# Table of Contents

Greengrass API .......................................................................................................................... 1

Actions ........................................................................................................................................ 1

AssociateRoleToGroup ............................................................................................................. 3
AssociateServiceRoleToAccount ............................................................................................... 6
CreateConnectorDefinition ...................................................................................................... 9
CreateConnectorDefinitionVersion .......................................................................................... 13
CreateCoreDefinition ............................................................................................................ 16
CreateCoreDefinitionVersion ............................................................................................... 20
CreateDeployment .................................................................................................................. 24
CreateDeviceDefinition ......................................................................................................... 26
CreateDeviceDefinitionVersion ............................................................................................. 30
CreateFunctionDefinition ...................................................................................................... 34
CreateFunctionDefinitionVersion ........................................................................................... 41
CreateGroup ............................................................................................................................ 47
CreateGroupCertificateAuthority ........................................................................................... 51
CreateGroupVersion ................................................................................................................ 54
CreateLoggerDefinition ......................................................................................................... 57
CreateLoggerDefinitionVersion ............................................................................................... 62
CreateResourceDefinition ...................................................................................................... 65
CreateResourceDefinitionVersion ............................................................................................ 73
CreateSoftwareUpdateJob ...................................................................................................... 80
CreateSubscriptionDefinition ................................................................................................. 84
CreateSubscriptionDefinitionVersion ...................................................................................... 88
DeleteConnectorDefinition ..................................................................................................... 92
DeleteCoreDefinition ............................................................................................................. 93
DeleteDeviceDefinition .......................................................................................................... 95
DeleteFunctionDefinition ....................................................................................................... 97
DeleteGroup ............................................................................................................................. 99
DeleteLoggerDefinition .......................................................................................................... 100
DeleteResourceDefinition ........................................................................................................ 102
DeleteSubscriptionDefinition ............................................................................................... 104
DisassociateRoleFromGroup .................................................................................................... 105
DisassociateServiceRoleFromAccount ..................................................................................... 108
GetAssociatedRole .................................................................................................................. 110
GetBulkDeploymentStatus ...................................................................................................... 112
GetConnectivityInfo ............................................................................................................... 115
GetConnectorDefinition ........................................................................................................ 119
GetConnectorDefinitionVersion ............................................................................................. 121
GetCoreDefinition .................................................................................................................. 125
GetCoreDefinitionVersion ...................................................................................................... 127
GetDeploymentStatus ............................................................................................................. 131
GetDeviceDefinition ............................................................................................................... 133
GetDeviceDefinitionVersion ................................................................................................... 136
GetFunctionDefinition ........................................................................................................... 139
GetFunctionDefinitionVersion ............................................................................................... 142
GetGroup .................................................................................................................................. 148
GetGroupCertificateAuthority ............................................................................................... 151
GetGroupCertificateConfiguration .......................................................................................... 154
GetGroupVersion .................................................................................................................... 156
GetLoggerDefinition ............................................................................................................... 160
GetLoggerDefinitionVersion ................................................................................................. 162
GetResourceDefinition .......................................................................................................... 166
GetResourceDefinitionVersion ............................................................................................... 168
GetServiceRoleForAccount ...................................................................................................... 175
Parameters

AssociateRoleToGroupRequestBody ................................................................. 544
AssociateServiceRoleToAccountRequestBody .................................................. 545
BulkDeploymentId ................................................................................................. 545
CertificateAuthorityId ......................................................................................... 546
ConnectorDefinitionId ......................................................................................... 546
ConnectorDefinitionVersionId .............................................................................. 546
CoreDefinitionId .................................................................................................. 547
CoreDefinitionVersionId ...................................................................................... 547
CreateConnectorDefinitionRequestBody ............................................................ 547
CreateConnectorDefinitionVersionRequestBody ................................................. 548
CreateCoreDefinitionRequestBody ...................................................................... 549
CreateCoreDefinitionVersionRequestBody .......................................................... 550
CreateDeploymentRequestBody .......................................................................... 551
CreateDeviceDefinitionRequestBody ................................................................. 551
CreateDeviceDefinitionVersionRequestBody ...................................................... 552
CreateFunctionDefinitionRequestBody .............................................................. 553
CreateFunctionDefinitionVersionRequestBody .................................................. 554
CreateGroupRequestBody .................................................................................... 555
CreateGroupVersionRequestBody ........................................................................ 557
CreateLoggerDefinitionRequestBody .................................................................... 558
CreateLoggerDefinitionVersionRequestBody ...................................................... 559
CreateResourceDefinitionRequestBody ............................................................. 560
CreateResourceDefinitionVersionRequestBody .................................................. 561
CreateSubscriptionDefinitionRequestBody ...................................................... 562
CreateSubscriptionDefinitionVersionRequestBody ............................................ 563
CreateSoftwareUpdateJobRequestBody .............................................................. 564
CreateSoftwareUpdateJobVersionRequestBody ................................................... 565
CreateUpdateDefinitionVersionRequestBody ...................................................... 566
CreateUpdateDefinitionRequestBody ................................................................. 567
CreateUpdateDefinitionVersionRequestBody ...................................................... 568
AWS IoT Greengrass provides a resource-based API that uses Hypertext Application Language (HAL). HAL provides a standard way for expressing the resources and relationships of an API as hyperlinks. Using HAL, you use HTTP methods (GET, PUT, POST, DELETE) to submit requests and receive information about the API in the response. Applications can use the information returned to explore the functionality of the API. For more information about HAL, see the JSON Hypertext Application Language draft.

To request a HAL response from AWS Greengrass, specify application/hal+json for the accept request header.

Contents
- Actions (p. 1)
- Endpoints (p. 269)
- Parameters (p. 543)
- Definitions (p. 579)

Actions

The following actions are supported:
- AssociateRoleToGroup (p. 3)
- AssociateServiceRoleToAccount (p. 6)
- CreateConnectorDefinition (p. 9)
- CreateConnectorDefinitionVersion (p. 13)
- CreateCoreDefinition (p. 16)
- CreateCoreDefinitionVersion (p. 20)
- CreateDeployment (p. 24)
- CreateDeviceDefinition (p. 26)
- CreateDeviceDefinitionVersion (p. 30)
- CreateFunctionDefinition (p. 34)
- CreateFunctionDefinitionVersion (p. 41)
- CreateGroup (p. 47)
- CreateGroupCertificateAuthority (p. 51)
- CreateGroupVersion (p. 54)
- CreateLoggerDefinition (p. 57)
- CreateLoggerDefinitionVersion (p. 62)
- CreateResourceDefinition (p. 65)
- CreateResourceDefinitionVersion (p. 73)
- CreateSoftwareUpdateJob (p. 80)
- CreateSubscriptionDefinition (p. 84)
- CreateSubscriptionDefinitionVersion (p. 88)
- DeleteConnectorDefinition (p. 92)
- DeleteCoreDefinition (p. 93)
- DeleteDeviceDefinition (p. 95)
• DeleteFunctionDefinition (p. 97)
• DeleteGroup (p. 99)
• DeleteLoggerDefinition (p. 100)
• DeleteResourceDefinition (p. 102)
• DeleteSubscriptionDefinition (p. 104)
• DisassociateRoleFromGroup (p. 105)
• DisassociateServiceRoleFromAccount (p. 108)
• GetAssociatedRole (p. 110)
• GetBulkDeploymentStatus (p. 112)
• GetConnectivityInfo (p. 115)
• GetConnectorDefinition (p. 119)
• GetConnectorDefinitionVersion (p. 121)
• GetCoreDefinition (p. 125)
• GetCoreDefinitionVersion (p. 127)
• GetDeploymentStatus (p. 131)
• GetDeviceDefinition (p. 133)
• GetDeviceDefinitionVersion (p. 136)
• GetFunctionDefinition (p. 139)
• GetFunctionDefinitionVersion (p. 142)
• GetGroup (p. 148)
• GetGroupCertificateAuthority (p. 151)
• GetGroupCertificateConfiguration (p. 154)
• GetGroupVersion (p. 156)
• GetLoggerDefinition (p. 160)
• GetLoggerDefinitionVersion (p. 162)
• GetResourceDefinition (p. 166)
• GetResourceDefinitionVersion (p. 168)
• GetServiceRoleForAccount (p. 175)
• GetSubscriptionDefinition (p. 176)
• GetSubscriptionDefinitionVersion (p. 179)
• ListBulkDeploymentDetailedReports (p. 182)
• ListBulkDeployments (p. 186)
• ListConnectorDefinitionVersions (p. 188)
• ListConnectorDefinitions (p. 191)
• ListCoreDefinitionVersions (p. 193)
• ListCoreDefinitions (p. 196)
• ListDeployments (p. 198)
• ListDeviceDefinitionVersions (p. 201)
• ListDeviceDefinitions (p. 204)
• ListFunctionDefinitionVersions (p. 206)
• ListFunctionDefinitions (p. 209)
• ListGroupCertificateAuthorities (p. 211)
• ListGroupVersions (p. 214)
• ListGroups (p. 217)
• ListLoggerDefinitionVersions (p. 219)
AssociateRoleToGroup

Associates a role with a group. Your Greengrass core uses the role to access AWS services. The role's permissions should allow Greengrass Lambda functions and connectors to perform actions against the cloud.

URI: PUT /greengrass/groups/GroupId/role

CLI:

```bash
aws greengrass associate-role-to-group
   --group-id <value> \
   [--role-arn <value>] \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "GroupId": "string",
   "RoleArn": "string"
}
```

Parameters:

**GroupId (p. 571)**

The ID of the Greengrass group.
where used: path; required: true

type: string

**AssociateRoleToGroupRequestBody (p. 544)**

where used: body; required: true

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

schema:

AssociateRoleToGroupRequest

type: object

required: ["RoleArn"]

RoleArn

The ARN of the role to associate with this group.

type: string

**Responses:**

**200**

Success.

**AssociateRoleToGroupResponse (p. 638)**

```json
{
   "AssociatedAt": "string"
}
```

Group

type: object

AssociatedAt

The time, in milliseconds since the epoch, when the role ARN was associated with the group.

type: string

**400**

Invalid request.

**GeneralError (p. 614)**

```json
{
   "Message": "string",
   "ErrorDetails": [
      {
         "DetailedErrorCode": "string",
         "DetailedErrorMessage": "string"
      }
   ]
}
```
GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

500

Server error.

GeneralError (p. 614)

```json
{
"Message": "string",
"ErrorDetails": [
{
"DetailedErrorCode": "string",
"DetailedErrorMessage": "string"
}
]
}
```

GeneralError

General error information.

type: object

required: ["Message"]
Message
  A message that contains information about the error.
  type: string

ErrorDetails
  A list of error details.
  type: array
  items: ErrorDetail (p. 600)

ErrorDetail
  Details about the error.
  type: object

DetailedErrorCode
  A detailed error code.
  type: string

DetailedErrorMessage
  A detailed error message.
  type: string

**AssociateServiceRoleToAccount**

Associates a role with your account. AWS IoT Greengrass uses the role to access your Lambda functions and AWS IoT resources. A role is required for deployments to succeed. The role must have at least minimum permissions in the policy AWSGreengrassResourceAccessRolePolicy.

**URI:** PUT /greengrass/servicerole

**CLI:**

```bash
aws greengrass associate-service-role-to-account \  [--role-arn <value>] \  [--cli-input-json <value>] \  [--generate-cli-skeleton]
```

**cli-input-json format:**

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

**Parameters:**

**AssociateServiceRoleToAccountRequestBody** (p. 545)

where used: body; required: true

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

schema:
AssociateServiceRoleToAccountRequest
type: object
required: ["RoleArn"]
RoleArn
The ARN of the service role to associate with your account.
type: string

Responses:

200
Success.

AssociateServiceRoleToAccountResponse (p. 582)

{
  "AssociatedAt": "string"
}

AssociateServiceRoleToAccountResponse
type: object
AssociatedAt
The time when the service role was associated with the account.
type: string

400
Invalid request.

GeneralError (p. 614)

{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}

GeneralError
General error information.
type: object
required: ["Message"]
Message
    A message that contains information about the error.
    type: string

ErrorDetails
    A list of error details.
    type: array
    items: ErrorDetail (p. 600)

ErrorDetail
    Details about the error.
    type: object

DetailedErrorCode
    A detailed error code.
    type: string

DetailedErrorMessage
    A detailed error message.
    type: string

500
    Server error.

GeneralError (p. 614)

    
    "Message": "string",
    "ErrorDetails": [
    
    "DetailedErrorCode": "string",
    "DetailedErrorMessage": "string"
    ]

GeneralError
    General error information.
    type: object
    required: ["Message"]

Message
    A message that contains information about the error.
    type: string

ErrorDetails
    A list of error details.
CreateConnectorDefinition

Creates a connector definition. You can provide the initial version of the connector definition now or use CreateConnectorDefinitionVersion later.

URL: POST /greengrass/definition/connectors

Produces: application/json

CLI:

```
aws greengrass create-connector-definition \
    [--name <value>] \
    [--initial-version <value>] \
    [--tags <value>] \
    [--amzn-client-token <value>] \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

cli-input-json format:

```
{
    "Name": "string",
    "InitialVersion": {
        "Connectors": [ 
            {
                "Id": "string",
                "ConnectorArn": "string",
                "Parameters": {
                    "additionalProperty0": "string",
                    "additionalProperty1": "string",
                    "additionalProperty2": "string"
                }
            }
        ],
        "tags": {
            "additionalProperty0": "string",
            "additionalProperty1": "string",
        }
    }
}
```
"additionalProperty2": "string"
},
"AmznClientToken": "string"
}

Parameters:

CreateConnectorDefinitionRequestBody (p. 547)

where used: body; required: true

{
  "Name": "string",
  "InitialVersion": {
    "Connectors": [
      {
        "Id": "string",
        "ConnectorArn": "string",
        "Parameters": {
          "additionalProperty0": "string",
          "additionalProperty1": "string",
          "additionalProperty2": "string"
        }
      }
    ],
    "tags": {
      "additionalProperty0": "string",
      "additionalProperty1": "string",
      "additionalProperty2": "string"
    }
  }
}

schema:
Name
The name of the connector definition.
type: string

InitialVersion
Information about the connector definition version, which is a container for connectors.
type: object

Connectors
A list of references to connectors in this version, with their corresponding configuration settings.
type: array

items: Connector (p. 588)

Connector
Information about a connector. Connectors run on the Greengrass core and contain built-in integration with local infrastructure, device protocols, AWS, and other cloud services.
type: object

required: ["Id", "ConnectorArn"]
Id

A descriptive or arbitrary ID for the connector. This value must be unique within the connector
definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-]+.

type: string

ConnectorArn

The ARN of the connector.

type: string

Parameters

The parameters or configuration used by the connector.

type: object

additionalProperties: An object with properties of type string that represent the connector
configuration.

tags

The resource tags.

type: object

additionalProperties: The key-value pair for the resource tag. Type: string

X-Amzn-Client-Token (p. 579)

A client token used to enforce the idempotency of this API.

where used: header; required: false

type: string

Responses:

200 (CreateConnectorDefinitionResponse)

DefinitionInformation (p. 594)

