingPolicy": "string"
},
"iotJobConfiguration": {
  "abortConfig": {
    "criteriaList": [
      {
        "action": "string",
        "failureType": "string",
        "minNumberOfExecutedThings": number,
        "thresholdPercentage": number
      }
    ]
  },
  "jobExecutionsRolloutConfig": {
    "exponentialRate": {
      "baseRatePerMinute": number,
      "incrementFactor": number,
      "rateIncreaseCriteria": {
        "numberOfNotifiedThings": number,
        "numberOfSucceededThings": number
      }
    },
    "maximumPerMinute": number
  },
  "timeoutConfig": {
    "inProgressTimeoutInMinutes": number
  }
},
"parentTargetArn": "string",
"tags": {
  "string" : "string"
},
"targetArn": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

clientToken

A unique, case-sensitive identifier that you can provide to ensure that the request is idempotent. Idempotency means that the request is successfully processed only once, even if you send the request multiple times. When a request succeeds, and you specify the same client token for subsequent successful requests, the AWS IoT Greengrass V2 service returns the successful response that it caches from the previous request. AWS IoT Greengrass V2 caches successful responses for idempotent requests for up to 8 hours.

Type: String

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

Pattern: `^[a-zA-Z0-9-]+$`

Required: No

components

The components to deploy. This is a dictionary, where each key is the name of a component, and each key's value is the version and configuration to deploy for that component.

Type: String to [ComponentDeploymentSpecification](#) object map

Key Length Constraints: Minimum length of 1.

Required: No

deploymentName

The name of the deployment.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Required: No

deploymentPolicies

The deployment policies for the deployment. These policies define how the deployment updates components and handles failure.

Type: [DeploymentPolicies](#) object

Required: No

iotJobConfiguration

The job configuration for the deployment configuration. The job configuration specifies the rollout, timeout, and stop configurations for the deployment configuration.

Type: [DeploymentIoTJobConfiguration](#) object

Required: No

parentTargetArn

The parent deployment's target [ARN](#) within a subdeployment.

Type: String

Pattern: `arn:[^:]*:iot:[^:]*:[0-9]+:thinggroup/.+`

Required: No

tags

A list of key-value pairs that contain metadata for the resource. For more information, see [Tag your resources](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String to string map

Map Entries: Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Value Length Constraints: Maximum length of 256.

Required: No

targetArn

The [ARN](#) of the target AWS IoT thing or thing group. When creating a subdeployment, the targetARN can only be a thing group.

Type: String

Pattern: `arn:[^:]*:iot:[^:]*:[0-9]+:(thing|thinggroup)/.+`

Required: Yes

Response Syntax

```
HTTP/1.1 201
Content-type: application/json

{
  "deploymentId": "string",
  "iotJobArn": "string",
  "iotJobId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 201 response.

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

deploymentId

The ID of the deployment.

Type: String

Length Constraints: Minimum length of 1.

iotJobArn

The [ARN](#) of the AWS IoT job that applies the deployment to target devices.

Type: String

Pattern: `arn:[^:]*:iot:[^:]*:[0-9]+:job/.+`

iotJobId

The ID of the AWS IoT job that applies the deployment to target devices.

Type: String

Length Constraints: Minimum length of 1.

Errors

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

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

ConflictException

Your request has conflicting operations. This can occur if you're trying to perform more than one operation on the same resource at the same time.

HTTP Status Code: 409

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

RequestAlreadyInProgressException

The request is already in progress. This exception occurs when you use a client token for multiple requests while AWS IoT Greengrass is still processing an earlier request that uses the same client token.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

DeleteComponent

Deletes a version of a component from AWS IoT Greengrass.

Note

This operation deletes the component's recipe and artifacts. As a result, deployments that refer to this component version will fail. If you have deployments that use this component version, you can remove the component from the deployment or update the deployment to use a valid version.

Request Syntax

```
DELETE /greengrass/v2/components/arn HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

arn

The [ARN](#) of the component version.

Pattern: `arn:[^:]*:greengrass:[^:]*:(aws|[0-9]+):components:[^:]*:versions:[^:]*`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 204
```

Response Elements

If the action is successful, the service sends back an HTTP 204 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 permission to perform the action.

HTTP Status Code: 403

ConflictException

Your request has conflicting operations. This can occur if you're trying to perform more than one operation on the same resource at the same time.

HTTP Status Code: 409

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

DeleteCoreDevice

Deletes a Greengrass core device, which is an AWS IoT thing. This operation removes the core device from the list of core devices. This operation doesn't delete the AWS IoT thing. For more information about how to delete the AWS IoT thing, see [DeleteThing](#) in the *AWS IoT API Reference*.

Request Syntax

```
DELETE /greengrass/v2/coreDevices/coreDeviceThingName HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

[coreDeviceThingName](#)

The name of the core device. This is also the name of the AWS IoT thing.

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 204
```

Response Elements

If the action is successful, the service sends back an HTTP 204 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 permission to perform the action.

HTTP Status Code: 403

ConflictException

Your request has conflicting operations. This can occur if you're trying to perform more than one operation on the same resource at the same time.

HTTP Status Code: 409

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

DeleteDeployment

Deletes a deployment. To delete an active deployment, you must first cancel it. For more information, see [CancelDeployment](#).

Deleting a deployment doesn't affect core devices that run that deployment, because core devices store the deployment's configuration on the device. Additionally, core devices can roll back to a previous deployment that has been deleted.

Request Syntax

```
DELETE /greengrass/v2/deployments/deploymentId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

[deploymentId](#)

The ID of the deployment.

Length Constraints: Minimum length of 1.

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 204
```

Response Elements

If the action is successful, the service sends back an HTTP 204 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 permission to perform the action.

HTTP Status Code: 403

ConflictException

Your request has conflicting operations. This can occur if you're trying to perform more than one operation on the same resource at the same time.

HTTP Status Code: 409

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

DescribeComponent

Retrieves metadata for a version of a component.

Request Syntax

```
GET /greengrass/v2/components/arn/metadata HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

arn

The [ARN](#) of the component version.

Pattern: `arn:[^:]*:greengrass:[^:]*:(aws|[0-9]+):components:[^:]+:versions:[^:]+`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "arn": "string",
  "componentName": "string",
  "componentVersion": "string",
  "creationTimestamp": number,
  "description": "string",
  "platforms": [
    {
      "attributes": {
        "string": "string"
      }
    }
  ]
}
```

```
    },
    "name": "string"
  }
],
"publisher": "string",
"status": {
  "componentState": "string",
  "errors": {
    "string" : "string"
  },
  "message": "string",
  "vendorGuidance": "string",
  "vendorGuidanceMessage": "string"
},
"tags": {
  "string" : "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 [ARN](#) of the component version.

Type: String

Pattern: `arn:[^:]*:greengrass:[^:]*:(aws|[0-9]+):components:[^:]+:versions:[^:]+`

componentName

The name of the component.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

componentVersion

The version of the component.

Type: String

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

creationTimestamp

The time at which the component was created, expressed in ISO 8601 format.

Type: Timestamp

description

The description of the component version.

Type: String

platforms

The platforms that the component version supports.

Type: Array of [ComponentPlatform](#) objects

publisher

The publisher of the component version.

Type: String

status

The status of the component version in AWS IoT Greengrass V2. This status is different from the status of the component on a core device.

Type: [CloudComponentStatus](#) object

tags

A list of key-value pairs that contain metadata for the resource. For more information, see [Tag your resources](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String to string map

Map Entries: Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Value Length Constraints: Maximum length of 256.

Errors

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

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

DisassociateServiceRoleFromAccount

Disassociates the Greengrass service role from AWS IoT Greengrass for your AWS account in this AWS Region. Without a service role, AWS IoT Greengrass can't verify the identity of client devices or manage core device connectivity information. For more information, see [Greengrass service role](#) in the *AWS IoT Greengrass Version 2 Developer Guide*.

Request Syntax

```
DELETE /greengrass/servicerole HTTP/1.1
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "DisassociatedAt": "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.

[DisassociatedAt](#)

The time when the service role was disassociated from AWS IoT Greengrass for your AWS account in this AWS Region.

Type: String

Errors

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

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

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](#)

GetComponent

Gets the recipe for a version of a component.

Request Syntax

```
GET /greengrass/v2/components/arn?recipeOutputFormat=recipeOutputFormat HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

arn

The [ARN](#) of the component version.