```json
{
    "Name": "string",
    "Id": "string",
    "Arn": "string",
    "LastUpdatedTimestamp": "string",
    "CreationTimestamp": "string",
    "LatestVersion": "string",
    "LatestVersionArn": "string"
}
```

DefinitionInformation

Information about a definition.

type: object

Name

The name of the definition.
type: string
Id
The ID of the definition.

type: string
Arn
The ARN of the definition.

type: string
LastUpdatedTimestamp
The time, in milliseconds since the epoch, when the definition was last updated.

type: string
CreationTimestamp
The time, in milliseconds since the epoch, when the definition was created.

type: string
LatestVersion
The ID of the latest version associated with the definition.

type: string
LatestVersionArn
The ARN of the latest version associated with the definition.

type: string

400
Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError
General error information.

type: object
required: ["Message"]

Message
A message that contains information about the error.

type: string
ErrorDetails
   A list of error details.
   type: array
   items: ErrorDetail (p. 600)

ErrorDetail
   Details about the error.
   type: object

DetailedErrorCode
   A detailed error code.
   type: string

DetailedErrorMessage
   A detailed error message.
   type: string

CreateConnectorDefinitionVersion

Creates a version of a connector definition that has already been defined.

URI: POST /greengrass/definition/connectors/ConnectorDefinitionId/versions

Produces: application/json

CLI:

```
aws greengrass create-connector-definition-version \  
   --connector-definition-id <value> \  
   [--connectors <value>] \  
   [--amzn-client-token <value>] \  
   [--cli-input-json <value>] \  
   [--generate-cli-skeleton]
```

cli-input-json format:

```
{
   "ConnectorDefinitionId": "string",
   "Connectors": [
   {
      "Id": "string",
      "ConnectorArn": "string",
      "Parameters": {
         "additionalProperty0": "string",
         "additionalProperty1": "string",
         "additionalProperty2": "string"
      }
   },
   
   "AmznClientToken": "string"
```
Parameters:

**ConnectorDefinitionId** (p. 546)

The ID of the connector definition.

where used: path; required: true
type: string

**CreateConnectorDefinitionVersionRequestBody** (p. 548)

where used: body; required: true

```json
{
  "Connectors": [
    {
      "Id": "string",
      "ConnectorArn": "string",
      "Parameters": {
        "additionalProperty0": "string",
        "additionalProperty1": "string",
        "additionalProperty2": "string"
      }
    }
  ]
}
```

schema:
ConnectorDefinitionVersion

Information about the connector definition version, which is a container for connectors.
type: object

Connectors

A list of references to connectors in this version, with their corresponding configuration settings.
type: array
items: Connector (p. 588)

Connector

Information about a connector. Connectors run on the Greengrass core and contain built-in integration with local infrastructure, device protocols, AWS, and other cloud services.
type: object
required: ["Id", "ConnectorArn"]

Id

A descriptive or arbitrary ID for the connector. This value must be unique within the connector definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9_\-]+.
type: string

ConnectorArn

The ARN of the connector.
type: string
Parameters

The parameters or configuration used by the connector.

type: object

additionalProperties: An object with properties of type string that represent the connector configuration.

X-Amzn-Client-Token (p. 579)

A client token used to enforce the idempotency of this API.

where used: header; required: false

type: string

Responses:

200 (CreateConnectorDefinitionVersionResponse)

VersionInformation (p. 676)

```json
{
  "Arn": "string",
  "Id": "string",
  "Version": "string",
  "CreationTimestamp": "string"
}
```

VersionInformation

Information about a version.

type: object

Arn

The ARN of the version.

type: string

Id

The ID of the parent definition that the version is associated with.

type: string

Version

The ID of the version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the version was created.

type: string

400

Invalid request.
CreateCoreDefinition

Creates a core definition. You can provide the initial version of the core definition now or use CreateCoreDefinitionVersion later. Greengrass groups must each contain exactly one Greengrass core.

URL: POST /greengrass/definition/cores

Produce: application/json
CLI:

```
aws greengrass create-core-definition \
[--name <value>] \ 
[--initial-version <value>] \ 
[--tags <value>] \ 
[--amzn-client-token <value>] \ 
[--cli-input-json <value>] \ 
[--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "Name": "string",
  "InitialVersion": {
    "Cores": [
      {
        "Id": "string",
        "ThingArn": "string",
        "CertificateArn": "string",
        "SyncShadow": "boolean"
      }
    ],
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  },
  "AmznClientToken": "string"
}
```

Parameters:

**CreateCoreDefinitionRequestBody (p. 549)**

Information required to create a core definition.

where used: body; required: true

```
{
  "Name": "string",
  "InitialVersion": {
    "Cores": [
      {
        "Id": "string",
        "ThingArn": "string",
        "CertificateArn": "string",
        "SyncShadow": true
      }
    ],
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  }
}
```

schema:
Name

The name of the core definition.

type: string

InitialVersion

Information about a core definition version.

type: object

Cores

A list of cores in the core definition version.

type: array

items: Core (p. 590)

Core

Information about a core.

type: object

required: ["Id", "ThingArn", "CertificateArn"]

Id

A descriptive or arbitrary ID for the core. This value must be unique within the core definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9_: ]+

type: string

ThingArn

The ARN of the thing that is the core.

type: string

CertificateArn

The ARN of the certificate associated with the core.

type: string

SyncShadow

If true, the core's local shadow is synced with the cloud automatically.

type: boolean

tags

The resource tags.

type: object

additionalProperties: The key-value pair for the resource tag. Type: string

X-Amzn-Client-Token (p. 579)

A client token used to correlate requests and responses.

where used: header; required: false
type: string

Responses:

200 (CreateCoreDefinitionResponse)

DefinitionInformation (p. 594)

```
{
    "Name": "string",
    "Id": "string",
    "Arn": "string",
    "LastUpdatedTimestamp": "string",
    "CreationTimestamp": "string",
    "LatestVersion": "string",
    "LatestVersionArn": "string"
}
```

DefinitionInformation

Information about a definition.

Name

The name of the definition.

type: string

Id

The ID of the definition.

type: string

Arn

The ARN of the definition.

type: string

LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.

type: string

LatestVersion

The ID of the latest version associated with the definition.

type: string

LatestVersionArn

The ARN of the latest version associated with the definition.

type: string
400

Invalid request.

GeneralError (p. 614)

```json
{
   "Message": "string",
   "ErrorDetails": [
      {
         "DetailedErrorCode": "string",
         "DetailedErrorMessage": "string"
      }
   ]
}
```

GeneralError

General error information.

- **type**: object
- **required**: ["Message"]

**Message**

A message that contains information about the error.

- **type**: string

**ErrorDetails**

A list of error details.

- **type**: array
- **items**: ErrorDetail (p. 600)

**ErrorDetail**

Details about the error.

- **type**: object

**DetailedErrorCode**

A detailed error code.

- **type**: string

**DetailedErrorMessage**

A detailed error message.

- **type**: string

---

**CreateCoreDefinitionVersion**

Creates a version of a core definition that has already been defined. Greengrass groups must each contain exactly one Greengrass core.

**URI:** POST /greengrass/definition/cores/CoreDefinitionId/versions

**Produces:** application/json
CLI:

```bash
aws greengrass create-core-definition-version
--core-definition-id <value> \
[--cores <value>] \
[--amzn-client-token <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "CoreDefinitionId": "string",
  "Cores": [
    {
      "Id": "string",
      "ThingArn": "string",
      "CertificateArn": "string",
      "SyncShadow": "boolean"
    }
  ],
  "AmznClientToken": "string"
}
```

Parameters:

CoreDefinitionId (p. 547)

The ID of the core definition.

where used: path; required: true
type: string

CreateCoreDefinitionVersionRequestBody (p. 550)

where used: body; required: true

```json
{
  "Cores": [
    {
      "Id": "string",
      "ThingArn": "string",
      "CertificateArn": "string",
      "SyncShadow": true
    }
  ]
}
```

schema:
CoreDefinitionVersion

Information about a core definition version.
type: object

Cores

A list of cores in the core definition version.
type: array
Core

Information about a core.

type: object

required: ["Id", "ThingArn", "CertificateArn"]

Id

A descriptive or arbitrary ID for the core. This value must be unique within the core definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9_:.-]+.

type: string

ThingArn

The ARN of the thing that is the core.

type: string

CertificateArn

The ARN of the certificate associated with the core.

type: string

SyncShadow

If true, the core's local shadow is synced with the cloud automatically.

type: boolean

X-Amzn-Client-Token (p. 579)

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200 (CreateCoreDefinitionVersionResponse)

VersionInformation (p. 676)

```
{
"Arn": "string",
"Id": "string",
"Version": "string",
"CreationTimestamp": "string"
}
```

VersionInformation

Information about a version.

type: object

Arn

The ARN of the version.
CreateCoreDefinitionVersion

type: string
Id
The ID of the parent definition that the version is associated with.
type: string
Version
The ID of the version.
type: string
CreationTimestamp
The time, in milliseconds since the epoch, when the version was created.
type: string

400
Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError
General error information.
type: object
required: ["Message"]
Message
A message that contains information about the error.
type: string
ErrorDetails
A list of error details.
type: array
items: ErrorDetail (p. 600)

ErrorDetail
Details about the error.
type: object
DetailedErrorCode
A detailed error code.
CreateDeployment

Creates a deployment. CreateDeployment requests are idempotent with respect to the X-Amzn-Client-Token token and the request parameters.

URI: POST /greengrass/groups/GroupId/deployments

Produces: application/json

CLI:

```
aws greengrass create-deployment
  --group-id <value>
  [--deployment-type <value>]
  [--deployment-id <value>]
  [--group-version-id <value>]
  [--amzn-client-token <value>]
  [--cli-input-json <value>]
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "GroupId": "string",
  "DeploymentType": "NewDeployment|Redeployment|ResetDeployment|ForceResetDeployment",
  "DeploymentId": "string",
  "GroupVersionId": "string",
  "AmznClientToken": "string"
}
```

Parameters:

GroupId (p. 571)

The ID of the Greengrass group.

where used: path; required: true

type: string

CreateDeploymentRequestBody (p. 551)

where used: body; required: true

```
{
  "DeploymentType": "NewDeployment|Redeployment|ResetDeployment|ForceResetDeployment",
  "DeploymentId": "string",
  "GroupVersionId": "string"
}
```

schema:
CreateDeploymentRequest

Information about a deployment.

type: object

required: ["DeploymentType"]

DeploymentType

The type of deployment. When used for CreateDeployment, only NewDeployment and Redeployment are valid.

type: string

enum: ["NewDeployment", "Redeployment", "ResetDeployment", "ForceResetDeployment"]

DeploymentId

The ID of the previous deployment you want to redeploy.

type: string

GroupVersionId

The ID of the group version to be deployed.

type: string

X-Amzn-Client-Token (p. 579)

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200

Success. The group was deployed.

CreateDeploymentResponse (p. 592)

```json
{
  "DeploymentArn": "string",
  "DeploymentId": "string"
}
```

CreateDeploymentResponse

type: object

DeploymentArn

The ARN of the deployment.

type: string

DeploymentId

The ID of the deployment.
CreateDeviceDefinition

Creates a device definition. You can provide the initial version of the device definition now or use CreateDeviceDefinitionVersion later.

URI: POST /greengrass/definition/devices
CreateDeviceDefinition

Parameters:

CreateDeviceDefinitionRequestBody (p. 551)

Information required to create a device definition.

where used: body; required: true

```json
{
  "Name": "string",
  "InitialVersion": {
    "Devices": [
      {
        "Id": "string",
        "ThingArn": "string",
        "CertificateArn": "string",
        "SyncShadow": true
      }
    ],
    "tags": {
      "additionalProperty0": "string",
      "additionalProperty1": "string",
      "additionalProperty2": "string"
    },
    "AmznClientToken": "string"
  }
}
```
schema:
Name

The name of the device definition.
type: string
InitialVersion

Information about a device definition version.
type: object
Devices

A list of devices in the definition version.
type: array
items: Device (p. 597)
Device

Information about a device.
type: object
required: ["Id", "ThingArn", "CertificateArn"]
Id

A descriptive or arbitrary ID for the device. This value must be unique within the device
definition version. Maximum length is 128 characters with the pattern [ a-zA-Z0-9_: ]+

type: string
ThingArn

The thing ARN of the device.
type: string
CertificateArn

The ARN of the certificate associated with the device.
type: string
SyncShadow

If true, the device's local shadow is synced with the cloud automatically.
type: boolean
tags

The resource tags.
type: object
additionalProperties: The key-value pair for the resource tag. Type: string

X-Amzn-Client-Token (p. 579)

A client token used to correlate requests and responses.
where used: header; required: false
Responses:

200 (CreateDeviceDefinitionResponse)

DefinitionInformation (p. 594)

```json
{
    "Name": "string",
    "Id": "string",
    "Arn": "string",
    "LastUpdatedTimestamp": "string",
    "CreationTimestamp": "string",
    "LatestVersion": "string",
    "LatestVersionArn": "string"
}
```

DefinitionInformation

Information about a definition.

type: object

Name

The name of the definition.

type: string

Id

The ID of the definition.

type: string

Arn

The ARN of the definition.

type: string

LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.

type: string

LatestVersion

The ID of the latest version associated with the definition.

type: string

LatestVersionArn

The ARN of the latest version associated with the definition.

type: string
400

Invalid request.

GeneralError (p. 614)

```
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```
CLI:

```bash
aws greengrass create-device-definition-version
   --device-definition-id <value>
   [--devices <value>]
   [--amzn-client-token <value>]
   [--cli-input-json <value>]
   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "DeviceDefinitionId": "string",
   "Devices": [
       {
           "Id": "string",
           "ThingArn": "string",
           "CertificateArn": "string",
           "SyncShadow": true
       }
   ],
   "AmznClientToken": "string"
}
```

Parameters:

**DeviceDefinitionId** (p. 570)

The ID of the device definition.

where used: path; required: true

type: string

**CreateDeviceDefinitionVersionRequestBody** (p. 552)

where used: body; required: true

```json
{
   "Devices": [
       {
           "Id": "string",
           "ThingArn": "string",
           "CertificateArn": "string",
           "SyncShadow": true
       }
   ]
}
```

schema:

**DeviceDefinitionVersion**

Information about a device definition version.

type: object

**Devices**

A list of devices in the definition version.

type: array
items: **Device** *(p. 597)*

**Device**

Information about a device.

type: object

required: ["Id", "ThingArn", "CertificateArn"]

**Id**

A descriptive or arbitrary ID for the device. This value must be unique within the device definition version. Maximum length is 128 characters with the pattern `[a-zA-Z0-9_-]*`.  

type: string  

**ThingArn**

The thing ARN of the device.

type: string

**CertificateArn**

The ARN of the certificate associated with the device.

type: string

**SyncShadow**

If true, the device's local shadow is synced with the cloud automatically.

type: boolean

**X-Amzn-Client-Token** *(p. 579)*

A client token used to correlate requests and responses.  

where used: header; required: false

type: string

**Responses:**

**200** *(CreateDeviceDefinitionVersionResponse)*

**VersionInformation** *(p. 676)*

```json
{  "Arn": "string",  "Id": "string",  "Version": "string",  "CreationTimestamp": "string" }
```

**VersionInformation**

Information about a version.

type: object

**Arn**

The ARN of the version.
type: string
Id
The ID of the parent definition that the version is associated with.
type: string
Version
The ID of the version.
type: string
CreationTimestamp
The time, in milliseconds since the epoch, when the version was created.
type: string

400
Invalid request.

GeneralError (p. 614)

{
    "Message": "string",
    "ErrorDetails": [
    {
        "DetailedErrorCode": "string",
        "DetailedErrorMessage": "string"
    }
    ]
}

GeneralError
General error information.
type: object
required: ["Message"]
Message
A message that contains information about the error.
type: string
ErrorDetails
A list of error details.
type: array
items: ErrorDetail (p. 600)
ErrorDetail
Details about the error.
type: object
DetailedErrorCode
A detailed error code.
CreateFunctionDefinition

Creates a function definition that contains a list of Lambda functions and their configurations to use in a group. You can create an initial version of the definition now by providing a list of Lambda functions and their configurations, or later by using CreateFunctionDefinitionVersion.