Pattern: `arn:[^:]*:greengrass:[^:]*:(aws|[0-9]+):components:[^:]+:versions:[^:]+`

Required: Yes

recipeOutputFormat

The format of the recipe.

Valid Values: JSON | YAML

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "recipe": blob,
  "recipeOutputFormat": "string",
  "tags": {
    "string" : "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.

recipe

The recipe of the component version.

Type: Base64-encoded binary data object

recipeOutputFormat

The format of the recipe.

Type: String

Valid Values: JSON | YAML

tags

A list of key-value pairs that contain metadata for the resource. For more information, see [Tag your resources](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String to string map

Map Entries: Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Value Length Constraints: Maximum length of 256.

Errors

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

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

GetComponentVersionArtifact

Gets the pre-signed URL to download a public or a Lambda component artifact. Core devices call this operation to identify the URL that they can use to download an artifact to install.

Request Syntax

```
GET /greengrass/v2/components/arn/artifacts/artifactName?s3EndpointType=s3EndpointType
HTTP/1.1
x-amz-iot-endpoint-type: iotEndpointType
```

URI Request Parameters

The request uses the following URI parameters.

arn

The [ARN](#) of the component version. Specify the ARN of a public or a Lambda component version.

Pattern: `arn:[^:]*:greengrass:[^:]*:(aws|[0-9]+):components:[^:]+:versions:[^:]+`

Required: Yes

artifactName

The name of the artifact.

You can use the [GetComponent](#) operation to download the component recipe, which includes the URI of the artifact. The artifact name is the section of the URI after the scheme. For example, in the artifact URI `greengrass:SomeArtifact.zip`, the artifact name is `SomeArtifact.zip`.

Length Constraints: Minimum length of 1.

Required: Yes

iotEndpointType

Determines if the Amazon S3 URL returned is a FIPS pre-signed URL endpoint. Specify `fips` if you want the returned Amazon S3 pre-signed URL to point to an Amazon S3 FIPS endpoint. If you don't specify a value, the default is `standard`.

Valid Values: fips | standard

s3EndpointType

Specifies the endpoint to use when getting Amazon S3 pre-signed URLs.

All AWS Regions except US East (N. Virginia) use REGIONAL in all cases. In the US East (N. Virginia) Region the default is GLOBAL, but you can change it to REGIONAL with this parameter.

Valid Values: REGIONAL | GLOBAL

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "preSignedUrl": "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.

preSignedUrl

The URL of the artifact.

Type: String

Length Constraints: Minimum length of 1.

Errors

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

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

GetConnectivityInfo

Retrieves connectivity information for a Greengrass core device.

Connectivity information includes endpoints and ports where client devices can connect to an MQTT broker on the core device. When a client device calls the [AWS IoT Greengrass discovery API](#), AWS IoT Greengrass returns connectivity information for all of the core devices where the client device can connect. For more information, see [Connect client devices to core devices](#) in the *AWS IoT Greengrass Version 2 Developer Guide*.

Request Syntax

```
GET /greengrass/things/thingName/connectivityInfo HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

thingName

The name of the core device. This is also the name of the AWS IoT thing.

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "ConnectivityInfo": [
    {
      "HostAddress": "string",
```

```
    "Id": "string",
    "Metadata": "string",
    "PortNumber": number
  }
],
"Message": "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.

ConnectivityInfo

The connectivity information for the core device.

Type: Array of [ConnectivityInfo](#) objects

Message

A message about the connectivity information request.

Type: String

Errors

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

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

GetCoreDevice

Retrieves metadata for a Greengrass core device.

Note

AWS IoT Greengrass relies on individual devices to send status updates to the AWS Cloud. If the AWS IoT Greengrass Core software isn't running on the device, or if device isn't connected to the AWS Cloud, then the reported status of that device might not reflect its current status. The status timestamp indicates when the device status was last updated. Core devices send status updates at the following times:

- When the AWS IoT Greengrass Core software starts
- When the core device receives a deployment from the AWS Cloud
- When the status of any component on the core device becomes BROKEN
- At a [regular interval that you can configure](#), which defaults to 24 hours
- For AWS IoT Greengrass Core v2.7.0, the core device sends status updates upon local deployment and cloud deployment

Request Syntax

```
GET /greengrass/v2/coreDevices/coreDeviceThingName HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

coreDeviceThingName

The name of the core device. This is also the name of the AWS IoT thing.

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "architecture": "string",
  "coreDeviceThingName": "string",
  "coreVersion": "string",
  "lastStatusUpdateTimestamp": number,
  "platform": "string",
  "status": "string",
  "tags": {
    "string" : "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.

architecture

The computer architecture of the core device.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

coreDeviceThingName

The name of the core device. This is also the name of the AWS IoT thing.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

coreVersion

The version of the AWS IoT Greengrass Core software that the core device runs. This version is equivalent to the version of the Greengrass nucleus component that runs on the core device. For more information, see the [Greengrass nucleus component](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

lastStatusUpdateTimestamp

The time at which the core device's status last updated, expressed in ISO 8601 format.

Type: Timestamp

platform

The operating system platform that the core device runs.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

status

The status of the core device. The core device status can be:

- HEALTHY – The AWS IoT Greengrass Core software and all components run on the core device without issue.
- UNHEALTHY – The AWS IoT Greengrass Core software or a component is in a failed state on the core device.

Type: String

Valid Values: HEALTHY | UNHEALTHY

tags

A list of key-value pairs that contain metadata for the resource. For more information, see [Tag your resources](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String to string map

Map Entries: Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Value Length Constraints: Maximum length of 256.

Errors

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

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

GetDeployment

Gets a deployment. Deployments define the components that run on Greengrass core devices.

Request Syntax

```
GET /greengrass/v2/deployments/deploymentId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

deploymentId

The ID of the deployment.

Length Constraints: Minimum length of 1.

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "components": {
    "string" : {
      "componentVersion": "string",
      "configurationUpdate": {
        "merge": "string",
        "reset": [ "string ]
      },
      "runWith": {
        "posixUser": "string",
        "systemResourceLimits": {
          "cpus": number,
```

```

        "memory": number
      },
      "windowsUser": "string"
    }
  }
},
"creationTimestamp": number,
"deploymentId": "string",
"deploymentName": "string",
"deploymentPolicies": {
  "componentUpdatePolicy": {
    "action": "string",
    "timeoutInSeconds": number
  },
  "configurationValidationPolicy": {
    "timeoutInSeconds": number
  },
  "failureHandlingPolicy": "string"
},
"deploymentStatus": "string",
"iotJobArn": "string",
"iotJobConfiguration": {
  "abortConfig": {
    "criteriaList": [
      {
        "action": "string",
        "failureType": "string",
        "minNumberOfExecutedThings": number,
        "thresholdPercentage": number
      }
    ]
  }
},
"jobExecutionsRolloutConfig": {
  "exponentialRate": {
    "baseRatePerMinute": number,
    "incrementFactor": number,
    "rateIncreaseCriteria": {
      "numberOfNotifiedThings": number,
      "numberOfSucceededThings": number
    }
  },
  "maximumPerMinute": number
},
"timeoutConfig": {

```

```
    "inProgressTimeoutInMinutes": number
  }
},
"iotJobId": "string",
"isLatestForTarget": boolean,
"parentTargetArn": "string",
"revisionId": "string",
"tags": {
  "string" : "string"
},
"targetArn": "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.

components

The components to deploy. This is a dictionary, where each key is the name of a component, and each key's value is the version and configuration to deploy for that component.

Type: String to [ComponentDeploymentSpecification](#) object map

Key Length Constraints: Minimum length of 1.

creationTimestamp

The time at which the deployment was created, expressed in ISO 8601 format.

Type: Timestamp

deploymentId

The ID of the deployment.

Type: String

Length Constraints: Minimum length of 1.

deploymentName

The name of the deployment.

Type: String

deploymentPolicies

The deployment policies for the deployment. These policies define how the deployment updates components and handles failure.

Type: [DeploymentPolicies](#) object

deploymentStatus

The status of the deployment.

Type: String

Valid Values: ACTIVE | COMPLETED | CANCELED | FAILED | INACTIVE

iotJobArn

The [ARN](#) of the AWS IoT job that applies the deployment to target devices.

Type: String

Pattern: `arn:[^:]*:iot:[^:]+:[0-9]+:job/.+`

iotJobConfiguration

The job configuration for the deployment configuration. The job configuration specifies the rollout, timeout, and stop configurations for the deployment configuration.

Type: [DeploymentIoTJobConfiguration](#) object

iotJobId

The ID of the AWS IoT job that applies the deployment to target devices.

Type: String

isLatestForTarget

Whether or not the deployment is the latest revision for its target.

Type: Boolean

parentTargetArn

The parent deployment's target [ARN](#) within a subdeployment.

Type: String

Pattern: `arn:[^:]*:iot:[^:]*:[0-9]+:(thing|thinggroup)/.+`

revisionId

The revision number of the deployment.

Type: String

Length Constraints: Minimum length of 1.

tags

A list of key-value pairs that contain metadata for the resource. For more information, see [Tag your resources](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String to string map

Map Entries: Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Value Length Constraints: Maximum length of 256.

targetArn

The [ARN](#) of the target AWS IoT thing or thing group.

Type: String

Pattern: `arn:[^:]*:iot:[^:]*:[0-9]+:(thing|thinggroup)/.+`

Errors

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

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

GetServiceRoleForAccount

Gets the service role associated with AWS IoT Greengrass for your AWS account in this AWS Region. AWS IoT Greengrass uses this role to verify the identity of client devices and manage core device connectivity information. For more information, see [Greengrass service role](#) in the *AWS IoT Greengrass Version 2 Developer Guide*.

Request Syntax