**URI:** POST /greengrass/definition/functions

**Produces:** application/json

**CLI:**

```
aws greengrass create-function-definition
    [--name <value>] 
    [--initial-version <value>] 
    [--tags <value>] 
    [--amzn-client-token <value>]  
    [--cli-input-json <value>] 
    [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "Name": "string",
   "InitialVersion": {
      "DefaultConfig": {
         "Execution": {
            "IsolationMode": "GreengrassContainer|NoContainer",
            "RunAs": {
               "Uid": "integer",
               "Gid": "integer"
            }
         }
      },
      "Functions": [ 
         {
            "Id": "string",
            "FunctionArn": "string",
            "FunctionConfiguration": {
               "Pinned": "boolean",
               "Executable": "string",
               "ExecArgs": "string",
               "MemorySize": "integer",
               "Timeout": "integer",
               "EncodingType": "binary|json",
               "Environment": {
                  "Variables": {
                     "additionalProperty0": "string",
                     "additionalProperty1": "string",
                     "additionalProperty2": "string"
                  },
                  "ResourceAccessPolicies": [  
```
CreateFunctionDefinition

"ResourceId": "string",
"Permission": "ro|rw"
],
"AccessSysfs": "boolean",
"Execution": {
  "IsolationMode": "GreengrassContainer|NoContainer",
  "RunAs": {
    "Uid": "integer",
    "Gid": "integer"
  }
}
}
}

tag": {
  "additionalProperty0": "string",
  "additionalProperty1": "string",
  "additionalProperty2": "string"
},
"AmznClientToken": "string"

Parameters:

CreateFunctionDefinitionRequestBody (p. 553)

where used: body; required: true

{
  "Name": "string",
  "InitialVersion": {
    "DefaultConfig": {
      "Execution": {
        "IsolationMode": "GreengrassContainer|NoContainer",
        "RunAs": {
          "Uid": 1001,
          "Gid": 1002
        }
      }
    },
    "Functions": [
      {
        "Id": "string",
        "FunctionArn": "string",
        "FunctionConfiguration": {
          "Pinned": true,
          "Executable": "string",
          "ExecArgs": "string",
          "MemorySize": 0,
          "Timeout": 0,
          "EncodingType": "binary|json",
          "Environment": {
            "Variables": {
              "additionalProperty0": "string",
              "additionalProperty1": "string",
              "additionalProperty2": "string"
            },
            "ResourceAccessPolicies": [
              {
                "ResourceId": "string",
                "Permission": "ro|rw"
              }
            ]
          }
        }
      }
    ]
  }
}
schema:

Name
The name of the function definition.

type: string

InitialVersion
Information about a function definition version.

type: object

DefaultConfig
The default configuration that applies to all Lambda functions in the group. Individual Lambda functions can override these settings.

type: object

Execution
Configuration information that specifies how a Lambda function runs.

Functions
A list of Lambda functions in this function definition version.

type: array

items: Function (p. 601)

Information about a Lambda function.

type: object

required: ["Id"]

Id
A descriptive or arbitrary ID for the function. This value must be unique within the function definition version. Maximum length is 128 characters with the pattern `[a-zA-Z0-9_\-]+`.

type: string
FunctionArn

The ARN of the alias (recommended) or version of the target Lambda function.

type: string

FunctionConfiguration

The configuration of the Lambda function.

type: object

Pinned

True if the function is pinned. Pinned means the function is long-lived and starts when the core starts.

type: boolean

Executable

The name of the function executable.

type: string

ExecArgs

The execution arguments.

type: string

MemorySize

The memory size, in KB, required by the function. This setting does not apply and should be cleared when you run the Lambda function without containerization.

type: integer

Timeout

The allowed function execution time, after which Lambda should terminate the function. This timeout still applies to pinned Lambda functions for each request.

type: integer

EncodingType

The expected encoding type of the input payload for the function. The default is json.

type: string

enum: ["binary", "json"]

Environment

The environment configuration of the function.

type: object

Variables

Environment variables for the Lambda function's configuration.

type: object

additionalProperties: An object with properties of type string that represent the environment variables.
ResourceAccessPolicies

A list of the resources, with their permissions, to which the Lambda function is granted access. A Lambda function can have at most 10 resources. ResourceAccessPolicies applies only when you run the Lambda function in a Greengrass container.

type: array

items: ResourceAccessPolicy (p. 660)

ResourceAccessPolicy

A policy used by the function to access a resource.

type: object

required: ["ResourceId"]

ResourceId

The ID of the resource. (This ID is assigned to the resource when you create the resource definition.)

type: string

Permission

The type of permission a function has to access a resource.

type: string

enum: ["ro", "rw"]

AccessSysfs

If true, the Lambda function is allowed to access the host's /sys folder. Use this when the Lambda function needs to read device information from /sys. This setting applies only when you run the Lambda function in a Greengrass container.

type: boolean

Execution

Configuration information that specifies how a Lambda function runs.

type: object

IsolationMode

Specifies whether the Lambda function runs in a Greengrass container (default) or without containerization. Unless your scenario requires that you run without containerization, we recommend that you run in a Greengrass container. Omit this value to run the Lambda function with the default containerization for the group.

type: string

enum: ["GreengrassContainer", "NoContainer"]

RunAs

Specifies the user and group whose permissions are used when running the Lambda function. You can specify one or both values to override the default values. To minimize the risk of unintended changes or malicious attacks, we recommend that you avoid running as root unless absolutely necessary. To run as root, you must update config.json in greengrass-root/config to set allowFunctionsToRunAsRoot to yes.

type: object
**Uid**

The user ID whose permissions are used to run a Lambda function.

type: integer

**Gid**

The group ID whose permissions are used to run a Lambda function.

type: integer

**tags**

The resource tags.

type: object

additionalProperties: The key-value pair for the resource tag. Type: string

**X-Amzn-Client-Token (p. 579)**

A client token used to correlate requests and responses.

where used: header; required: false

type: string

**Responses:**

**200** (CreateFunctionDefinitionResponse)

**DefinitionInformation (p. 594)**

```json
{
    "Name": "string",
    "Id": "string",
    "Arn": "string",
    "LastUpdatedTimestamp": "string",
    "CreationTimestamp": "string",
    "LatestVersion": "string",
    "LatestVersionArn": "string"
}
```

**DefinitionInformation**

Information about a definition.

type: object

**Name**

The name of the definition.

type: string

**Id**

The ID of the definition.

type: string

**Arn**

The ARN of the definition.
type: string

LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.

type: string

LatestVersion

The ID of the latest version associated with the definition.

type: string

LatestVersionArn

The ARN of the latest version associated with the definition.

type: string

400

Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.
CreateFunctionDefinitionVersion

Creates a version of a Lambda function definition that has already been defined.

URL: POST /greengrass/definition/functions/FunctionDefinitionId/versions

Produces: application/json

CLI:

```
aws greengrass create-function-definition-version \
    --function-definition-id <value> \ 
    [--default-config <value>] \ 
    [--functions <value>] \ 
    [--amzn-client-token <value>] \ 
    [--cli-input-json <value>] \ 
    [--generate-cli-skeleton]
```

cli-input-json format:

```
{
    "FunctionDefinitionId": "string",
    "DefaultConfig": {
        "Execution": {
            "IsolationMode": "GreengrassContainer|NoContainer",
            "RunAs": {
                "Uid": "integer",
                "Gid": "integer"
            }
        }
    },
    "Functions": [
        {
            "Id": "string",
            "FunctionArn": "string",
            "FunctionConfiguration": {
                "Pinned": "boolean",
                "Executable": "string",
                "ExecArgs": "string",
                "MemorySize": "integer",
                "Timeout": "integer",
                "EncodingType": "binary|json",
                "Environment": {
                    "Variables": {
                        "additionalProperty0": "string",
                        "additionalProperty1": "string",
                        "additionalProperty2": "string"
                    }
                }
            }
        }
    ]
}
```
Parameters:

**FunctionDefinitionId** *(p. 571)*

The ID of the function definition.

where used: path; required: true

type: string

**CreateFunctionDefinitionVersionRequestBody** *(p. 554)*

Information about the function definition version.

where used: body; required: true

```json
{
  "DefaultConfig": {
    "Execution": {
      "IsolationMode": "GreengrassContainer|NoContainer",
      "RunAs": {
        "Uid": 1001,
        "Gid": 1002
      }
    }
  },
  "Functions": [
    {
      "Id": "string",
      "FunctionArn": "string",
      "FunctionConfiguration": {
        "Pinned": true,
        "Executable": "string",
        "ExecArgs": "string",
        "MemorySize": 0,
        "Timeout": 0,
        "EncodingType": "binary|json",
        "Environment": {
          "Variables": {
            "additionalProperty0": "string",
            "additionalProperty1": "string",
            "additionalProperty2": "string"
          }
        }
      }
    }
  ]
}
```
CreateFunctionDefinitionVersion

```
{
  "ResourceId": "string",
  "Permission": "ro|rw"
}
"AccessSysfs": true,
"Execution": {
  "IsolationMode": "GreengrassContainer|NoContainer",
  "RunAs": {
    "Uid": 1001,
    "Gid": 1002
  }
}
}
}
```
FunctionConfiguration

The configuration of the Lambda function.

type: object

Pinned

True if the function is pinned. Pinned means the function is long-lived and starts when the core starts.

type: boolean

Executable

The name of the function executable.

type: string

ExecArgs

The execution arguments.

type: string

MemorySize

The memory size, in KB, required by the function. This setting does not apply and should be cleared when you run the Lambda function without containerization.

type: integer

Timeout

The allowed function execution time, after which Lambda should terminate the function. This timeout still applies to pinned Lambda functions for each request.

type: integer

EncodingType

The expected encoding type of the input payload for the function. The default is json.

type: string

enum: ["binary", "json"]

Environment

The environment configuration of the function.

type: object

Variables

Environment variables for the Lambda function's configuration.

type: object

additionalProperties: An object with properties of type string that represent the environment variables.

ResourceAccessPolicies

A list of the resources, with their permissions, to which the Lambda function is granted access. A Lambda function can have at most 10 resources. ResourceAccessPolicies applies only when you run the Lambda function in a Greengrass container.
type: array

items: ResourceAccessPolicy (p. 660)

ResourceAccessPolicy

A policy used by the function to access a resource.

type: object

required: ["ResourceId"]

ResourceId

The ID of the resource. (This ID is assigned to the resource when you create the resource definition.)

type: string

Permission

The type of permission a function has to access a resource.

type: string

enum: ["ro", "rw"]

AccessSysfs

If true, the Lambda function is allowed to access the host's /sys folder. Use this when the Lambda function needs to read device information from /sys. This setting applies only when you run the Lambda function in a Greengrass container.

type: boolean

Execution

Configuration information that specifies how a Lambda function runs.

type: object

IsolationMode

Specifies whether the Lambda function runs in a Greengrass container (default) or without containerization. Unless your scenario requires that you run without containerization, we recommend that you run in a Greengrass container. Omit this value to run the Lambda function with the default containerization for the group.

type: string

enum: ["GreengrassContainer", "NoContainer"]

RunAs

Specifies the user and group whose permissions are used when running the Lambda function. You can specify one or both values to override the default values. To minimize the risk of unintended changes or malicious attacks, we recommend that you avoid running as root unless absolutely necessary. To run as root, you must update config.json in greengrass-root/config to set allowFunctionsToRunAsRoot to yes.

type: object

Uid

The user ID whose permissions are used to run a Lambda function.

type: integer
Gid

The group ID whose permissions are used to run a Lambda function.

type: integer

**X-Amzn-Client-Token** *(p. 579)*

A client token used to correlate requests and responses.

where used: header; required: false

type: string

**Responses:**

**200** *(CreateFunctionDefinitionVersionResponse)*

**VersionInformation** *(p. 676)*

```json
{
    "Arn": "string",
    "Id": "string",
    "Version": "string",
    "CreationTimestamp": "string"
}
```

**VersionInformation**

Information about a version.

type: object

**Arn**

The ARN of the version.

type: string

**Id**

The ID of the parent definition that the version is associated with.

type: string

**Version**

The ID of the version.

type: string

**CreationTimestamp**

The time, in milliseconds since the epoch, when the version was created.

type: string

**400**

Invalid request.

**GeneralError** *(p. 614)*

```json
{
}
```
"Message": "string",
"ErrorDetails": [
  {
    "DetailedErrorCode": "string",
    "DetailedErrorMessage": "string"
  }
]
}

**GeneralError**

General error information.

- **type**: object
- required: ["Message"]

**Message**

A message that contains information about the error.

- **type**: string

**ErrorDetails**

A list of error details.

- **type**: array
- **items**: ErrorDetail (p. 600)

**ErrorDetail**

Details about the error.

- **type**: object

**DetailedErrorCode**

A detailed error code.

- **type**: string

**DetailedErrorMessage**

A detailed error message.

- **type**: string

---

**CreateGroup**

Creates a group. You can provide the initial version of the group or use CreateGroupVersion at a later time. Tip: You can use the gg_group_setup package (https://github.com/awslabs/aws-greengrass-group-setup) as a library or command-line application to create and deploy Greengrass groups.

**URI**: `POST /greengrass/groups`

**Produces**: application/json

**CLI**:

```
aws greengrass create-group \
    [--name <value>] \
```
CreateGroup

Parameters:

CreateGroupRequestBody (p. 558)

where used: body; required: true

```
{
  "Name": "string",
  "InitialVersion": {
    "CoreDefinitionVersionArn": "string",
    "DeviceDefinitionVersionArn": "string",
    "FunctionDefinitionVersionArn": "string",
    "SubscriptionDefinitionVersionArn": "string",
    "LoggerDefinitionVersionArn": "string",
    "ResourceDefinitionVersionArn": "string",
    "ConnectorDefinitionVersionArn": "string"
  },
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  }
}
```

schema:

Name

The name of the group.

type: string

InitialVersion

Information about a group version.
CreateGroup

- type: object
  CoreDefinitionVersionArn
    The ARN of the core definition version for this group.
    type: string
  DeviceDefinitionVersionArn
    The ARN of the device definition version for this group.
    type: string
  FunctionDefinitionVersionArn
    The ARN of the function definition version for this group.
    type: string
  SubscriptionDefinitionVersionArn
    The ARN of the subscription definition version for this group.
    type: string
  LoggerDefinitionVersionArn
    The ARN of the logger definition version for this group.
    type: string
  ResourceDefinitionVersionArn
    The ARN of the resource definition version for this group.
    type: string
  ConnectorDefinitionVersionArn
    The ARN of the connector definition version for this group.
    type: string
  tags
    The resource tags.
    type: object
    additionalProperties: The key-value pair for the resource tag. Type: string

X-Amzn-Client-Token (p. 579)

A client token used to correlate requests and responses.
where used: header; required: false
  type: string

Responses:

200 (CreateGroupResponse)

Success. The group was created.