```
GET /greengrass/servicerole HTTP/1.1
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "AssociatedAt": "string",
  "RoleArn": "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.

[AssociatedAt](#)

The time when the service role was associated with AWS IoT Greengrass for your AWS account in this AWS Region.

Type: String

RoleArn

The ARN of the service role that is associated with AWS IoT Greengrass for your AWS account in this AWS Region.

Type: String

Errors

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

InternalServerError

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

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](#)

ListClientDevicesAssociatedWithCoreDevice

Retrieves a paginated list of client devices that are associated with a core device.

Request Syntax

```
GET /greengrass/v2/coreDevices/coreDeviceThingName/associatedClientDevices?  
maxResults=maxResults&nextToken=nextToken HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

coreDeviceThingName

The name of the core device. This is also the name of the AWS IoT thing.

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

maxResults

The maximum number of results to be returned per paginated request.

Valid Range: Minimum value of 1. Maximum value of 100.

nextToken

The token to be used for the next set of paginated results.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200  
Content-type: application/json  
  
{
```

```
"associatedClientDevices": [  
  {  
    "associationTimestamp": number,  
    "thingName": "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.

associatedClientDevices

A list that describes the client devices that are associated with the core device.

Type: Array of [AssociatedClientDevice](#) objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

nextToken

The token for the next set of results, or null if there are no additional results.

Type: String

Errors

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

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

ListComponents

Retrieves a paginated list of component summaries. This list includes components that you have permission to view.

Request Syntax

```
GET /greengrass/v2/components?maxResults=maxResults&nextToken=nextToken&scope=scope
HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

maxResults

The maximum number of results to be returned per paginated request.

Valid Range: Minimum value of 1. Maximum value of 100.

nextToken

The token to be used for the next set of paginated results.

scope

The scope of the components to list.

Default: PRIVATE

Valid Values: PRIVATE | PUBLIC

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json
```



```
{
  "components": [
    {
      "arn": "string",
      "componentName": "string",
      "latestVersion": {
        "arn": "string",
        "componentVersion": "string",
        "creationTimestamp": number,
        "description": "string",
        "platforms": [
          {
            "attributes": {
              "string": "string"
            },
            "name": "string"
          }
        ],
        "publisher": "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.

components

A list that summarizes each component.

Type: Array of [Component](#) objects

nextToken

The token for the next set of results, or null if there are no additional results.

Type: String

Errors

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

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

ListComponentVersions

Retrieves a paginated list of all versions for a component. Greater versions are listed first.

Request Syntax

```
GET /greengrass/v2/components/arn/versions?maxResults=maxResults&nextToken=nextToken
HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

arn

The [ARN](#) of the component.

Pattern: `arn:[^:]*:greengrass:[^:]*:(aws|[0-9]+):components:[^:]+`

Required: Yes

maxResults

The maximum number of results to be returned per paginated request.

Valid Range: Minimum value of 1. Maximum value of 100.

nextToken

The token to be used for the next set of paginated results.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
```

```
"componentVersions": [  
  {  
    "arn": "string",  
    "componentName": "string",  
    "componentVersion": "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.

componentVersions

A list of versions that exist for the component.

Type: Array of [ComponentVersionListItem](#) objects

nextToken

The token for the next set of results, or null if there are no additional results.

Type: String

Errors

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

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerError

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

ListCoreDevices

Retrieves a paginated list of Greengrass core devices.

Note

AWS IoT Greengrass relies on individual devices to send status updates to the AWS Cloud. If the AWS IoT Greengrass Core software isn't running on the device, or if device isn't connected to the AWS Cloud, then the reported status of that device might not reflect its current status. The status timestamp indicates when the device status was last updated. Core devices send status updates at the following times:

Core devices send status updates at the following times:

- When the AWS IoT Greengrass Core software starts
- When the core device receives a deployment from the AWS Cloud
- For Greengrass nucleus 2.12.2 and earlier, the core device sends status updates when the status of any component on the core device becomes `ERRORED` or `BROKEN`.
- For Greengrass nucleus 2.12.3 and later, the core device sends status updates when the status of any component on the core device becomes `ERRORED`, `BROKEN`, `RUNNING`, or `FINISHED`.
- At a [regular interval that you can configure](#), which defaults to 24 hours
- For AWS IoT Greengrass Core v2.7.0, the core device sends status updates upon local deployment and cloud deployment

Request Syntax

```
GET /greengrass/v2/coreDevices?  
maxResults=maxResults&nextToken=nextToken&status=status&thingGroupArn=thingGroupArn  
HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

maxResults

The maximum number of results to be returned per paginated request.

Valid Range: Minimum value of 1. Maximum value of 100.

nextToken

The token to be used for the next set of paginated results.

status

The core device status by which to filter. If you specify this parameter, the list includes only core devices that have this status. Choose one of the following options:

- HEALTHY – The AWS IoT Greengrass Core software and all components run on the core device without issue.
- UNHEALTHY – The AWS IoT Greengrass Core software or a component is in a failed state on the core device.

Valid Values: HEALTHY | UNHEALTHY

thingGroupArn

The [ARN](#) of the AWS IoT thing group by which to filter. If you specify this parameter, the list includes only core devices that have successfully deployed a deployment that targets the thing group. When you remove a core device from a thing group, the list continues to include that core device.

Pattern: `arn:[^:]*:iot:[^:]*:[0-9]+:thinggroup/.+`

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "coreDevices": [
    {
      "coreDeviceThingName": "string",
      "lastStatusUpdateTimestamp": number,
      "status": "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.

coreDevices

A list that summarizes each core device.

Type: Array of [CoreDevice](#) objects

nextToken

The token for the next set of results, or null if there are no additional results.

Type: String

Errors

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

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerError

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

ListDeployments

Retrieves a paginated list of deployments.

Request Syntax

```
GET /greengrass/v2/deployments?  
historyFilter=historyFilter&maxResults=maxResults&nextToken=nextToken&parentTargetArn=parentTargetArn  
HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

historyFilter

The filter for the list of deployments. Choose one of the following options:

- ALL – The list includes all deployments.
- LATEST_ONLY – The list includes only the latest revision of each deployment.

Default: LATEST_ONLY

Valid Values: ALL | LATEST_ONLY

maxResults

The maximum number of results to be returned per paginated request.

Default: 50

Valid Range: Minimum value of 1. Maximum value of 100.

nextToken

The token to be used for the next set of paginated results.

parentTargetArn

The parent deployment's target [ARN](#) within a subdeployment.

Pattern: arn:[^:]*:iot:[^:]*:[0-9]+:(thing|thinggroup)/.+

targetArn

The [ARN](#) of the target AWS IoT thing or thing group.

Pattern: `arn:[^:]*:iot:[^:]*:[0-9]+:(thing|thinggroup)/.+`

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "deployments": [
    {
      "creationTimestamp": number,
      "deploymentId": "string",
      "deploymentName": "string",
      "deploymentStatus": "string",
      "isLatestForTarget": boolean,
      "parentTargetArn": "string",
      "revisionId": "string",
      "targetArn": "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.

deployments

A list that summarizes each deployment.

Type: Array of [Deployment](#) objects

nextToken

The token for the next set of results, or null if there are no additional results.

Type: String

Errors

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

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerError

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

ListEffectiveDeployments

Retrieves a paginated list of deployment jobs that AWS IoT Greengrass sends to Greengrass core devices.

Request Syntax

```
GET /greengrass/v2/coreDevices/coreDeviceThingName/effectiveDeployments?  
maxResults=maxResults&nextToken=nextToken HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

coreDeviceThingName

The name of the core device. This is also the name of the AWS IoT thing.