DefinitionInformation (p. 594)
CreateGroup

```json
{
  "Name": "string",
  "Id": "string",
  "Arn": "string",
  "LastUpdatedTimestamp": "string",
  "CreationTimestamp": "string",
  "LatestVersion": "string",
  "LatestVersionArn": "string"
}
```

**DefinitionInformation**

Information about a definition.

- **Name**: The name of the definition.
  - type: string

- **Id**: The ID of the definition.
  - type: string

- **Arn**: The ARN of the definition.
  - type: string

- **LastUpdatedTimestamp**: The time, in milliseconds since the epoch, when the definition was last updated.
  - type: string

- **CreationTimestamp**: The time, in milliseconds since the epoch, when the definition was created.
  - type: string

- **LatestVersion**: The ID of the latest version associated with the definition.
  - type: string

- **LatestVersionArn**: The ARN of the latest version associated with the definition.
  - type: string

**400**

Invalid request.

**GeneralError (p. 614)**

```json
{
  "Message": "string",
  "ErrorDetails": []
}
```
CreateGroupCertificateAuthority

Creates a CA for the group. If a CA already exists, it rotates the existing CA.

URI: POST /greengrass/groups/GroupId/certificateauthorities

Produces: application/json

CLI:

```bash
aws greengrass create-group-certificate-authority
  --group-id <value> \n  [--amzn-client-token <value>] \n  [--cli-input-json <value>] \n  [--generate-cli-skeleton]
```
### CreateGroupCertificateAuthority

**cli-input-json format:**

```json
{
  "GroupId": "string",
  "AmznClientToken": "string"
}
```

**Parameters:**

- **GroupId (p. 571)**
  - The ID of the Greengrass group.
  - where used: path; required: true
  - type: string

- **X-Amzn-Client-Token (p. 579)**
  - A client token used to correlate requests and responses.
  - where used: header; required: false
  - type: string

**Responses:**

- **200**
  - Success. The response body contains the new, active CA ARN.
  - CreateGroupCertificateAuthorityResponse (p. 593)

  ```json
  {
    "GroupCertificateAuthorityArn": "string"
  }
  ```

  **CreateGroupCertificateAuthorityResponse**
  - type: object
  - **GroupCertificateAuthorityArn**
    - The ARN of the group certificate authority.
    - type: string

- **400**
  - Invalid request.
  - GeneralError (p. 614)

  ```json
  {
    "Message": "string",
    "ErrorDetails": [
      {
        "DetailedErrorCode": "string",
        "DetailedErrorMessage": "string"
      }
    ]
  }
  ```
GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

500

Server error.

GeneralError (p. 614)
CreateGroupVersion

Creates a version of a group that has already been defined.

URI: POST /greengrass/groups/GroupID/versions

Produces: application/json

CLI:

```
aws greengrass create-group-version \
  --group-id <value> \
  [--core-definition-version-arn <value>] \
  [--device-definition-version-arn <value>] \
  [--function-definition-version-arn <value>] \
  [--subscription-definition-version-arn <value>] \
  [--logger-definition-version-arn <value>] \
  [--resource-definition-version-arn <value>] \
  [--connector-definition-version-arn <value>] \
  [--amzn-client-token <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "GroupId": "string",
  "CoreDefinitionVersionArn": "string",
  "DeviceDefinitionVersionArn": "string",
  "FunctionDefinitionVersionArn": "string",
  "..."
}
```
CreateGroupVersion

```
"SubscriptionDefinitionVersionArn": "string",
"LoggerDefinitionVersionArn": "string",
"ResourceDefinitionVersionArn": "string",
"ConnectorDefinitionVersionArn": "string",
"AmznClientToken": "string"
```

Parameters:

**GroupId (p. 571)**

The ID of the Greengrass group.

where used: path; required: true
type: string

**CreateGroupVersionRequestBody (p. 558)**

where used: body; required: true

```
{
  "CoreDefinitionVersionArn": "string",
  "DeviceDefinitionVersionArn": "string",
  "FunctionDefinitionVersionArn": "string",
  "SubscriptionDefinitionVersionArn": "string",
  "LoggerDefinitionVersionArn": "string",
  "ResourceDefinitionVersionArn": "string",
  "ConnectorDefinitionVersionArn": "string"
}
```

schema:

GroupVersion

Information about a group version.
type: object

CoreDefinitionVersionArn

The ARN of the core definition version for this group.
type: string

DeviceDefinitionVersionArn

The ARN of the device definition version for this group.
type: string

FunctionDefinitionVersionArn

The ARN of the function definition version for this group.
type: string

SubscriptionDefinitionVersionArn

The ARN of the subscription definition version for this group.
type: string

LoggerDefinitionVersionArn

The ARN of the logger definition version for this group.
type: string
ResourceDefinitionVersionArn

The ARN of the resource definition version for this group.

type: string
ConnectorDefinitionVersionArn

The ARN of the connector definition version for this group.

X-Amzn-Client-Token (p. 579)

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200 (CreateGroupVersionResponse)

Success. The response contains information about the group version.

VersionInformation (p. 676)

{
  "Arn": "string",
  "Id": "string",
  "Version": "string",
  "CreationTimestamp": "string"
}

VersionInformation

Information about a version.

type: object

Arn

The ARN of the version.

type: string

Id

The ID of the parent definition that the version is associated with.

type: string

Version

The ID of the version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the version was created.
400

Invalid request.

GeneralError (p. 614)

```
{
    "Message": "string",
    "ErrorDetails": [
    {
        "DetailedErrorCode": "string",
        "DetailedErrorMessage": "string"
    }
    ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

CreateLoggerDefinition

Creates a logger definition. You can provide the initial version of the logger definition now or use CreateLoggerDefinitionVersion later.
CreateLoggerDefinition

URI: POST /greengrass/definition/loggers

Produces: application/json

**CLI:**

```bash
aws greengrass create-logger-definition
    [--name <value>] \
    [--initial-version <value>] \
    [--tags <value>] \
    [--amzn-client-token <value>] \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

**cli-input-json format:**

```json
{
    "Name": "string",
    "InitialVersion": {
        "Loggers": [
            {
                "Id": "string",
                "Type": "FileSystem|AWSCloudWatch",
                "Component": "GreengrassSystem|Lambda",
                "Level": "DEBUG|INFO|WARN|ERROR|FATAL",
                "Space": "integer"
            }
        ],
        "tags": {
            "additionalProperty0": "string",
            "additionalProperty1": "string",
            "additionalProperty2": "string"
        },
        "AmznClientToken": "string"
    }
}
```

**Parameters:**

CreateLoggerDefinitionRequestBody (p. 559)

where used: body; required: true

```json
{
    "Name": "string",
    "InitialVersion": {
        "Loggers": [
            {
                "Id": "string",
                "Type": "FileSystem|AWSCloudWatch",
                "Component": "GreengrassSystem|Lambda",
                "Level": "DEBUG|INFO|WARN|ERROR|FATAL",
                "Space": 0
            }
        ],
        "tags": {
            "additionalProperty0": "string",
            "additionalProperty1": "string",
            "additionalProperty2": "string"
        }
    }
}
```
CreateLoggerDefinition

schema:

Name

The name of the logger definition.

type: string

InitialVersion

Information about a logger definition version.

type: object

Loggers

A list of loggers.

type: array

items: Logger (p. 652)

Logger

Information about a logger

type: object

required: ["Id", "Type", "Component", "Level"]

Id

A descriptive or arbitrary ID for the logger. This value must be unique within the logger definition version. Maximum length is 128 characters with the pattern \[a-zA-Z0-9:\_\-\+]\+.

type: string

Type

type: string

enum: ["FileSystem", "AWSCloudWatch"]

Component

type: string

enum: ["GreengrassSystem", "Lambda"]

Level

type: string

enum: ["DEBUG", "INFO", "WARN", "ERROR", "FATAL"]

Space

The amount of file space, in KB, to use if the local file system is used for logging purposes.

type: integer

tags

The resource tags.

type: object
**additionalProperties**: The key-value pair for the resource tag. Type: string

**X-Amzn-Client-Token** (p. 579)

A client token used to correlate requests and responses.

where used: header; required: false

type: string

**Responses:**

**200** *(CreateLoggerDefinitionResponse)*

**DefinitionInformation** (p. 594)

```json
{
    "Name": "string",
    "Id": "string",
    "Arn": "string",
    "LastUpdatedTimestamp": "string",
    "CreationTimestamp": "string",
    "LatestVersion": "string",
    "LatestVersionArn": "string"
}
```

**DefinitionInformation**

Information about a definition.

type: object

**Name**

The name of the definition.

type: string

**Id**

The ID of the definition.

type: string

**Arn**

The ARN of the definition.

type: string

**tags**

The resource tags.

type: object

**additionalProperties**: The key-value pair for the resource tag. Type: string

**LastUpdatedTimestamp**

The time, in milliseconds since the epoch, when the definition was last updated.

type: string
CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.

type: string

LatestVersion

The ID of the latest version associated with the definition.

type: string

LatestVersionArn

The ARN of the latest version associated with the definition.

type: string

400

Invalid request.

GeneralError (p. 614)

```
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string
DetailedErrorMessage

A detailed error message.

type: string

CreateLoggerDefinitionVersion

Creates a version of a logger definition that has already been defined.

URI: POST /greengrass/definition/loggers/LoggerDefinitionId/versions

Produces: application/json

CLI:

```bash
aws greengrass create-logger-definition-version \
--logger-definition-id <value> \
[--loggers <value>] \
[--amzn-client-token <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

cli-input-json format:

```json
{
    "LoggerDefinitionId": "string",
    "Loggers": [
        {
            "Id": "string",
            "Type": "FileSystem|AWSCloudWatch",
            "Component": "GreengrassSystem|Lambda",
            "Level": "DEBUG|INFO|WARN|ERROR|FATAL",
            "Space": "integer"
        }
    ],
    "AmznClientToken": "string"
}
```

Parameters:

**LoggerDefinitionId (p. 572)**

The ID of the logger definition.

where used: path; required: true

type: string

**CreateLoggerDefinitionVersionRequestBody (p. 560)**

where used: body; required: true

```json
{
    "Loggers": [
        {
            "Id": "string",
            "Type": "FileSystem|AWSCloudWatch",
            "Component": "GreengrassSystem|Lambda",
            "Level": "DEBUG|INFO|WARN|ERROR|FATAL",
            "Space": "integer"
        }
    ]
}
```
CreateLoggerDefinitionVersion

"Component": "GreengrassSystem|Lambda",
"Level": "DEBUG|INFO|WARN|ERROR|FATAL",
"Space": 0
}
]

schema:
LoggerDefinitionVersion

Information about a logger definition version.

type: object

Loggers
A list of loggers.

type: array

items: Logger (p. 652)

Logger
Information about a logger

type: object

required: ["Id", "Type", "Component", "Level"]

Id

A descriptive or arbitrary ID for the logger. This value must be unique within the logger definition version. Maximum length is 128 characters with the pattern \[a-zA-Z0-9_:\-]+\.

type: string

Type

type: string

enum: ["FileSystem", "AWSCloudWatch"]

Component

type: string

enum: ["GreengrassSystem", "Lambda"]

Level

type: string

enum: ["DEBUG", "INFO", "WARN", "ERROR", "FATAL"]

Space
The amount of file space, in KB, to use if the local file system is used for logging purposes.

type: integer

X-Amzn-Client-Token (p. 579)

A client token used to correlate requests and responses.

where used: header; required: false
type: string

Responses:

200 (CreateLoggerDefinitionVersionResponse)

VersionInformation (p. 676)

{
   "Arn": "string",
   "Id": "string",
   "Version": "string",
   "CreationTimestamp": "string"
}

VersionInformation

Information about a version.

type: object

Arn

The ARN of the version.

type: string

Id

The ID of the parent definition that the version is associated with.

type: string

Version

The ID of the version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the version was created.

type: string

400

Invalid request.

GeneralError (p. 614)

{
   "Message": "string",
   "ErrorDetails": [
   {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
   }
   ]
}

GeneralError

General error information.
CreateResourceDefinition

Creates a resource definition that contains a list of resources to be used in a group. You can create an initial version of the definition by providing a list of resources now, or use CreateResourceDefinitionVersion later.

URI: POST /greengrass/definition/resources

Produces: application/json

CLI:

```
aws greengrass create-resource-definition \
  [--name <value>] \ 
  [--initial-version <value>] \ 
  [--tags <value>] \ 
  [--amzn-client-token <value>] \ 
  [--cli-input-json <value>] \ 
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "Name": "string",
  "InitialVersion": {
    "Resources": [
```
Parameters:

CreateResourceDefinitionRequestBody (p. 561)

where used: body; required: true
AWS Greengrass API Reference
CreateResourceDefinition

```json
{
  "Id": "string",
  "Name": "string",
  "ResourceDataContainer": {
    "**-- NOTE:": "choose one of the following --**",
    "LocalDeviceResourceData": {
      "SourcePath": "string",
      "GroupOwnerSetting": {
        "AutoAddGroupOwner": true,
        "GroupOwner": "string"
      }
    },
    "LocalVolumeResourceData": {
      "SourcePath": "string",
      "DestinationPath": "string",
      "GroupOwnerSetting": {
        "AutoAddGroupOwner": true,
        "GroupOwner": "string"
      }
    },
    "SageMakerMachineLearningModelResourceData": {
      "SageMakerJobArn": "string",
      "DestinationPath": "string",
      "OwnerSetting": {
        "GroupOwner": "string",
        "GroupPermission": "ro|rw"
      }
    },
    "S3MachineLearningModelResourceData": {
      "S3Uri": "string",
      "DestinationPath": "string",
      "OwnerSetting": {
        "GroupOwner": "string",
        "GroupPermission": "ro|rw"
      }
    },
    "SecretsManagerSecretResourceData": {
      "ARN": "string",
      "AdditionalStagingLabelsToDownload": [
        "string"
      ]
    }
  }
},
"tags": {
  "additionalProperty0": "string",
  "additionalProperty1": "string",
  "additionalProperty2": "string"
}
}

schema:
Name

The name of the resource definition.

type: string

InitialVersion

Information about a resource definition version.

type: object
Resources

A list of resources.

type: array

items: Resource (p. 656)

Resource

Information about a resource.

type: object

required: ["Id", "Name", "ResourceDataContainer"]

Id

The resource ID, used to refer to a resource in the Lambda function configuration. Maximum length is 128 characters with the pattern \[a-zA-Z0-9:_-]+\]. This must be unique within a Greengrass group.

type: string

Name

The descriptive resource name, which is displayed on the AWS IoT Greengrass console. Maximum length is 128 characters with the pattern \[a-zA-Z0-9:_-]+\]. This must be unique within a Greengrass group.

type: string

ResourceDataContainer

A container for resource data. The container takes only one of the following supported resource data types:

- LocalDeviceResourceData
- LocalVolumeResourceData
- SageMakerMachineLearningModelResourceData
- S3MachineLearningModelResourceData
- SecretsManagerSecretResourceData.

type: object

NOTE: oneOf

LocalDeviceResourceData

Attributes that define a local device resource.

type: object

SourcePath

The local absolute path of the device resource. The source path for a device resource can refer only to a character device or block device under /dev.

type: string

GroupOwnerSetting

Group owner related settings for local resources.

type: object

AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.
CreateResourceDefinition

- **GroupOwner**
  - The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.
  - type: string

- **LocalVolumeResourceData**
  - Attributes that define a local volume resource.
  - type: object

  - **SourcePath**
    - The local absolute path of the volume resource on the host. The source path for a volume resource type cannot start with `/sys`.
    - type: string

  - **DestinationPath**
    - The absolute local path of the resource inside the Lambda environment.
    - type: string

- **GroupOwnerSetting**
  - Group owner related settings for local resources.
  - type: object

  - **AutoAddGroupOwner**
    - If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.
    - type: boolean

- **SageMakerMachineLearningModelResourceData**
  - Attributes that define an Amazon SageMaker machine learning resource.
  - type: object

  - **SageMakerJobArn**
    - The ARN of the Amazon SageMaker training job that represents the source model.
    - type: string

  - **DestinationPath**
    - The absolute local path of the resource inside the Lambda environment.
    - type: string
OwnerSetting

The owner setting for the downloaded machine learning resource.

type: object

required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are `rw` (read-write) or `ro` (read-only).

type: string

enum: ["ro", "rw"]

S3MachineLearningModelResourceData

Attributes that define an Amazon S3 machine learning resource.

type: object

S3Uri

The URI of the source model in an S3 bucket. The model package must be in tar.gz or .zip format.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.

type: object

required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are `rw` (read-write) or `ro` (read-only).

type: string

enum: ["ro", "rw"]
SecretsManagerSecretResourceData

Attributes that define a secret resource, which references a secret from AWS Secrets Manager. AWS IoT Greengrass stores a local, encrypted copy of the secret on the Greengrass core, where it can be securely accessed by connectors and Lambda functions.

type: object
AR

The ARN of the Secrets Manager secret to make available on the core. The value of the secret's latest version (represented by the _AWSCURRENT_ staging label) is included by default.

type: string
AdditionalStagingLabelsToDownload

Optional. The staging labels in addition to _AWSCURRENT_ whose values you want to make available on the core.

type: array
tags

The resource tags.

type: object

additionalProperties: The key-value pair for the resource tag. Type: string

_X-Amzn-Client-Token_ (p. 579)

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200 (CreateResourceDefinitionResponse)

DefinitionInformation (p. 594)

```json
{
  "Name": "string",
  "Id": "string",
  "Arn": "string",
  "LastUpdatedTimestamp": "string",
  "CreationTimestamp": "string",
  "LatestVersion": "string",
  "LatestVersionArn": "string"
}
```

DefinitionInformation

Information about a definition.

type: object
Name

The name of the definition.
CreateResourceDefinition

- `Id` (type: string)
  - The ID of the definition.

- `Arn` (type: string)
  - The ARN of the definition.

- `LastUpdatedTimestamp` (type: string)
  - The time, in milliseconds since the epoch, when the definition was last updated.

- `CreationTimestamp` (type: string)
  - The time, in milliseconds since the epoch, when the definition was created.

- `LatestVersion` (type: string)
  - The ID of the latest version associated with the definition.

- `LatestVersionArn` (type: string)
  - The ARN of the latest version associated with the definition.

400

Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError

- General error information.
  - type: object
    - required: ["Message"]

Message

- A message that contains information about the error.
  - type: string
ErrorDetails
A list of error details.
type: array
items: ErrorDetail (p. 600)

ErrorDetail
Details about the error.
type: object

DetailedErrorCode
A detailed error code.
type: string

DetailedErrorMessage
A detailed error message.
type: string

CreateResourceDefinitionVersion
Creates a version of a resource definition that has already been defined.

URL: POST /greengrass/definition/resources/<ResourceDefinitionId>/versions

Produces: application/json

CLI:

```
aws greengrass create-resource-definition-version
  --resource-definition-id <value> \
  [--resource <value>] \
  [--amzn-client-token <value>] \n  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "ResourceDefinitionId": "string",
  "Resources": [
    {
      "Id": "string",
      "Name": "string",
      "ResourceDataContainer": {
        "-- NOTE:": "choose one of the following --*",
        "LocalDeviceResourceData": {
          "SourcePath": "string",
          "GroupOwnerSetting": {
            "AutoAddGroupOwner": "boolean",
            "GroupOwner": "string"
          }
        },
        "LocalVolumeResourceData": {
          "SourcePath": "string",
        }
      }
    }
  ]
}
```
"DestinationPath": "string",
"GroupOwnerSetting": {
  "AutoAddGroupOwner": "boolean",
  "GroupOwner": "string"
}
},
"SageMakerMachineLearningModelResourceData": {
  "SageMakerJobArn": "string",
  "DestinationPath": "string",
  "OwnerSetting": {
    "GroupOwner": "string",
    "GroupPermission": "ro|rw"
  }
},
"S3MachineLearningModelResourceData": {
  "S3Uri": "string",
  "DestinationPath": "string",
  "OwnerSetting": {
    "GroupOwner": "string",
    "GroupPermission": "ro|rw"
  }
},
"SecretsManagerSecretResourceData": {
  "ARN": "string",
  "AdditionalStagingLabelsToDownload": [
    "string"
  ]
}
],
"AmznClientToken": "string"

Parameters:

**ResourceDefinitionId (p. 574)**

The ID of the resource definition.

where used: path; required: true

type: string

**CreateResourceDefinitionVersionRequestBody (p. 562)**

Information about the resource definition version.

where used: body; required: true

```json
{
  "Resources": [
    {
      "Id": "string",
      "Name": "string",
      "ResourceDataContainer": {
        "--- NOTE:": "choose one of the following --",
        "LocalDeviceResourceData": {
          "SourcePath": "string",
          "OwnerSetting": {
            "AutoAddGroupOwner": true,
            "GroupOwner": "string"
          }
        }
      }
    }
  ]
}
```
"LocalVolumeResourceData": {  
  "SourcePath": "string",
  "DestinationPath": "string",
  "GroupOwnerSetting": {  
    "AutoAddGroupOwner": true,
    "GroupOwner": "string"
  }
},
"SageMakerMachineLearningModelResourceData": {  
  "SageMakerJobArn": "string",
  "DestinationPath": "string",
  "OwnerSetting": {  
    "GroupOwner": "string",
    "GroupPermission": "ro|rw"
  }
},
"S3MachineLearningModelResourceData": {  
  "S3Uri": "string",
  "DestinationPath": "string",
  "OwnerSetting": {  
    "GroupOwner": "string",
    "GroupPermission": "ro|rw"
  }
},
"SecretsManagerSecretResourceData": {  
  "ARN": "string",
  "AdditionalStagingLabelsToDownload": [  
    "string"
  ]
}
}

schema:
ResourceDefinitionVersion
  Information about a resource definition version.

  type: object

Resources
  A list of resources.

  type: array

  items: Resource (p. 656)

Resource
  Information about a resource.

  type: object

  required: ["Id", "Name", "ResourceDataContainer"]

Id
  The resource ID, used to refer to a resource in the Lambda function configuration. Maximum length is 128 characters with the pattern \([a-zA-Z0-9_:.-]+\). This must be unique within a Greengrass group.

  type: string
Name

The descriptive resource name, which is displayed on the AWS IoT Greengrass console. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-]+. This must be unique within a Greengrass group.

type: string

ResourceDataContainer

A container for resource data. The container takes only one of the following supported resource data types: LocalDeviceResourceData, LocalVolumeResourceData, SageMakerMachineLearningModelResourceData, S3MachineLearningModelResourceData, SecretsManagerSecretResourceData.

type: object

NOTE: oneOf

LocalDeviceResourceData

Attributes that define a local device resource.

type: object

SourcePath

The local absolute path of the device resource. The source path for a device resource can refer only to a character device or block device under /dev.

type: string

GroupOwnerSetting

Group owner related settings for local resources.

type: object

AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.

type: boolean

GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.

type: string

LocalVolumeResourceData

Attributes that define a local volume resource.

type: object

SourcePath

The local absolute path of the volume resource on the host. The source path for a volume resource type cannot start with /sys.

type: string
DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

GroupOwnerSetting

Group owner related settings for local resources.

type: object

AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.

type: boolean

GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.

type: string

SageMakerMachineLearningModelResourceData

Attributes that define an Amazon SageMaker machine learning resource.

type: object

SageMakerJobArn

The ARN of the Amazon SageMaker training job that represents the source model.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.

type: object

required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are rw (read-write) or ro (read-only).

type: string

enum: ["ro", "rw"]
S3MachineLearningModelResourceData

Attributes that define an Amazon S3 machine learning resource.

type: object

S3Uri

The URI of the source model in an S3 bucket. The model package must be in tar.gz or .zip format.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.

type: object

required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are rw (read-write) or ro (read-only).

type: string

enum: ["ro", "rw"]

SecretsManagerSecretResourceData

Attributes that define a secret resource, which references a secret from AWS Secrets Manager. AWS IoT Greengrass stores a local, encrypted copy of the secret on the Greengrass core, where it can be securely accessed by connectors and Lambda functions.

type: object

ARN

The ARN of the Secrets Manager secret to make available on the core. The value of the secret's latest version (represented by the AWSCURRENT staging label) is included by default.

type: string

AdditionalStagingLabelsToDownload

Optional. The staging labels in addition to AWSCURRENT whose values you want to make available on the core.

type: array

X-Amzn-Client-Token (p. 579)

A client token used to correlate requests and responses.
where used: header; required: false
type: string

**Responses:**

**200** *(CreateResourceDefinitionVersionResponse)*

```
VersionInformation (p. 676)
{
    "Arn": "string",
    "Id": "string",
    "Version": "string",
    "CreationTimestamp": "string"
}
```

VersionInformation
Information about a version.
type: object
Arn
The ARN of the version.
type: string
Id
The ID of the parent definition that the version is associated with.
type: string
Version
The ID of the version.
type: string
CreationTimestamp
The time, in milliseconds since the epoch, when the version was created.
type: string

**400**
Invalid request.

**GeneralError (p. 614)**

```
{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}
```
CreateSoftwareUpdateJob

Creates a software update for a core or group of cores (specified as an IoT thing group.) Use this to update the OTA agent and the Greengrass core software. It uses the AWS IoT Jobs feature, which provides more commands for managing a Greengrass Core software update job.

URI: POST /greengrass/updates

Produces: application/json

CLI:

```
aws greengrass create-software-update-job \
  [--update-targets-architecture <value>] \ 
  [--update-targets <value>] \ 
  [--software-to-update <value>] \ 
  [--s3-url-signer-role <value>] \ 
  [--update-agent-log-level <value>] \ 
  [--update-targets-operating-system <value>] \ 
  [--amzn-client-token <value>] \ 
  [--cli-input-json <value>] \ 
  [--generate-cli-skeleton]
```
cli-input-json format:

```json
{
    "UpdateTargetsArchitecture": "armv6l|armv7l|x86_64|aarch64",
    "UpdateTargets": [
        "string"
    ],
    "SoftwareToUpdate": "core|ota_agent",
    "S3UrlSignerRole": "string",
    "UpdateAgentLogLevel": "NONE|TRACE|DEBUG|VERBOSE|INFO|WARN|ERROR|FATAL",
    "UpdateTargetsOperatingSystem": "ubuntu|raspbian|amazon_linux|openwrt",
    "AmznClientToken": "string"
}
```

Parameters:

CreateSoftwareUpdateJobRequestBody (p. 567)

where used: body; required: true

```json
{
    "UpdateTargetsArchitecture": "armv6l|armv7l|x86_64|aarch64",
    "UpdateTargets": [
        "string"
    ],
    "SoftwareToUpdate": "core|ota_agent",
    "S3UrlSignerRole": "string",
    "UpdateAgentLogLevel": "NONE|TRACE|DEBUG|VERBOSE|INFO|WARN|ERROR|FATAL",
    "UpdateTargetsOperatingSystem": "ubuntu|raspbian|amazon_linux|openwrt"
}
```

schema:
CreateSoftwareUpdateJobRequest

Request for the CreateSoftwareUpdateJob API.

type: object

required: ["UpdateTargetsArchitecture", "UpdateTargets", "SoftwareToUpdate",
"S3UrlSignerRole", "UpdateTargetsOperatingSystem"]

UpdateTargetsArchitecture

The architecture of the cores that are the targets of an update.

type: string

enum: ["armv6l", "armv7l", "x86_64", "aarch64"]

UpdateTargets

The ARNs of the targets (IoT things or IoT thing groups) that this update is applied to.

type: array

SoftwareToUpdate

The piece of software on the Greengrass core that will be updated.

type: string

enum: ["core", "ota_agent"]
S3UrlSignerRole

The IAM role that Greengrass uses to create presigned URLs that point to the update artifact.

type: string

UpdateAgentLogLevel

The minimum level of log statements that should be logged by the OTA agent during an update.

type: string
	enum: ["NONE", "TRACE", "DEBUG", "VERBOSE", "INFO", "WARN", "ERROR", "FATAL"]

UpdateTargetsOperatingSystem

The operating system of the cores that are the targets of an update.

type: string
	enum: ["ubuntu", "raspbian", "amazon_linux", "openwrt"]

X-Amzn-Client-Token (p. 579)

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200

Success.

CreateSoftwareUpdateJobResponse (p. 594)

{
    "IotJobId": "string",
    "IotJobArn": "string",
    "PlatformSoftwareVersion": "string"
}

CreateSoftwareUpdateJobResponse

type: object

IotJobId

The IoT job ID that corresponds to this update.

type: string

IotJobArn

The IoT job ARN that corresponds to this update.

type: string

PlatformSoftwareVersion

The software version installed on the device or devices after the update.

type: string
**400**

Invalid request.

**GeneralError (p. 614)**

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

**GeneralError**

General error information.

- type: object
- required: ["Message"]

**Message**

A message that contains information about the error.

- type: string

**ErrorDetails**

A list of error details.

- type: array
- items: ErrorDetail (p. 600)

**ErrorDetail**

Details about the error.

- type: object

**DetailedErrorCode**

A detailed error code.

- type: string

**DetailedErrorMessage**

A detailed error message.

- type: string

**500**

Server error.

**GeneralError (p. 614)**

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```
CreateSubscriptionDefinition

Creates a subscription definition. You can provide the initial version of the subscription definition now or use CreateSubscriptionDefinitionVersion later.

URI: POST /greengrass/definition/subscriptions

Produces: application/json

CLI:

```bash
aws greengrass create-subscription-definition \
    [--name <value>] \
    [--initial-version <value>] \
    [--tags <value>] \
    [--amzn-client-token <value>] \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```
cli-input-json format:

```json
{
   "Name": "string",
   "InitialVersion": {
      "Subscriptions": [
         {
            "Id": "string",
            "Source": "string",
            "Subject": "string",
            "Target": "string"
         }
      ]
   }
   "tags": {
      "additionalProperty0": "string",
      "additionalProperty1": "string",
      "additionalProperty2": "string"
   },
   "AmznClientToken": "string"
}
```

Parameters:

**CreateSubscriptionDefinitionRequestBody (p. 568)**

where used: body; required: true

```json
{
   "Name": "string",
   "InitialVersion": {
      "Subscriptions": [
         {
            "Id": "string",
            "Source": "string",
            "Subject": "string",
            "Target": "string"
         }
      ]
   }
   "tags": {
      "additionalProperty0": "string",
      "additionalProperty1": "string",
      "additionalProperty2": "string"
   }
}
```

schema:

**Name**

The name of the subscription definition.