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

maxResults

The maximum number of results to be returned per paginated request.

Valid Range: Minimum value of 1. Maximum value of 100.

nextToken

The token to be used for the next set of paginated results.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200  
Content-type: application/json
```

```
{
  "effectiveDeployments": [
    {
      "coreDeviceExecutionStatus": "string",
      "creationTimestamp": number,
      "deploymentId": "string",
      "deploymentName": "string",
      "description": "string",
      "iotJobArn": "string",
      "iotJobId": "string",
      "modifiedTimestamp": number,
      "reason": "string",
      "statusDetails": {
        "errorStack": [ "string" ],
        "errorTypes": [ "string" ]
      },
      "targetArn": "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.

effectiveDeployments

A list that summarizes each deployment on the core device.

Type: Array of [EffectiveDeployment](#) objects

nextToken

The token for the next set of results, or null if there are no additional results.

Type: String

Errors

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

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

ListInstalledComponents

Retrieves a paginated list of the components that a Greengrass core device runs. By default, this list doesn't include components that are deployed as dependencies of other components. To include dependencies in the response, set the `topologyFilter` parameter to ALL.

Note

AWS IoT Greengrass relies on individual devices to send status updates to the AWS Cloud. If the AWS IoT Greengrass Core software isn't running on the device, or if device isn't connected to the AWS Cloud, then the reported status of that device might not reflect its current status. The status timestamp indicates when the device status was last updated. Core devices send status updates at the following times:

- When the AWS IoT Greengrass Core software starts
- When the core device receives a deployment from the AWS Cloud
- When the status of any component on the core device becomes BROKEN
- At a [regular interval that you can configure](#), which defaults to 24 hours
- For AWS IoT Greengrass Core v2.7.0, the core device sends status updates upon local deployment and cloud deployment

Request Syntax

```
GET /greengrass/v2/coreDevices/coreDeviceThingName/installedComponents?  
maxResults=maxResults&nextToken=nextToken&topologyFilter=topologyFilter HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

coreDeviceThingName

The name of the core device. This is also the name of the AWS IoT thing.

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

maxResults

The maximum number of results to be returned per paginated request.

Valid Range: Minimum value of 1. Maximum value of 100.

nextToken

The token to be used for the next set of paginated results.

topologyFilter

The filter for the list of components. Choose from the following options:

- ALL – The list includes all components installed on the core device.
- ROOT – The list includes only *root* components, which are components that you specify in a deployment. When you choose this option, the list doesn't include components that the core device installs as dependencies of other components.

Default: ROOT

Valid Values: ALL | ROOT

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "installedComponents": [
    {
      "componentName": "string",
      "componentVersion": "string",
      "isRoot": boolean,
      "lastInstallationSource": "string",
      "lastReportedTimestamp": number,
      "lastStatusChangeTimestamp": number,
      "lifecycleState": "string",
      "lifecycleStateDetails": "string",
      "lifecycleStatusCodes": [ "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.

installedComponents

A list that summarizes each component on the core device.

Note

Greengrass nucleus v2.7.0 or later is required to get an accurate `lastStatusChangeTimestamp` response. This response can be inaccurate in earlier Greengrass nucleus versions.

Note

Greengrass nucleus v2.8.0 or later is required to get an accurate `lastInstallationSource` and `lastReportedTimestamp` response. This response can be inaccurate or null in earlier Greengrass nucleus versions.

Type: Array of [InstalledComponent](#) objects

nextToken

The token for the next set of results, or null if there are no additional results.

Type: String

Errors

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

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

ListTagsForResource

Retrieves the list of tags for an AWS IoT Greengrass resource.

Request Syntax

```
GET /tags/resourceArn HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

resourceArn

The [ARN](#) of the resource.

Pattern: `arn:[^:]*:greengrass:[^:]*:(aws|[0-9]+):(components|deployments|coreDevices):.*`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json
```

```
{
  "tags": {
    "string" : "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.

tags

A list of key-value pairs that contain metadata for the resource. For more information, see [Tag your resources](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String to string map

Map Entries: Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Value Length Constraints: Maximum length of 256.

Errors

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

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

ResolveComponentCandidates

Retrieves a list of components that meet the component, version, and platform requirements of a deployment. Greengrass core devices call this operation when they receive a deployment to identify the components to install.

This operation identifies components that meet all dependency requirements for a deployment. If the requirements conflict, then this operation returns an error and the deployment fails. For example, this occurs if component A requires version $>2.0.0$ and component B requires version $<2.0.0$ of a component dependency.

When you specify the component candidates to resolve, AWS IoT Greengrass compares each component's digest from the core device with the component's digest in the AWS Cloud. If the digests don't match, then AWS IoT Greengrass specifies to use the version from the AWS Cloud.

Important

To use this operation, you must use the data plane API endpoint and authenticate with an AWS IoT device certificate. For more information, see [AWS IoT Greengrass endpoints and quotas](#).

Request Syntax

```
POST /greengrass/v2/resolveComponentCandidates HTTP/1.1
Content-type: application/json
```

```
{
  "componentCandidates": [
    {
      "componentName": "string",
      "componentVersion": "string",
      "versionRequirements": {
        "string" : "string"
      }
    }
  ],
  "platform": {
    "attributes": {
      "string" : "string"
    }
  }
}
```

```
    },  
    "name": "string"  
  }  
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

componentCandidates

The list of components to resolve.

Type: Array of [ComponentCandidate](#) objects

Required: No

platform

The platform to use to resolve compatible components.

Type: [ComponentPlatform](#) object

Required: No

Response Syntax

```
HTTP/1.1 200  
Content-type: application/json  
  
{  
  "resolvedComponentVersions": [  
    {  
      "arn": "string",  
      "componentName": "string",  
      "componentVersion": "string",  
      "message": "string",  
      "recipe": blob,  
    }  
  ]  
}
```

```
    "vendorGuidance": "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.

resolvedComponentVersions

A list of components that meet the requirements that you specify in the request. This list includes each component's recipe that you can use to install the component.

Type: Array of [ResolvedComponentVersion](#) objects

Errors

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

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

ConflictException

Your request has conflicting operations. This can occur if you're trying to perform more than one operation on the same resource at the same time.

HTTP Status Code: 409

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

TagResource

Adds tags to an AWS IoT Greengrass resource. If a tag already exists for the resource, this operation updates the tag's value.

Request Syntax

```
POST /tags/resourceArn HTTP/1.1
Content-type: application/json
```

```
{
  "tags": {
    "string" : "string"
  }
}
```

URI Request Parameters

The request uses the following URI parameters.

resourceArn

The [ARN](#) of the resource to tag.

Pattern: `arn:[^:]*:greengrass:[^:]*:(aws|[0-9]+):(components|deployments|coreDevices):.*`

Required: Yes

Request Body

The request accepts the following data in JSON format.

tags

A list of key-value pairs that contain metadata for the resource. For more information, see [Tag your resources](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String to string map

Map Entries: Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Value Length Constraints: Maximum length of 256.

Required: Yes

Response Syntax

```
HTTP/1.1 200
```

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](#).

InternalServerError

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

UntagResource

Removes a tag from an AWS IoT Greengrass resource.

Request Syntax

```
DELETE /tags/resourceArn?tagKeys=tagKeys HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

resourceArn

The [ARN](#) of the resource to untag.

Pattern: `arn:[^:]*:greengrass:[^:]*:(aws|[0-9]+):(components|deployments|coreDevices):.*`

Required: Yes

tagKeys

A list of keys for tags to remove from the resource.

Array Members: Minimum number of 1 item. Maximum number of 200 items.

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
```

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](#).

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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](#)

UpdateConnectivityInfo

Updates connectivity information for a Greengrass core device.

Connectivity information includes endpoints and ports where client devices can connect to an MQTT broker on the core device. When a client device calls the [AWS IoT Greengrass discovery API](#), AWS IoT Greengrass returns connectivity information for all of the core devices where the client device can connect. For more information, see [Connect client devices to core devices](#) in the *AWS IoT Greengrass Version 2 Developer Guide*.