- type: string

**InitialVersion**

Information about a subscription definition version.

- type: object

**Subscriptions**

A list of subscriptions.
CreateSubscriptionDefinition

type: array

items: Subscription (p. 672)

Subscription

Information about a subscription.

type: object

required: ["Id", "Source", "Subject", "Target"]

Id

A descriptive or arbitrary ID for the subscription. This value must be unique within the subscription definition version. Maximum length is 128 characters with the pattern `[a-zA-Z0-9:_-]+`.  

type: string

Source

The source of the subscription. Can be a thing ARN, the ARN of a Lambda function alias (recommended) or version, a connector ARN, 'cloud' (which represents AWS IoT), or 'GGShadowService'. If you specify a Lambda function, this ARN should match the ARN used to add the function to the Greengrass group.


type: string

Subject

The MQTT topic used to route the message.


type: string

Target

Where the message is sent. Can be a thing ARN, the ARN of a Lambda function alias (recommended) or version, a connector ARN, 'cloud' (which represents AWS IoT), or 'GGShadowService'. If you specify a Lambda function, this ARN should match the ARN used to add the function to the Greengrass group.


type: string

tags

The resource tags.


type: object

additionalProperties: The key-value pair for the resource tag. Type: string

X-Amzn-Client-Token (p. 579)

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200 (CreateSubscriptionDefinitionResponse)

DefinitionInformation (p. 594)
{
   "Name": "string",
   "Id": "string",
   "Arn": "string",
   "LastUpdatedTimestamp": "string",
   "CreationTimestamp": "string",
   "LatestVersion": "string",
   "LatestVersionArn": "string"
}

DefinitionInformation

Information about a definition.

type: object

Name

The name of the definition.

type: string

Id

The ID of the definition.

type: string

Arn

The ARN of the definition.

type: string

LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.

type: string

LatestVersion

The ID of the latest version associated with the definition.

type: string

LatestVersionArn

The ARN of the latest version associated with the definition.

type: string

400

Invalid request.

GeneralError (p. 614)


"ErrorDetails": [ 
  { 
    "DetailedErrorCode": "string",
    "DetailedErrorMessage": "string"
  }
]

GeneralError
  General error information.
  type: object
  required: ["Message"]
Message
  A message that contains information about the error.
  type: string
ErrorDetails
  A list of error details.
  type: array
  items: ErrorDetail (p. 600)
ErrorDetail
  Details about the error.
  type: object
DetailedErrorCode
  A detailed error code.
  type: string
DetailedErrorMessage
  A detailed error message.
  type: string

CreateSubscriptionDefinitionVersion

Creates a version of a subscription definition that has already been defined.

URI: POST /greengrass/definition/subscriptions/SubscriptionDefinitionId/versions

Produces: application/json

CLI:

```bash
aws greengrass create-subscription-definition-version \
  --subscription-definition-id <value> \
  [--subscriptions <value>] \
  [--amzn-client-token <value>] \
```
CreateSubscriptionDefinitionVersion

[--cli-input-json <value>] \n[--generate-cli-skeleton]

cli-input-json format:

```
{
   "SubscriptionDefinitionId": "string",
   "Subscriptions": [ 
   { 
      "Id": "string",
      "Source": "string",
      "Subject": "string",
      "Target": "string"
   }
   ],
   "AmznClientToken": "string"
}
```

Parameters:

**SubscriptionDefinitionId** *(p. 576)*

The ID of the subscription definition.

where used: path; required: true
type: string

**CreateSubscriptionDefinitionVersionRequestBody** *(p. 568)*

where used: body; required: true

```
{
   "Subscriptions": [ 
   { 
      "Id": "string",
      "Source": "string",
      "Subject": "string",
      "Target": "string"
   }
   ]
}
```

schema:

SubscriptionDefinitionVersion

Information about a subscription definition version.
type: object

Subscriptions

A list of subscriptions.
type: array

items: Subscription *(p. 672)*

Subscription

Information about a subscription.
CreateSubscriptionDefinitionVersion

type: object

required: ["Id", "Source", "Subject", "Target"]

Id

A descriptive or arbitrary ID for the subscription. This value must be unique within the subscription definition version. Maximum length is 128 characters with the pattern `[a-zA-Z0-9:_-]+`.

type: string

Source

The source of the subscription. Can be a thing ARN, the ARN of a Lambda function alias (recommended) or version, a connector ARN, 'cloud' (which represents AWS IoT), or 'GGShadowService'. If you specify a Lambda function, this ARN should match the ARN used to add the function to the Greengrass group.

type: string

Subject

The MQTT topic used to route the message.

type: string

Target

Where the message is sent. Can be a thing ARN, the ARN of a Lambda function alias (recommended) or version, a connector ARN, 'cloud' (which represents AWS IoT), or 'GGShadowService'. If you specify a Lambda function, this ARN should match the ARN used to add the function to the Greengrass group.

type: string

X-Amzn-Client-Token (p. 579)

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200 (CreateSubscriptionDefinitionVersionResponse)

VersionInformation (p. 676)

{
   "Arn": "string",
   "Id": "string",
   "Version": "string",
   "CreationTimestamp": "string"
}

VersionInformation

Information about a version.

type: object
Arn

The ARN of the version.

type: string

Id

The ID of the parent definition that the version is associated with.

type: string

Version

The ID of the version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the version was created.

type: string

400

Invalid request.

GeneralError (p. 614)

```json

{  
    "Message": "string",  
    "ErrorDetails": [  
        {  
            "DetailedErrorCode": "string",  
            "DetailedErrorMessage": "string"  
        }  
    ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.
DeleteConnectorDefinition

Deletes a connector definition.

URI: DELETE /greengrass/definition/connectors/{ConnectorDefinitionId}

Produces: application/json

CLI:

```bash
aws greengrass delete-connector-definition
   --connector-definition-id <value> \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "ConnectorDefinitionId": "string"
}
```

Parameters:

`ConnectorDefinitionId` (p. 546)

The ID of the connector definition.

- type: string
- where used: path; required: true

Responses:

200

Success.

Empty (p. 600)

```json
{
}
```
Empty Schema
Empty
  type: object

400
Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [ {
    "DetailedErrorCode": "string",
    "DetailedErrorMessage": "string"
  } ]
}
```

GeneralError

  General error information.
  type: object
  required: ["Message"]

Message

  A message that contains information about the error.
  type: string

ErrorDetails

  A list of error details.
  type: array
  items: ErrorDetail (p. 600)

ErrorDetail

  Details about the error.
  type: object

DetailedErrorCode

  A detailed error code.
  type: string

DetailedErrorMessage

  A detailed error message.
  type: string

DeleteCoreDefinition

Delete a core definition.
URI: `DELETE /greengrass/definition/cores/CoreDefinitionId`

Produces: application/json

**CLI:**

```
aws greengrass delete-core-definition \
   --core-definition-id <value> \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```
{
   "CoreDefinitionId": "string"
}
```

**Parameters:**

**CoreDefinitionId (p. 547)**

The ID of the core definition.

where used: path; required: true

type: string

**Responses:**

**200**

Success.

**Empty (p. 600)**

```
{
}
```

Empty Schema

Empty
type: object

**400**

Invalid request.

**GeneralError (p. 614)**

```
{
   "Message": "string",
   "ErrorDetails": [
       {
           "DetailedErrorCode": "string",
           "DetailedErrorMessage": "string"
       }
   ]
}
```
DeleteDeviceDefinition

Deletes a device definition.

**URI:** DELETE /greengrass/definition/devices/*DeviceDefinitionId*

**Produces:** application/json

**CLI:**

```
aws greengrass delete-device-definition \
   --device-definition-id <value> \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

**cli-input-json format:**

```json
{
   "DeviceDefinitionId": "string"
}
```
Parameters:

DeviceDefinitionId (p. 570)

The ID of the device definition.

where used: path; required: true

type: string

Responses:

200

Success.

Empty (p. 600)

```
{
}
```

Empty Schema

Empty

type: object

400

Invalid request.

GeneralError (p. 614)

```
{
  "Message": "string",
  "ErrorDetails": [ ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.
DeleteFunctionDefinition

Deletes a Lambda function definition.

URI: DELETE /greengrass/definition/functions/FunctionDefinitionId

Produces: application/json

CLI:

```
aws greengrass delete-function-definition \
   --function-definition-id <value>  \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```
{
   "FunctionDefinitionId": "string"
}
```

Parameters:

**FunctionDefinitionId (p. 571)**

The ID of the function definition.

where used: path; required: true
type: string

Responses:

200
Success.
Empty (p. 600)

```json
{
}
```

Empty Schema

Empty

type: object

400

Invalid request.

GeneralError (p. 614)

```json
{
   "Message": "string",
   "ErrorDetails": [
   {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
   }
   ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.
type: string

DeleteGroup

Deletes a group.

URI: DELETE /greengrass/groups/GroupId

Produces: application/json

CLI:

```bash
aws greengrass delete-group \
  --group-id <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "GroupId": "string"
}
```

Parameters:

GroupId (p. 571)

The ID of the Greengrass group.

where used: path; required: true

type: string

Responses:

200

Success.

Empty (p. 600)

```json
{
}
```

Empty Schema

Empty
type: object

400

Invalid request.

GeneralError (p. 614)
DeleteLoggerDefinition

Deletes a logger definition.

URI: DELETE /greengrass/definition/loggers/

Produces: application/json

CLI:

```bash
aws greengrass delete-logger-definition
```
DeleteLoggerDefinition

```
--logger-definition-id <value>  
[--cli-input-json <value>]  
[--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "LoggerDefinitionId": "string"
}
```

**Parameters:**

**LoggerDefinitionId** *(p. 572)*

The ID of the logger definition.

where used: path; required: true

type: string

**Responses:**

**200**

Success.

*Empty* *(p. 600)*

```
{
}
```

Empty Schema

Empty


type: object

**400**

Invalid request.

*GeneralError* *(p. 614)*

```
{
  "Message": "string",
  "ErrorDetails": [
  
  
  ]
}
```

GeneralError

General error information.


type: object
DeleteResourceDefinition

Deletes a resource definition.

**URI:** DELETE /greengrass/definition/resources/{ResourceDefinitionId}

**Produces:** application/json

**CLI:**

```bash
aws greengrass delete-resource-definition \
--resource-definition-id <value> \ 
[--cli-input-json <value>] \ 
[--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "ResourceDefinitionId": "string"
}
```

**Parameters:**

ResourceDefinitionId (p. 574)

The ID of the resource definition.
where used: path; required: true
type: string

Responses:

200
Success.
Empty (p. 600)
{

}
Empty Schema
Empty
type: object

400
Invalid request.
GeneralError (p. 614)
{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}
GeneralError
General error information.
type: object
required: ["Message"]
Message
A message that contains information about the error.
type: string
ErrorDetails
A list of error details.
type: array
items: ErrorDetail (p. 600)
ErrorDetail
Details about the error.
DeleteSubscriptionDefinition

Deletes a subscription definition.

URI: DELETE /greengrass/definition/subscriptions/SubscriptionDefinitionId

Produces: application/json

CLI:

```
aws greengrass delete-subscription-definition  
  --subscription-definition-id <value>  
  [--cli-input-json <value>]  
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "SubscriptionDefinitionId": "string"
}
```

Parameters:

SubscriptionDefinitionId (p. 576)

The ID of the subscription definition.

where used: path; required: true

type: string

Responses:

200

Success.

Empty (p. 600)
Empty Schema

Empty

type: object

400

Invalid request.

GeneralError (p. 614)

{
   "Message": "string",
   "ErrorDetails": [
      {
         "DetailedErrorCode": "string",
         "DetailedErrorMessage": "string"
      }
   ]
}

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

DisassociateRoleFromGroup

Disassociates the role from a group.
DisassociateRoleFromGroup

URI: DELETE /greengrass/groups/GroupId/role

CLI:

```bash
aws greengrass disassociate-role-from-group \
--group-id <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "GroupId": "string"
}
```

Parameters:

**GroupId (p. 571)**

The ID of the Greengrass group.

where used: path; required: true

type: string

Responses:

**200**

Success.

**DisassociateRoleFromGroupResponse (p. 599)**

```json
{
  "DisassociatedAt": "string"
}
```

**DisassociateRoleFromGroupResponse**

- type: object

  **DisassociatedAt**

  The time, in milliseconds since the epoch, when the role was disassociated from the group.

  type: string

**400**

Invalid request.

**GeneralError (p. 614)**

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
    }
  ]
}
```
GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

500

Server error.

GeneralError (p. 614)
type: object
required: ["Message"]

Message
A message that contains information about the error.
type: string

ErrorDetails
A list of error details.
type: array
items: ErrorDetail (p. 600)

ErrorDetail
Details about the error.
type: object

DetailedErrorCode
A detailed error code.
type: string

DetailedErrorMessage
A detailed error message.
type: string

DisassociateServiceRoleFromAccount
Disassociates the service role from your account. Without a service role, deployments do not work.

URI: DELETE /greengrass/servicerole

CLI:

aws greengrass disassociate-service-role-from-account  
[--cli-input-json <value>]  
[--generate-cli-skeleton]

Responses:

200
Success.

DisassociateServiceRoleFromAccountResponse (p. 599)

{
  "DisassociatedAt": "string"
}
DisassociateServiceRoleFromAccountResponse

type: object

DisassociatedAt

The time when the service role was disassociated from the account.

type: string

500

Server error.

GeneralError (p. 614)

{
   "Message": "string",
   "ErrorDetails": [
      {
         "DetailedErrorCode": "string",
         "DetailedErrorMessage": "string"
      }
   ]
}

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string
GetAssociatedRole

Retrieves the role associated with a group.

URI: GET /greengrass/groups/GroupId/role

**CLI:**

```bash
aws greengrass get-associated-role \
   --group-id <value> \ 
   [--cli-input-json <value>] \ 
   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "GroupId": "string"
}
```

**Parameters:**

**GroupId (p. 571)**

The ID of the Greengrass group.

- where used: path; required: true
- type: string

**Responses:**

**200**

Success.