Request Syntax

```
PUT /greengrass/things/thingName/connectivityInfo HTTP/1.1
Content-type: application/json
```

```
{
  "ConnectivityInfo": [
    {
      "HostAddress": "string",
      "Id": "string",
      "Metadata": "string",
      "PortNumber": number
    }
  ]
}
```

URI Request Parameters

The request uses the following URI parameters.

thingName

The name of the core device. This is also the name of the AWS IoT thing.

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

Request Body

The request accepts the following data in JSON format.

ConnectivityInfo

The connectivity information for the core device.

Type: Array of [ConnectivityInfo](#) objects

Required: Yes

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "Message": "string",
  "Version": "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.

Message

A message about the connectivity information update request.

Type: String

Version

The new version of the connectivity information for the core device.

Type: String

Errors

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

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

HTTP Status Code: 400

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 AWS IoT Greengrass V2 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:

- [AssociateClientDeviceWithCoreDeviceEntry](#)
- [AssociateClientDeviceWithCoreDeviceErrorEntry](#)
- [AssociatedClientDevice](#)
- [CloudComponentStatus](#)
- [Component](#)
- [ComponentCandidate](#)
- [ComponentConfigurationUpdate](#)
- [ComponentDependencyRequirement](#)
- [ComponentDeploymentSpecification](#)
- [ComponentLatestVersion](#)
- [ComponentPlatform](#)
- [ComponentRunWith](#)
- [ComponentVersionListItem](#)
- [ConnectivityInfo](#)
- [CoreDevice](#)
- [Deployment](#)
- [DeploymentComponentUpdatePolicy](#)
- [DeploymentConfigurationValidationPolicy](#)
- [DeploymentIoTJobConfiguration](#)
- [DeploymentPolicies](#)

- [DisassociateClientDeviceFromCoreDeviceEntry](#)
- [DisassociateClientDeviceFromCoreDeviceErrorEntry](#)
- [EffectiveDeployment](#)
- [EffectiveDeploymentStatusDetails](#)
- [InstalledComponent](#)
- [IoTJobAbortConfig](#)
- [IoTJobAbortCriteria](#)
- [IoTJobExecutionsRolloutConfig](#)
- [IoTJobExponentialRolloutRate](#)
- [IoTJobRateIncreaseCriteria](#)
- [IoTJobTimeoutConfig](#)
- [LambdaContainerParams](#)
- [LambdaDeviceMount](#)
- [LambdaEventSource](#)
- [LambdaExecutionParameters](#)
- [LambdaFunctionRecipeSource](#)
- [LambdaLinuxProcessParams](#)
- [LambdaVolumeMount](#)
- [ResolvedComponentVersion](#)
- [SystemResourceLimits](#)
- [ValidationExceptionField](#)

AssociateClientDeviceWithCoreDeviceEntry

Contains a request to associate a client device with a core device. The [BatchAssociateClientDeviceWithCoreDevice](#) operation consumes a list of these requests.

Contents

thingName

The name of the AWS IoT thing that represents the client device to associate.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

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](#)

AssociateClientDeviceWithCoreDeviceErrorEntry

Contains an error that occurs from a request to associate a client device with a core device. The [BatchAssociateClientDeviceWithCoreDevice](#) operation returns a list of these errors.

Contents

code

The error code for the request.

Type: String

Length Constraints: Minimum length of 1.

Required: No

message

A message that provides additional information about the error.

Type: String

Length Constraints: Minimum length of 1.

Required: No

thingName

The name of the AWS IoT thing whose associate request failed.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

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](#)

AssociatedClientDevice

Contains information about a client device that is associated to a core device for cloud discovery.

Contents

associationTimestamp

The time that the client device was associated, expressed in ISO 8601 format.

Type: Timestamp

Required: No

thingName

The name of the AWS IoT thing that represents the associated client device.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

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](#)

CloudComponentStatus

Contains the status of a component version in the AWS IoT Greengrass service.

Contents

componentState

The state of the component version.

Type: String

Valid Values: REQUESTED | INITIATED | DEPLOYABLE | FAILED | DEPRECATED

Required: No

errors

A dictionary of errors that communicate why the component version is in an error state. For example, if AWS IoT Greengrass can't access an artifact for the component version, then `errors` contains the artifact's URI as a key, and the error message as the value for that key.

Type: String to string map

Key Length Constraints: Minimum length of 1.

Value Length Constraints: Minimum length of 1.

Required: No

message

A message that communicates details, such as errors, about the status of the component version.

Type: String

Length Constraints: Minimum length of 1.

Required: No

vendorGuidance

The vendor guidance state for the component version. This state indicates whether the component version has any issues that you should consider before you deploy it. The vendor guidance state can be:

- **ACTIVE** – This component version is available and recommended for use.
- **DISCONTINUED** – This component version has been discontinued by its publisher. You can deploy this component version, but we recommend that you use a different version of this component.
- **DELETED** – This component version has been deleted by its publisher, so you can't deploy it. If you have any existing deployments that specify this component version, those deployments will fail.

Type: String

Valid Values: ACTIVE | DISCONTINUED | DELETED

Required: No

vendorGuidanceMessage

A message that communicates details about the vendor guidance state of the component version. This message communicates why a component version is discontinued or deleted.

Type: String

Length Constraints: Minimum length of 1.

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](#)

Component

Contains information about a component.

Contents

arn

The [ARN](#) of the component version.

Type: String

Pattern: `arn:[^:]*:greengrass:[^:]*:(aws|[0-9]+):components:[^:]+`

Required: No

componentName

The name of the component.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: No

latestVersion

The latest version of the component and its details.

Type: [ComponentLatestVersion](#) 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](#)

ComponentCandidate

Contains information about a component that is a candidate to deploy to a Greengrass core device.

Contents

componentName

The name of the component.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: No

componentVersion

The version of the component.

Type: String

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

Required: No

versionRequirements

The version requirements for the component's dependencies. Greengrass core devices get the version requirements from component recipes.

AWS IoT Greengrass V2 uses semantic version constraints. For more information, see [Semantic Versioning](#).

Type: String to string map

Key Length Constraints: Minimum length of 1.

Value Length Constraints: Minimum length of 1.

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](#)

ComponentConfigurationUpdate

Contains information about a deployment's update to a component's configuration on Greengrass core devices. For more information, see [Update component configurations](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Contents

merge

A serialized JSON string that contains the configuration object to merge to target devices. The core device merges this configuration with the component's existing configuration. If this is the first time a component deploys on a device, the core device merges this configuration with the component's default configuration. This means that the core device keeps its existing configuration for keys and values that you don't specify in this object. For more information, see [Merge configuration updates](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 10485760.

Required: No

reset

The list of configuration nodes to reset to default values on target devices. Use JSON pointers to specify each node to reset. JSON pointers start with a forward slash (/) and use forward slashes to separate the key for each level in the object. For more information, see the [JSON pointer specification](#) and [Reset configuration updates](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: Array of strings

Length Constraints: Minimum length of 0. Maximum length of 256.

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](#)

ComponentDependencyRequirement

Contains information about a component dependency for a Lambda function component.

Contents

dependencyType

The type of this dependency. Choose from the following options:

- SOFT – The component doesn't restart if the dependency changes state.
- HARD – The component restarts if the dependency changes state.

Default: HARD

Type: String

Valid Values: HARD | SOFT

Required: No

versionRequirement

The component version requirement for the component dependency.

AWS IoT Greengrass V2 uses semantic version constraints. For more information, see [Semantic Versioning](#).

Type: String

Length Constraints: Minimum length of 1.

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](#)

ComponentDeploymentSpecification

Contains information about a component to deploy.

Contents

componentVersion

The version of the component.

Type: String

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

Required: Yes

configurationUpdate

The configuration updates to deploy for the component. You can define *reset* updates and *merge* updates. A reset updates the keys that you specify to the default configuration for the component. A merge updates the core device's component configuration with the keys and values that you specify. The AWS IoT Greengrass Core software applies reset updates before it applies merge updates. For more information, see [Update component configurations](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: [ComponentConfigurationUpdate](#) object

Required: No

runWith

The system user and group that the AWS IoT Greengrass Core software uses to run component processes on the core device. If you omit this parameter, the AWS IoT Greengrass Core software uses the system user and group that you configure for the core device. For more information, see [Configure the user and group that run components](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: [ComponentRunWith](#) 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](#)

ComponentLatestVersion

Contains information about the latest version of a component.

Contents

arn

The [ARN](#) of the component version.

Type: String

Pattern: `arn:[^:]*:greengrass:[^:]*:(aws|[0-9]+):components:[^:]+:versions:[^:]+`

Required: No

componentVersion

The version of the component.

Type: String

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

Required: No

creationTimestamp

The time at which the component was created, expressed in ISO 8601 format.

Type: Timestamp

Required: No

description

The description of the component version.

Type: String

Length Constraints: Minimum length of 1.

Required: No

platforms

The platforms that the component version supports.

Type: Array of [ComponentPlatform](#) objects

Required: No

publisher

The publisher of the component version.

Type: String

Length Constraints: Minimum length of 1.

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](#)

ComponentPlatform

Contains information about a platform that a component supports.

Contents

attributes

A dictionary of attributes for the platform. The AWS IoT Greengrass Core software defines the `os` and `architecture` by default. You can specify additional platform attributes for a core device when you deploy the Greengrass nucleus component. For more information, see the [Greengrass nucleus component](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String to string map

Key Length Constraints: Minimum length of 1.

Value Length Constraints: Minimum length of 1.

Required: No

name

The friendly name of the platform. This name helps you identify the platform.

If you omit this parameter, AWS IoT Greengrass creates a friendly name from the `os` and `architecture` of the platform.

Type: String

Length Constraints: Minimum length of 1.

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](#)

ComponentRunWith

Contains information system user and group that the AWS IoT Greengrass Core software uses to run component processes on the core device. For more information, see [Configure the user and group that run components](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Contents

posixUser

The POSIX system user and, optionally, group to use to run this component on Linux core devices. The user, and group if specified, must exist on each Linux core device. Specify the user and group separated by a colon (:) in the following format: `user:group`. The group is optional. If you don't specify a group, the AWS IoT Greengrass Core software uses the primary user for the group.

If you omit this parameter, the AWS IoT Greengrass Core software uses the default system user and group that you configure on the Greengrass nucleus component. For more information, see [Configure the user and group that run components](#).

Type: String

Length Constraints: Minimum length of 1.

Required: No

systemResourceLimits

The system resource limits to apply to this component's process on the core device. AWS IoT Greengrass currently supports this feature on only Linux core devices.

If you omit this parameter, the AWS IoT Greengrass Core software uses the default system resource limits that you configure on the Greengrass nucleus component. For more information, see [Configure system resource limits for components](#).

Type: [SystemResourceLimits](#) object

Required: No

windowsUser

The Windows user to use to run this component on Windows core devices. The user must exist on each Windows core device, and its name and password must be in the LocalSystem account's Credentials Manager instance.

If you omit this parameter, the AWS IoT Greengrass Core software uses the default Windows user that you configure on the Greengrass nucleus component. For more information, see [Configure the user and group that run components](#).

Type: String

Length Constraints: Minimum length of 1.

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](#)

ComponentVersionListItem

Contains information about a component version in a list.

Contents

arn

The [ARN](#) of the component version.

Type: String

Length Constraints: Minimum length of 1.

Required: No

componentName

The name of the component.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: No

componentVersion

The version of the component.

Type: String

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

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](#)

ConnectivityInfo

Contains information about an endpoint and port where client devices can connect to an MQTT broker on a Greengrass core device.

Contents

HostAddress

The IP address or DNS address where client devices can connect to an MQTT broker on the Greengrass core device.

Type: String

Required: No

Id

An ID for the connectivity information.

Type: String

Required: No

Metadata

Additional metadata to provide to client devices that connect to this core device.

Type: String

Required: No

PortNumber

The port where the MQTT broker operates on the core device. This port is typically 8883, which is the default port for the MQTT broker component that runs on core devices.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 65535.

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](#)

CoreDevice

Contains information about a Greengrass core device, which is an AWS IoT thing that runs the AWS IoT Greengrass Core software.

Contents

coreDeviceThingName

The name of the core device. This is also the name of the AWS IoT thing.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: No

lastStatusUpdateTimestamp

The time at which the core device's status last updated, expressed in ISO 8601 format.

Type: Timestamp

Required: No

status

The status of the core device. Core devices can have the following statuses:

- HEALTHY – The AWS IoT Greengrass Core software and all components run on the core device without issue.
- UNHEALTHY – The AWS IoT Greengrass Core software or a component is in a failed state on the core device.

Type: String

Valid Values: HEALTHY | UNHEALTHY

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](#)

Deployment

Contains information about a deployment.

Contents

creationTimestamp

The time at which the deployment was created, expressed in ISO 8601 format.

Type: Timestamp

Required: No

deploymentId

The ID of the deployment.

Type: String

Length Constraints: Minimum length of 1.

Required: No

deploymentName

The name of the deployment.

Type: String

Length Constraints: Minimum length of 1.

Required: No

deploymentStatus

The status of the deployment.

Type: String

Valid Values: ACTIVE | COMPLETED | CANCELED | FAILED | INACTIVE

Required: No

isLatestForTarget

Whether or not the deployment is the latest revision for its target.

Type: Boolean

Required: No

parentTargetArn

The parent deployment's target [ARN](#) within a subdeployment.

Type: String

Pattern: `arn:[^:]*:iot:[^:]*:[0-9]+:(thing|thinggroup)/.+`

Required: No

revisionId

The revision number of the deployment.

Type: String

Length Constraints: Minimum length of 1.

Required: No

targetArn

The [ARN](#) of the target AWS IoT thing or thing group. When creating a subdeployment, the targetARN can only be a thing group.

Type: String

Pattern: `arn:[^:]*:iot:[^:]*:[0-9]+:(thing|thinggroup)/.+`

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](#)

DeploymentComponentUpdatePolicy

Contains information about a deployment's policy that defines when components are safe to update.

Each component on a device can report whether or not it's ready to update. After a component and its dependencies are ready, they can apply the update in the deployment. You can configure whether or not the deployment notifies components of an update and waits for a response. You specify the amount of time each component has to respond to the update notification.

Contents

action

Whether or not to notify components and wait for components to become safe to update. Choose from the following options:

- `NOTIFY_COMPONENTS` – The deployment notifies each component before it stops and updates that component. Components can use the [SubscribeToComponentUpdates](#) IPC operation to receive these notifications. Then, components can respond with the [DeferComponentUpdate](#) IPC operation. For more information, see [Create deployments](#) in the *AWS IoT Greengrass V2 Developer Guide*.
- `SKIP_NOTIFY_COMPONENTS` – The deployment doesn't notify components or wait for them to be safe to update.

Default: `NOTIFY_COMPONENTS`

Type: String

Valid Values: `NOTIFY_COMPONENTS` | `SKIP_NOTIFY_COMPONENTS`

Required: No

timeoutInSeconds

The amount of time in seconds that each component on a device has to report that it's safe to update. If the component waits for longer than this timeout, then the deployment proceeds on the device.

Default: 60

Type: Integer

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](#)

DeploymentConfigurationValidationPolicy

Contains information about how long a component on a core device can validate its configuration updates before it times out. Components can use the [SubscribeToValidateConfigurationUpdates](#) IPC operation to receive notifications when a deployment specifies a configuration update. Then, components can respond with the [SendConfigurationValidityReport](#) IPC operation. For more information, see [Create deployments](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Contents

timeoutInSeconds

The amount of time in seconds that a component can validate its configuration updates. If the validation time exceeds this timeout, then the deployment proceeds for the device.

Default: 30

Type: Integer

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](#)

DeploymentIoTJobConfiguration

Contains information about an AWS IoT job configuration.

Contents

abortConfig

The stop configuration for the job. This configuration defines when and how to stop a job rollout.

Type: [IoTJobAbortConfig](#) object

Required: No

jobExecutionsRolloutConfig

The rollout configuration for the job. This configuration defines the rate at which the job rolls out to the fleet of target devices.

Type: [IoTJobExecutionsRolloutConfig](#) object

Required: No

timeoutConfig

The timeout configuration for the job. This configuration defines the amount of time each device has to complete the job.

Type: [IoTJobTimeoutConfig](#) 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](#)

DeploymentPolicies

Contains information about policies that define how a deployment updates components and handles failure.

Contents

componentUpdatePolicy

The component update policy for the configuration deployment. This policy defines when it's safe to deploy the configuration to devices.

Type: [DeploymentComponentUpdatePolicy](#) object

Required: No

configurationValidationPolicy

The configuration validation policy for the configuration deployment. This policy defines how long each component has to validate its configure updates.

Type: [DeploymentConfigurationValidationPolicy](#) object

Required: No

failureHandlingPolicy

The failure handling policy for the configuration deployment. This policy defines what to do if the deployment fails.