**GetAssociatedRoleResponse (p. 615)**

```json
{
   "AssociatedAt": "string",
   "RoleArn": "string"
}
```

- **GetAssociatedRoleResponse**
  - type: object
    - **AssociatedAt**
      - The time when the role was associated with the group.
      - type: string
    - **RoleArn**
      - The ARN of the role that is associated with the group.
      - type: string
400

Invalid request.

GeneralError (p. 614)

```
{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

500

Server error.

GeneralError (p. 614)

```
{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}
```
GetBulkDeploymentStatus

Returns the status of a bulk deployment.

URL: GET /greengrass/bulk/deployments/BulkDeploymentId/status

Produces: application/json

**CLI:**

```
aws greengrass get-bulk-deployment-status \\n  --bulk-deployment-id <value> \\n  [--cli-input-json <value>] \\n  [--generate-cli-skeleton]
```

cli-input-json format:
GetBulkDeploymentStatus

Parameters:

**BulkDeploymentId (p. 545)**

The ID of the bulk deployment.

where used: path; required: true
type: string

Responses:

200

Success. The response body contains the status of the bulk deployment.

GetBulkDeploymentStatusResponse (p. 616)

```json
{
  "BulkDeploymentStatus": "Initializing|Running|Completed|Stopping|Stopped|Failed",
  "BulkDeploymentMetrics": {
    "RecordsProcessed": 0,
    "InvalidInputRecords": 0,
    "RetryAttempts": 0
  },
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  },
  "CreatedAt": "string",
  "ErrorMessage": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GetBulkDeploymentStatusResponse

Information about the status of a bulk deployment at the time of the request.
type: object

**BulkDeploymentStatus**

The current status of the bulk deployment.
type: string


**BulkDeploymentMetrics**

Relevant metrics on input records processed during bulk deployment.
type: object

RecordsProcessed

The total number of group records from the input file that have been processed or attempted so far.
type: integer

InvalidInputRecords

The total number of records that returned a non-retryable error. For example, this can occur if a group record from the input file uses an invalid format or specifies a nonexistent group version, or if the execution role doesn't grant permission to deploy a group or group version.
type: integer

RetryAttempts

The total number of deployment attempts that returned a retryable error. For example, a retry is triggered if the attempt to deploy a group returns a throttling error. StartBulkDeployment retries a group deployment up to five times.
type: integer

tags

The resource tags.
type: object

additionalProperties: The key-value pair for the resource tag. Type: string

CreatedAt

The time, in ISO format, when the deployment was created.
type: string

ErrorMessage

Error message
type: string

ErrorDetails

A list of error details.
type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.
type: object

DetailedErrorCode

A detailed error code.
type: string

DetailedErrorMessage

A detailed error message.
GetConnectivityInfo

Retrieves the connectivity information for a core.

URI: GET /greengrass/things/ThingName/connectivityInfo
Produce: application/json

**CLI:**

```bash
aws greengrass get-connectivity-info \
   --thing-name <value>  \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "ThingName": "string"
}
```

**Parameters:**

**ThingName (p. 577)**

The thing name.

*where used: path; required: true*

*type: string*

**Responses:**

**200**

Success.

**GetConnectivityInfoResponse (p. 617)**

```json
{
   "message": "string",
   "ConnectivityInfo": [
      {
         "Id": "string",
         "HostAddress": "string",
         "PortNumber": 0x01,
         "Metadata": "string"
      }
   ]
}
```

Information about a Greengrass core's connectivity.

*message*

A message about the connectivity info request.

*type: string*

*ConnectivityInfo*

Connectivity info list.

*type: array*
items: ConnectivityInfo (p. 588)

Information about a Greengrass core's connectivity.

required: ["Id", "HostAddress"]

Id

The ID of the connectivity information.

type: string

HostAddress

The endpoint for the Greengrass core. Can be an IP address or DNS address.

type: string

PortNumber

The port of the Greengrass core, usually 8883.

type: integer

format: int32

Metadata

Metadata for this endpoint.

type: string

400

Invalid request.

GeneralError (p. 614)

{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}

GeneralError

General error information.

  type: object

  required: ["Message"]

Message

A message that contains information about the error.

  type: string

ErrorDetails

A list of error details.
type: array
items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.
type: object

DetailedErrorCode

A detailed error code.
type: string

DetailedErrorMessage

A detailed error message.
type: string

500

Server error.

GeneralError (p. 614)

{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}

GeneralError

General error information.
type: object

required: ["Message"]

Message

A message that contains information about the error.
type: string

ErrorDetails

A list of error details.
type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.
type: object
GetConnectorDefinition

Retrieves information about a connector definition.

URI: GET /greengrass/definition/connectors/ConnectorDefinitionId

Produces: application/json

CLI:

```bash
aws greengrass get-connector-definition \
  --connector-definition-id <value> \ 
  [--cli-input-json <value>] \ 
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "ConnectorDefinitionId": "string"
}
```

Parameters:

**ConnectorDefinitionId** (p. 546)

The ID of the connector definition.

where used: path; required: true

  type: string

Responses:

**200** (GetConnectorDefinitionResponse)

**DefinitionInformation** (p. 594)

```json
{
  "Name": "string",
  "Id": "string",
  "Arn": "string",
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
  }
}
```
"additionalProperty2": "string",

"LastUpdatedTimestamp": "string",

"CreationTimestamp": "string",

"LatestVersion": "string",

"LatestVersionArn": "string"

}

**DefinitionInformation**

Information about a definition.

type: object

**Name**

The name of the definition.

type: string

**Id**

The ID of the definition.

type: string

**Arn**

The ARN of the definition.

type: string

**tags**

The resource tags.

type: object

  **additionalProperties**

  The key-value pair for the resource tag. Type: string

**LastUpdatedTimestamp**

The time, in milliseconds since the epoch, when the definition was last updated.

type: string

**CreationTimestamp**

The time, in milliseconds since the epoch, when the definition was created.

type: string

**LatestVersion**

The ID of the latest version associated with the definition.

type: string

**LatestVersionArn**

The ARN of the latest version associated with the definition.

type: string

**400**

Invalid request.
GeneralError (p. 614)

```json
{
   "Message": "string",
   "ErrorDetails": [
      {
         "DetailedErrorCode": "string",
         "DetailedErrorMessage": "string"
      }
   ]
}
```

**GeneralError**

General error information.

type: object

required: ["Message"]

**Message**

A message that contains information about the error.

type: string

**ErrorDetails**

A list of error details.

type: array

items: ErrorDetail (p. 600)

**ErrorDetail**

Details about the error.

type: object

**DetailedErrorCode**

A detailed error code.

type: string

**DetailedErrorMessage**

A detailed error message.

type: string

---

**GetConnectorDefinitionVersion**

Retrieves information about a connector definition version, including the connectors that the version contains. Connectors are prebuilt modules that interact with local infrastructure, device protocols, AWS, and other cloud services.

**URI:**

GET /greengrass/definition/connectors/ConnectorDefinitionId/versions/ConnectorDefinitionVersionId

**Produces:** application/json

---
CLI:

```bash
aws greengrass get-connector-definition-version
  --connector-definition-id <value> \
  --connector-definition-version-id <value> \
  [--next-token <value>]  \
  [--cli-input-json <value>] \ 
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "ConnectorDefinitionId": "string",
  "ConnectorDefinitionVersionId": "string",
  "NextToken": "string"
}
```

Parameters:

**ConnectorDefinitionId** *(p. 546)*

The ID of the connector definition.

where used: path; required: true

  type: string

**ConnectorDefinitionVersionId** *(p. 546)*

The ID of the connector definition version. This value maps to the `Version` property of the corresponding `VersionInformation` object, which is returned by `ListConnectorDefinitionVersions` requests. If the version is the last one that was associated with a connector definition, the value also maps to the `LatestVersion` property of the corresponding `DefinitionInformation` object.

where used: path; required: true

  type: string

**NextToken** *(p. 573)*

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

where used: query; required: false

  type: string

Responses:

**200** *(GetConnectorDefinitionVersionResponse)*

GetConnectorDefinitionVersionResponse *(p. 618)*

```json
{
  "Arn": "string",
  "Id": "string",
  "Version": "string",
  "CreationTimestamp": "string",
}
"Definition": {
  "Connectors": [
    {
      "Id": "string",
      "ConnectorArn": "string",
      "Parameters": {
        "additionalProperty0": "string",
        "additionalProperty1": "string",
        "additionalProperty2": "string"
      }
    }
  ],
  "NextToken": "string"
}

GetConnectorDefinitionVersionResponse

  Information about a connector definition version.

  type: object

  Arn

    The ARN of the connector definition version.

    type: string

  Id

    The ID of the connector definition version.

    type: string

  Version

    The version of the connector definition version.

    type: string

  CreationTimestamp

    The time, in milliseconds since the epoch, when the connector definition version was created.

    type: string

  Definition

    Information about the connector definition version, which is a container for connectors.

    type: object

  Connectors

    A list of references to connectors in this version, with their corresponding configuration settings.

    type: array

    items: Connector (p. 588)

  Connector

    Information about a connector. Connectors run on the Greengrass core and contain built-in integration with local infrastructure, device protocols, AWS, and other cloud services.

    type: object
required: ["Id", "ConnectorArn"]

Id
A descriptive or arbitrary ID for the connector. This value must be unique within the connector
definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-]+.

type: string

ConnectorArn
The ARN of the connector.

type: string

Parameters
The parameters or configuration used by the connector.

type: object

additionalProperties: The key-value pair for the resource tag. Type: string

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

type: string

400
Invalid request.

GeneralError (p. 614)

{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}

GeneralError
General error information.

type: object

required: ["Message"]

Message
A message that contains information about the error.

type: string

ErrorDetails
A list of error details.

type: array

items: ErrorDetail (p. 600)
ErrorDetail
   Details about the error.
   type: object
DetailedErrorCode
   A detailed error code.
   type: string
DetailedErrorMessage
   A detailed error message.
   type: string

GetCoreDefinition
Retrieves information about a core definition version.

URI: GET /greengrass/definition/cores/CoreDefinitionId
Produces: application/json

CLI:
aws greengrass get-core-definition \  
--core-definition-id <value> \  
[--cli-input-json <value>] \  
[--generate-cli-skeleton]

cli-input-json format:
{
   "CoreDefinitionId": "string"
}

Parameters:

CoreDefinitionId (p. 547)
   The ID of the core definition.
   where used: path; required: true
   type: string

Responses:

200 (GetCoreDefinitionResponse)
   DefinitionInformation (p. 594)
   
   {
      "Name": "string",
   

"Id": "string",
"Arn": "string",
"tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
},
"LastUpdatedTimestamp": "string",
"CreationTimestamp": "string",
"LatestVersion": "string",
"LatestVersionArn": "string"

DefinitionInformation

Information about a definition.

type: object

Name

The name of the definition.

type: string

Id

The ID of the definition.

type: string

Arn

The ARN of the definition.

type: string

tags

The resource tags.

type: object

    additionalProperties: The key-value pair for the resource tag. Type: string

LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.

type: string

LatestVersion

The ID of the latest version associated with the definition.

type: string

LatestVersionArn

The ARN of the latest version associated with the definition.

type: string
400
Invalid request.

**GeneralError (p. 614)**

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

---

**GeneralError**

General error information.

- **type**: object
- **required**: ["Message"]

**Message**

A message that contains information about the error.

- **type**: string

**ErrorDetails**

A list of error details.

- **type**: array
- **items**: ErrorDetail (p. 600)

**ErrorDetail**

Details about the error.

- **type**: object

**DetailedErrorCode**

A detailed error code.

- **type**: string

**DetailedErrorMessage**

A detailed error message.

- **type**: string

---

**GetCoreDefinitionVersion**

Retrieves information about a core definition version.

**URI**: GET /greengrass/definition/cores/CoreDefinitionId/versions/CoreDefinitionVersionId

**Produces**: application/json
CLI:

```bash
aws greengrass get-core-definition-version
  --core-definition-id <value> 
  --core-definition-version-id <value> 
  [--cli-input-json <value>] 
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "CoreDefinitionId": "string",
   "CoreDefinitionVersionId": "string"
}
```

Parameters:

- **CoreDefinitionId** *(p. 547)*
  - The ID of the core definition.
  - where used: path; required: true
  - type: string

- **CoreDefinitionVersionId** *(p. 547)*
  - The ID of the core definition version. This value maps to the Version property of the corresponding VersionInformation object, which is returned by ListCoreDefinitionVersions requests. If the version is the last one that was associated with a core definition, the value also maps to the LatestVersion property of the corresponding DefinitionInformation object.
  - where used: path; required: true
  - type: string

Responses:

- **200**
  - Success.
  - **GetCoreDefinitionVersionResponse** *(p. 620)*

```json
{
   "Arn": "string",
   "Id": "string",
   "Version": "string",
   "CreationTimestamp": "string",
   "Definition": {
      "Cores": [
         {
            "Id": "string",
            "ThingArn": "string",
            "CertificateArn": "string",
            "SyncShadow": true
         }
      ]
   }
}
```
GetCoreDefinitionVersion

"NextToken": "string"
}

GetCoreDefinitionVersionResponse

type: object

Arn

The ARN of the core definition version.

type: string

Id

The ID of the core definition version.

type: string

Version

The version of the core definition version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the core definition version was created.

type: string

Definition

Information about a core definition version.

type: object

Cores

A list of cores in the core definition version.

type: array

items: Core (p. 590)

Core

Information about a core.

type: object

required: ["Id", "ThingArn", "CertificateArn"]

Id

A descriptive or arbitrary ID for the core. This value must be unique within the core definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9_\-]+.

type: string

ThingArn

The ARN of the thing that is the core.

type: string

CertificateArn

The ARN of the certificate associated with the core.
type: string
SyncShadow

If true, the core's local shadow is synced with the cloud automatically.

type: boolean
NextToken

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

type: string

400
Invalid request.

GeneralError (p. 614)

```
{
  "Message": "string",
  "ErrorDetails": [  
    {  
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError

General error information.

type: object
required: ["Message"]
Message

A message that contains information about the error.

type: string
ErrorDetails

A list of error details.

type: array
items: ErrorDetail (p. 600)
ErrorDetail

Details about the error.

type: object
DetailedErrorCode

A detailed error code.

type: string
DetailedErrorMessage

A detailed error message.
type: string

**GetDeploymentStatus**

Returns the status of a deployment.

**URI:** GET /greengrass/groups/*GroupId*/deployments/*DeploymentId*/status

**Produce:** application/json

**CLI:**

```bash
aws greengrass get-deployment-status
   --group-id <value>
   --deployment-id <value>
   [--cli-input-json <value>]
   [--generate-cli-skeleton]
```

**cli-input-json format:**

```json
{
   "GroupId": "string",
   "DeploymentId": "string"
}
```

### Parameters:

**GroupId** *(p. 571)*

The ID of the Greengrass group.

- where used: path; required: true
- type: string

**DeploymentId** *(p. 570)*

The ID of the deployment.

- where used: path; required: true
- type: string

### Responses:

**200**

Success. The response body contains the status of the deployment for the group.