Default: ROLLBACK

Type: String

Valid Values: ROLLBACK | DO_NOTHING

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](#)

DisassociateClientDeviceFromCoreDeviceEntry

Contains a request to disassociate a client device from a core device. The [BatchDisassociateClientDeviceWithCoreDevice](#) operation consumes a list of these requests.

Contents

thingName

The name of the AWS IoT thing that represents the client device to disassociate.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

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](#)

DisassociateClientDeviceFromCoreDeviceErrorEntry

Contains an error that occurs from a request to disassociate a client device from a core device. The [BatchDisassociateClientDeviceWithCoreDevice](#) operation returns a list of these errors.

Contents

code

The error code for the request.

Type: String

Length Constraints: Minimum length of 1.

Required: No

message

A message that provides additional information about the error.

Type: String

Length Constraints: Minimum length of 1.

Required: No

thingName

The name of the AWS IoT thing whose disassociate request failed.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

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](#)

EffectiveDeployment

Contains information about a deployment job that AWS IoT Greengrass sends to a Greengrass core device.

Contents

coreDeviceExecutionStatus

The status of the deployment job on the Greengrass core device.

- **IN_PROGRESS** – The deployment job is running.
- **QUEUED** – The deployment job is in the job queue and waiting to run.
- **FAILED** – The deployment failed. For more information, see the `statusDetails` field.
- **COMPLETED** – The deployment to an AWS IoT thing was completed successfully.
- **TIMED_OUT** – The deployment didn't complete in the allotted time.
- **CANCELED** – The deployment was canceled by the user.
- **REJECTED** – The deployment was rejected. For more information, see the `statusDetails` field.
- **SUCCEEDED** – The deployment to an AWS IoT thing group was completed successfully.

Type: String

Valid Values: `IN_PROGRESS` | `QUEUED` | `FAILED` | `COMPLETED` | `TIMED_OUT` | `CANCELED` | `REJECTED` | `SUCCEEDED`

Required: Yes

creationTimestamp

The time at which the deployment was created, expressed in ISO 8601 format.

Type: Timestamp

Required: Yes

deploymentId

The ID of the deployment.

Type: String

Required: Yes

deploymentName

The name of the deployment.

Type: String

Required: Yes

modifiedTimestamp

The time at which the deployment job was last modified, expressed in ISO 8601 format.

Type: Timestamp

Required: Yes

targetArn

The [ARN](#) of the target AWS IoT thing or thing group.

Type: String

Pattern: `arn:[^:]*:iot:[^:]*:[0-9]+:(thing|thinggroup)/.+`

Required: Yes

description

The description of the deployment job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 500.

Required: No

iotJobArn

The [ARN](#) of the AWS IoT job that applies the deployment to target devices.

Type: String

Pattern: `arn:[^:]*:iot:[^:]*:[0-9]+:job/.+`

Required: No

iotJobId

The ID of the AWS IoT job that applies the deployment to target devices.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: No

reason

The reason code for the update, if the job was updated.

Type: String

Required: No

statusDetails

The status details that explain why a deployment has an error. This response will be null if the deployment is in a success state.

Type: [EffectiveDeploymentStatusDetails](#) 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](#)

EffectiveDeploymentStatusDetails

Contains all error-related information for the deployment record. The status details will be null if the deployment is in a success state.

Note

Greengrass nucleus v2.8.0 or later is required to get an accurate `errorStack` and `errorTypes` response. This field will not be returned for earlier Greengrass nucleus versions.

Contents

errorStack

Contains an ordered list of short error codes that range from the most generic error to the most specific one. The error codes describe the reason for failure whenever the `coreDeviceExecutionStatus` is in a failed state. The response will be an empty list if there is no error.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: No

errorTypes

Contains tags which describe the error. You can use the error types to classify errors to assist with remediating the failure. The response will be an empty list if there is no error.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 255.

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](#)

InstalledComponent

Contains information about a component on a Greengrass core device.

Contents

componentName

The name of the component.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: No

componentVersion

The version of the component.

Type: String

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

Required: No

isRoot

Whether or not the component is a root component.

Type: Boolean

Required: No

lastInstallationSource

The most recent deployment source that brought the component to the Greengrass core device. For a thing group deployment or thing deployment, the source will be the ID of the last deployment that contained the component. For local deployments it will be LOCAL.

Note

Any deployment will attempt to reinstall currently broken components on the device, which will update the last installation source.

Type: String

Length Constraints: Minimum length of 1.

Required: No

lastReportedTimestamp

The last time the Greengrass core device sent a message containing a component's state to the AWS Cloud.

A component does not need to see a state change for this field to update.

Type: Timestamp

Required: No

lastStatusChangeTimestamp

The status of how current the data is.

This response is based off of component state changes. The status reflects component disruptions and deployments. If a component only sees a configuration update during a deployment, it might not undergo a state change and this status would not be updated.

Type: Timestamp

Required: No

lifecycleState

The lifecycle state of the component.

Type: String

Valid Values: NEW | INSTALLED | STARTING | RUNNING | STOPPING | ERRORED | BROKEN | FINISHED

Required: No

lifecycleStateDetails

A detailed response about the lifecycle state of the component that explains the reason why a component has an error or is broken.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1000.

Required: No

lifecycleStatusCodes

The status codes that indicate the reason for failure whenever the `lifecycleState` has an error or is in a broken state.

Note

Greengrass nucleus v2.8.0 or later is required to get an accurate `lifecycleStatusCodes` response. This response can be inaccurate in earlier Greengrass nucleus versions.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 255.

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](#)

IoTJobAbortConfig

Contains a list of criteria that define when and how to cancel a configuration deployment.

Contents

criteriaList

The list of criteria that define when and how to cancel the configuration deployment.

Type: Array of [IoTJobAbortCriteria](#) objects

Array Members: Minimum number of 1 item.

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](#)

IoTJobAbortCriteria

Contains criteria that define when and how to cancel a job.

The deployment stops if the following conditions are true:

1. The number of things that receive the deployment exceeds the `minNumberOfExecutedThings`.
2. The percentage of failures with type `failureType` exceeds the `thresholdPercentage`.

Contents

action

The action to perform when the criteria are met.

Type: String

Valid Values: CANCEL

Required: Yes

failureType

The type of job deployment failure that can cancel a job.

Type: String

Valid Values: FAILED | REJECTED | TIMED_OUT | ALL

Required: Yes

minNumberOfExecutedThings

The minimum number of things that receive the configuration before the job can cancel.

Type: Integer

Valid Range: Minimum value of 1.

Required: Yes

thresholdPercentage

The minimum percentage of `failureType` failures that occur before the job can cancel.

This parameter supports up to two digits after the decimal (for example, you can specify `10.9` or `10.99`, but not `10.999`).

Type: Double

Valid Range: Maximum value of 100.

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](#)

IoTJobExecutionsRolloutConfig

Contains information about the rollout configuration for a job. This configuration defines the rate at which the job deploys a configuration to a fleet of target devices.

Contents

exponentialRate

The exponential rate to increase the job rollout rate.

Type: [IoTJobExponentialRolloutRate](#) object

Required: No

maximumPerMinute

The maximum number of devices that receive a pending job notification, per minute.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

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](#)

IoTJobExponentialRolloutRate

Contains information about an exponential rollout rate for a configuration deployment job.

Contents

baseRatePerMinute

The minimum number of devices that receive a pending job notification, per minute, when the job starts. This parameter defines the initial rollout rate of the job.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: Yes

incrementFactor

The exponential factor to increase the rollout rate for the job.

This parameter supports up to one digit after the decimal (for example, you can specify 1.5, but not 1.55).

Type: Double

Valid Range: Minimum value of 1. Maximum value of 5.

Required: Yes

rateIncreaseCriteria

The criteria to increase the rollout rate for the job.

Type: [IoTJobRateIncreaseCriteria](#) 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](#)

IoTJobRateIncreaseCriteria

Contains information about criteria to meet before a job increases its rollout rate. Specify either `numberOfNotifiedThings` or `numberOfSucceededThings`.

Contents

`numberOfNotifiedThings`

The number of devices to receive the job notification before the rollout rate increases.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

`numberOfSucceededThings`

The number of devices to successfully run the configuration job before the rollout rate increases.

Type: Integer

Valid Range: Minimum value of 1.

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](#)

IoTJobTimeoutConfig

Contains information about the timeout configuration for a job.

Contents

inProgressTimeoutInMinutes

The amount of time, in minutes, that devices have to complete the job. The timer starts when the job status is set to `IN_PROGRESS`. If the job status doesn't change to a terminal state before the time expires, then the job status is set to `TIMED_OUT`.

The timeout interval must be between 1 minute and 7 days (10080 minutes).

Type: Long

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](#)

LambdaContainerParams

Contains information about a container in which AWS Lambda functions run on Greengrass core devices.

Contents

devices

The list of system devices that the container can access.

Type: Array of [LambdaDeviceMount](#) objects

Required: No

memorySizeInKB

The memory size of the container, expressed in kilobytes.

Default: 16384 (16 MB)

Type: Integer

Required: No

mountROSysfs

Whether or not the container can read information from the device's /sys folder.

Default: false

Type: Boolean

Required: No

volumes

The list of volumes that the container can access.

Type: Array of [LambdaVolumeMount](#) 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](#)

LambdaDeviceMount

Contains information about a device that Linux processes in a container can access.

Contents

path

The mount path for the device in the file system.

Type: String

Required: Yes

addGroupOwner

Whether or not to add the component's system user as an owner of the device.

Default: false

Type: Boolean

Required: No

permission

The permission to access the device: read/only (ro) or read/write (rw).

Default: ro

Type: String

Valid Values: ro | rw

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](#)

LambdaEventSource

Contains information about an event source for an AWS Lambda function. The event source defines the topics on which this Lambda function subscribes to receive messages that run the function.

Contents

topic

The topic to which to subscribe to receive event messages.

Type: String

Required: Yes

type

The type of event source. Choose from the following options:

- `PUB_SUB` – Subscribe to local publish/subscribe messages. This event source type doesn't support MQTT wildcards (+ and #) in the event source topic.
- `IOT_CORE` – Subscribe to AWS IoT Core MQTT messages. This event source type supports MQTT wildcards (+ and #) in the event source topic.

Type: String

Valid Values: `PUB_SUB` | `IOT_CORE`

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](#)

LambdaExecutionParameters

Contains parameters for a Lambda function that runs on AWS IoT Greengrass.

Contents

environmentVariables

The map of environment variables that are available to the Lambda function when it runs.

Type: String to string map

Key Length Constraints: Minimum length of 1.

Required: No

eventSources

The list of event sources to which to subscribe to receive work messages. The Lambda function runs when it receives a message from an event source. You can subscribe this function to local publish/subscribe messages and AWS IoT Core MQTT messages.

Type: Array of [LambdaEventSource](#) objects

Required: No

execArgs

The list of arguments to pass to the Lambda function when it runs.

Type: Array of strings

Required: No

inputPayloadEncodingType

The encoding type that the Lambda function supports.

Default: json

Type: String

Valid Values: json | binary

Required: No

linuxProcessParams

The parameters for the Linux process that contains the Lambda function.

Type: [LambdaLinuxProcessParams](#) object

Required: No

maxIdleTimeInSeconds

The maximum amount of time in seconds that a non-pinned Lambda function can idle before the AWS IoT Greengrass Core software stops its process.

Type: Integer

Required: No

maxInstancesCount

The maximum number of instances that a non-pinned Lambda function can run at the same time.

Type: Integer

Required: No

maxQueueSize

The maximum size of the message queue for the Lambda function component. The AWS IoT Greengrass core stores messages in a FIFO (first-in-first-out) queue until it can run the Lambda function to consume each message.

Type: Integer

Required: No

pinned

Whether or not the Lambda function is pinned, or long-lived.

- A pinned Lambda function starts when AWS IoT Greengrass starts and keeps running in its own container.
- A non-pinned Lambda function starts only when it receives a work item and exists after it idles for `maxIdleTimeInSeconds`. If the function has multiple work items, the AWS IoT Greengrass Core software creates multiple instances of the function.

Default: true

Type: Boolean

Required: No

statusTimeoutInSeconds

The interval in seconds at which a pinned (also known as long-lived) Lambda function component sends status updates to the Lambda manager component.

Type: Integer

Required: No

timeoutInSeconds

The maximum amount of time in seconds that the Lambda function can process a work item.

Type: Integer

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](#)

LambdaFunctionRecipeSource

Contains information about an AWS Lambda function to import to create a component.

Contents

lambdaArn

The [ARN](#) of the Lambda function. The ARN must include the version of the function to import. You can't use version aliases like \$LATEST.

Type: String

Length Constraints: Minimum length of 1.

Required: Yes

componentDependencies

The component versions on which this Lambda function component depends.

Type: String to [ComponentDependencyRequirement](#) object map

Key Length Constraints: Minimum length of 1.

Required: No

componentLambdaParameters

The system and runtime parameters for the Lambda function as it runs on the Greengrass core device.

Type: [LambdaExecutionParameters](#) object

Required: No

componentName

The name of the component.

Defaults to the name of the Lambda function.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: No

componentPlatforms

The platforms that the component version supports.

Type: Array of [ComponentPlatform](#) objects

Required: No

componentVersion

The version of the component.

Defaults to the version of the Lambda function as a semantic version. For example, if your function version is 3, the component version becomes 3.0.0.

Type: String

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

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](#)

LambdaLinuxProcessParams

Contains parameters for a Linux process that contains an AWS Lambda function.

Contents

containerParams

The parameters for the container in which the Lambda function runs.

Type: [LambdaContainerParams](#) object

Required: No

isolationMode

The isolation mode for the process that contains the Lambda function. The process can run in an isolated runtime environment inside the AWS IoT Greengrass container, or as a regular process outside any container.

Default: GreengrassContainer

Type: String

Valid Values: GreengrassContainer | NoContainer

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](#)

LambdaVolumeMount

Contains information about a volume that Linux processes in a container can access. When you define a volume, the AWS IoT Greengrass Core software mounts the source files to the destination inside the container.

Contents

destinationPath

The path to the logical volume in the file system.

Type: String

Required: Yes

sourcePath

The path to the physical volume in the file system.

Type: String

Required: Yes

addGroupOwner

Whether or not to add the AWS IoT Greengrass user group as an owner of the volume.

Default: false

Type: Boolean

Required: No

permission

The permission to access the volume: read/only (ro) or read/write (rw).

Default: ro

Type: String

Valid Values: ro | rw

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](#)

ResolvedComponentVersion

Contains information about a component version that is compatible to run on a Greengrass core device.

Contents

arn

The [ARN](#) of the component version.

Type: String

Pattern: `arn:[^:]*:greengrass:[^:]*:(aws|[0-9]+):components:[^:]*:versions:[^:]*`

Required: No

componentName

The name of the component.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: No

componentVersion

The version of the component.

Type: String

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

Required: No

message

A message that communicates details about the vendor guidance state of the component version. This message communicates why a component version is discontinued or deleted.

Type: String

Length Constraints: Minimum length of 1.

Required: No

recipe

The recipe of the component version.

Type: Base64-encoded binary data object

Required: No

vendorGuidance

The vendor guidance state for the component version. This state indicates whether the component version has any issues that you should consider before you deploy it. The vendor guidance state can be:

- **ACTIVE** – This component version is available and recommended for use.
- **DISCONTINUED** – This component version has been discontinued by its publisher. You can deploy this component version, but we recommend that you use a different version of this component.
- **DELETED** – This component version has been deleted by its publisher, so you can't deploy it. If you have any existing deployments that specify this component version, those deployments will fail.

Type: String

Valid Values: ACTIVE | DISCONTINUED | DELETED

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](#)

SystemResourceLimits

Contains information about system resource limits that the AWS IoT Greengrass Core software applies to a component's processes. For more information, see [Configure system resource limits for components](#).

Contents

cpus

The maximum amount of CPU time that a component's processes can use on the core device. A core device's total CPU time is equivalent to the device's number of CPU cores. For example, on a core device with 4 CPU cores, you can set this value to 2 to limit the component's processes to 50 percent usage of each CPU core. On a device with 1 CPU core, you can set this value to 0.25 to limit the component's processes to 25 percent usage of the CPU. If you set this value to a number greater than the number of CPU cores, the AWS IoT Greengrass Core software doesn't limit the component's CPU usage.

Type: Double

Valid Range: Minimum value of 0.

Required: No

memory

The maximum amount of RAM, expressed in kilobytes, that a component's processes can use on the core device.

Type: Long

Valid Range: Minimum value of 0. Maximum value of 9223372036854771712.

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

Contains information about a validation exception field.

Contents

message

The message of the exception field.

Type: String

Required: Yes

name

The name of the exception field.

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](#)

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