GetDeploymentStatusResponse *(p. 621)*

```json
{
   "DeploymentStatus": "string",
   "DeploymentType": "NewDeployment|Redeployment|ResetDeployment|ForceResetDeployment",
   "UpdatedAt": "string",
   "ErrorMessage": "string",
}
```
GetDeploymentStatus

Information about the status of a deployment for a group.

- **DeploymentStatus**: The status of the deployment: Building, InProgress, Success, or Failure.
  - **DeploymentType**: The type of deployment. When used for `CreateDeployment`, only NewDeployment and Redeployment are valid.
  - **UpdatedAt**: The time, in milliseconds since the epoch, when the deployment status was updated.
  - **ErrorMessage**: Error message
  - **ErrorDetails**: A list of error details.
    - **DetailedErrorCode**: A detailed error code.
    - **DetailedErrorMessage**: A detailed error message.
400

Invalid request.

**GeneralError (p. 614)**

```
{
   "Message": "string",
   "ErrorDetails": [
      {
         "DetailedErrorCode": "string",
         "DetailedErrorMessage": "string"
      }
   ]
}
```

**GeneralError**

General error information.

- type: object
- required: ["Message"]

**Message**

A message that contains information about the error.

- type: string

**ErrorDetails**

A list of error details.

- type: array

- items: [**ErrorDetail (p. 600)**]

**ErrorDetail**

Details about the error.

- type: object

**DetailedErrorCode**

A detailed error code.

- type: string

**DetailedErrorMessage**

A detailed error message.

- type: string

---

**GetDeviceDefinition**

Retrieves information about a device definition.

**URI:** GET /greengrass/definition/devices/*DeviceDefinitionId*

**Produces:** application/json
CLI:

```
aws greengrass get-device-definition \
   --device-definition-id <value> \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```
{
   "DeviceDefinitionId": "string"
}
```

Parameters:

**DeviceDefinitionId (p. 570)**

The ID of the device definition.

*where used: path; required: true*

*type: string*

Responses:

**200 (GetDeviceDefinitionResponse)**

```
DefinitionInformation (p. 594)
```

```
{
   "Name": "string",
   "Id": "string",
   "Arn": "string",
   "tags": {
      "additionalProperty0": "string",
      "additionalProperty1": "string",
      "additionalProperty2": "string"
   },
   "LastUpdatedTimestamp": "string",
   "CreationTimestamp": "string",
   "LatestVersion": "string",
   "LatestVersionArn": "string"
}
```

**DefinitionInformation**

Information about a definition.

*type: object*

**Name**

The name of the definition.

*type: string*

**Id**

The ID of the definition.
type: string
Arn
The ARN of the definition.
type: string
tags
The resource tags.
type: object
additionalProperties: The key-value pair for the resource tag. Type: string
LastUpdatedTimestamp
The time, in milliseconds since the epoch, when the definition was last updated.
type: string
CreationTimestamp
The time, in milliseconds since the epoch, when the definition was created.
type: string
LatestVersion
The ID of the latest version associated with the definition.
type: string
LatestVersionArn
The ARN of the latest version associated with the definition.
type: string

400
Invalid request.

GeneralError (p. 614)

```
{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}
```

GeneralError
General error information.
type: object
required: ["Message"]
Message
A message that contains information about the error.
type: string
ErrorDetails
A list of error details.
type: array
items: ErrorDetail (p. 600)
ErrorDetail
Details about the error.
type: object
DetailedErrorCode
A detailed error code.
type: string
DetailedErrorMessage
A detailed error message.
type: string

GetDeviceDefinitionVersion
Retrieves information about a device definition version.

URI: GET /greengrass/definition/devices/{DeviceDefinitionId}/versions/{DeviceDefinitionVersionId}

Produces: application/json

CLI:

```
aws greengrass get-device-definition-version
    --device-definition-id <value> \
    --device-definition-version-id <value> \
    [--next-token <value>] \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

cli-input-json format:

```
{
    "DeviceDefinitionId": "string",
    "DeviceDefinitionVersionId": "string",
    "NextToken": "string"
}
```

Parameters:

DeviceDefinitionId (p. 570)
The ID of the device definition.
DeviceDefinitionVersionId (p. 570)

The ID of the device definition version. This value maps to the Version property of the corresponding VersionInformation object, which is returned by ListDeviceDefinitionVersions requests. If the version is the last one that was associated with a device definition, the value also maps to the LatestVersion property of the corresponding DefinitionInformation object.

NextToken (p. 573)

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

Responses:

200 (GetDeviceDefinitionVersionResponse)

GetDeviceDefinitionVersionResponse (p. 623)

```
{
    "Arn": "string",
    "Id": "string",
    "Version": "string",
    "CreationTimestamp": "string",
    "Definition": {
        "Devices": [
            {
                "Id": "string",
                "ThingArn": "string",
                "CertificateArn": "string",
                "SyncShadow": true
            }
        ],
        "NextToken": "string"
    }
}
```

GetDeviceDefinitionVersionResponse

- type: object
- Arn
  The ARN of the device definition version.
  - type: string
- Id
  The ID of the device definition version.
  - type: string
Version

The version of the device definition version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the device definition version was created.

type: string

Definition

Information about a device definition version.

type: object

Devices

A list of devices in the definition version.

type: array

items: Device (p. 597)

Device

Information about a device.

type: object

required: ["Id", "ThingArn", "CertificateArn"]

Id

A descriptive or arbitrary ID for the device. This value must be unique within the device definition version. Maximum length is 128 characters with the pattern `[a-zA-Z0-9_:.-]`.+

type: string

ThingArn

The thing ARN of the device.

type: string

CertificateArn

The ARN of the certificate associated with the device.

type: string

SyncShadow

If true, the device's local shadow is synced with the cloud automatically.

type: boolean

NextToken

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

type: string

400

Invalid request.
**GetFunctionDefinition**

Retrieves information about a Lambda function definition, including its creation time and latest version.

**URI:** GET /greengrass/definition/functions/FunctionDefinitionId

**Produces:** application/json

**CLI:**

```
aws greengrass get-function-definition \
```
GetFunctionDefinition

--function-definition-id <value>  
[--cli-input-json <value>]  
[--generate-cli-skeleton]

cli-input-json format:

```json
{
  "FunctionDefinitionId": "string"
}
```

Parameters:

FunctionDefinitionId (p. 571)

The ID of the function definition.

where used: path; required: true

type: string

Responses:

200 (GetFunctionDefinitionResponse)

Success.

DefinitionInformation (p. 594)

```json
{
  "Name": "string",
  "Id": "string",
  "Arn": "string",
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  },
  "LastUpdatedTimestamp": "string",
  "CreationTimestamp": "string",
  "LatestVersion": "string",
  "LatestVersionArn": "string"
}
```

DefinitionInformation

Information about a definition.

type: object

Name

The name of the definition.

type: string

Id

The ID of the definition.

type: string
Arn
- The ARN of the definition.
  type: string

tags
- The resource tags.
  type: object
  additionalProperties: The key-value pair for the resource tag. Type: string

LastUpdatedTimestamp
- The time, in milliseconds since the epoch, when the definition was last updated.
  type: string

CreationTimestamp
- The time, in milliseconds since the epoch, when the definition was created.
  type: string

LatestVersion
- The ID of the latest version associated with the definition.
  type: string

LatestVersionArn
- The ARN of the latest version associated with the definition.
  type: string

400
- Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError
- General error information.
  type: object
  required: ["Message"]

Message
- A message that contains information about the error.
  type: string
GetFunctionDefinitionVersion

Retrieves information about a Lambda function definition version, including which Lambda functions are included in the version and their configurations.

URI: GET /greengrass/definition/functions/FunctionDefinitionId/versions/FunctionDefinitionVersionId

Produces: application/json

CLI:

```bash
aws greengrass get-function-definition-version
   --function-definition-id <value> \
   --function-definition-version-id <value> \
   [--next-token <value>] \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "FunctionDefinitionId": "string",
   "FunctionDefinitionVersionId": "string",
   "NextToken": "string"
}
```

Parameters:

FunctionDefinitionId (p. 571)
   The ID of the function definition.
   where used: path; required: true
type: string

**FunctionDefinitionVersionId (p. 571)**

The ID of the function definition version. This value maps to the `Version` property of the corresponding `VersionInformation` object, which is returned by `ListFunctionDefinitionVersions` requests. If the version is the last one that was associated with a function definition, the value also maps to the `LatestVersion` property of the corresponding `DefinitionInformation` object.

where used: path; required: true

type: string

**NextToken (p. 573)**

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

where used: query; required: false

type: string

**Responses:**

200

Success.

GetFunctionDefinitionVersionResponse (p. 624)

```json
{
    " Arn": "string",
    "Id": "string",
    "Version": "string",
    "CreationTimestamp": "string",
    "Definition": {
        "DefaultConfig": {
            "Execution": {
                "IsolationMode": "GreengrassContainer|NoContainer",
                "RunAs": {
                    "Uid": 1001,
                    "Gid": 1002
                }
            }
        },
        "Functions": [
            {
                "Id": "string",
                "FunctionArn": "string",
                "FunctionConfiguration": {
                    "Pinned": true,
                    "Executable": "string",
                    "ExecArgs": "string",
                    "MemorySize": 0,
                    "Timeout": 0,
                    "EncodingType": "binary|json",
                    "Environment": {
                        "Variables": {
                            "additionalProperty0": "string",
                            "additionalProperty1": "string",
                            "additionalProperty2": "string"
                        }
                    }
                }
            }
        ],
        "ResourceAccessPolicies": [
```
GetFunctionDefinitionVersion

Information about a function definition version.

type: object

Arn

The ARN of the function definition version.

type: string

Id

The ID of the function definition version.

type: string

Version

The version of the function definition version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the function definition version was created.

type: string

Definition

Information about a function definition version.

type: object

DefaultConfig

The default configuration that applies to all Lambda functions in the group. Individual Lambda functions can override these settings.

type: object

Execution

Configuration information that specifies how a Lambda function runs.
Functions

A list of Lambda functions in this function definition version.

- type: array
- items: Function (p. 601)

Information about a Lambda function.

- type: object
- required: ["Id"]

Id

A descriptive or arbitrary ID for the function. This value must be unique within the function definition version. Maximum length is 128 characters with the pattern \[a-zA-Z0-9_:\-\+]+\.

- type: string

FunctionArn

The ARN of the alias (recommended) or version of the target Lambda function.

- type: string

FunctionConfiguration

The configuration of the Lambda function.

- type: object

Pinned

True if the function is pinned. Pinned means the function is long-lived and starts when the core starts.

- type: boolean

Executable

The name of the function executable.

- type: string

ExecArgs

The execution arguments.

- type: string

MemorySize

The memory size, in KB, required by the function. This setting does not apply and should be cleared when you run the Lambda function without containerization.

- type: integer

Timeout

The allowed function execution time, after which Lambda should terminate the function. This timeout still applies to pinned Lambda functions for each request.

- type: integer
EncodingType

The expected encoding type of the input payload for the function. The default is `json`.

type: string
enum: ["binary", "json"]

Environment

The environment configuration of the function.

type: object

Variables

Environment variables for the Lambda function's configuration.

type: object

additionalProperties: An object with properties of type `string` that represent the environment variables.

ResourceAccessPolicies

A list of the resources, with their permissions, to which the Lambda function is granted access. A Lambda function can have at most 10 resources. ResourceAccessPolicies applies only when you run the Lambda function in a Greengrass container.

type: array

items: `ResourceAccessPolicy (p. 660)`

ResourceAccessPolicy

A policy used by the function to access a resource.

type: object

required: ["ResourceId"]

ResourceId

The ID of the resource. (This ID is assigned to the resource when you create the resource definition.)

type: string

Permission

The type of permission a function has to access a resource.

type: string

enum: ["ro", "rw"]

AccessSysfs

If true, the Lambda function is allowed to access the host's /sys folder. Use this when the Lambda function needs to read device information from /sys. This setting applies only when you run the Lambda function in a Greengrass container.

type: boolean

Execution

Configuration information that specifies how a Lambda function runs.
type: object

IsolationMode

Specifies whether the Lambda function runs in a Greengrass container (default) or without containerization. Unless your scenario requires that you run without containerization, we recommend that you run in a Greengrass container. Omit this value to run the Lambda function with the default containerization for the group.

type: string
enum: ["GreengrassContainer", "NoContainer"]

RunAs

Specifies the user and group whose permissions are used when running the Lambda function. You can specify one or both values to override the default values. To minimize the risk of unintended changes or malicious attacks, we recommend that you avoid running as root unless absolutely necessary. To run as root, you must update config.json in `greengrass-root/config` to set `allowFunctionsToRunAsRoot` to `true`.

type: object

Uid

The user ID whose permissions are used to run a Lambda function.

type: integer

Gid

The group ID whose permissions are used to run a Lambda function.

type: integer

NextToken

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

type: string

400

Invalid request.

GeneralError (p. 614)

```json
{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]
Message
   A message that contains information about the error.
   type: string
ErrorDetails
   A list of error details.
   type: array
   items: ErrorDetail (p. 600)
ErrorDetail
   Details about the error.
   type: object
DetailedErrorCode
   A detailed error code.
   type: string
DetailedErrorMessage
   A detailed error message.
   type: string

GetGroup

Retrieves information about a group.

URI: GET /greengrass/groups/GroupId

Produces: application/json

CLI:

```bash
aws greengrass get-group
   --group-id <value>
   [--cli-input-json <value>] 
   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "GroupId": "string"
}
```

Parameters:

**GroupId (p. 571)**

The ID of the Greengrass group.
where used: path; required: true
type: string

Responses:

200 (GetGroupResponse)

Success.

DefinitionInformation (p. 594)

{
    "Name": "string",
    "Id": "string",
    "Arn": "string",
    "tags": {
        "additionalProperty0": "string",
        "additionalProperty1": "string",
        "additionalProperty2": "string"
    },
    "LastUpdatedTimestamp": "string",
    "CreationTimestamp": "string",
    "LatestVersion": "string",
    "LatestVersionArn": "string"
}

DefinitionInformation

Information about a definition.

type: object

Name

The name of the definition.

type: string

Id

The ID of the definition.

type: string

Arn

The ARN of the definition.

type: string

Tags

The resource tags.

type: object

    additionalProperties: The key-value pair for the resource tag. Type: string

LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.
type: string

LatestVersion

The ID of the latest version associated with the definition.

type: string

LatestVersionArn

The ARN of the latest version associated with the definition.

type: string

400

Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.
GetGroupCertificateAuthority

Retrieves the CA associated with a group. Returns the public key of the CA.

URI: GET /greengrass/groups/GroupId/certificateauthorities/CertificateAuthorityId

Produces: application/json

CLI:

```
aws greengrass get-group-certificate-authority \
  --group-id <value> \
  --certificate-authority-id <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "GroupId": "string",
  "CertificateAuthorityId": "string"
}
```

Parameters:

**GroupId (p. 571)**

The ID of the Greengrass group.

where used: path; required: true

  type: string

**CertificateAuthorityId (p. 546)**

The ID of the certificate authority.

where used: path; required: true

  type: string

Responses:

**200**

Success. The response body contains the PKI configuration.

GetGroupCertificateAuthorityResponse (p. 628)

```
{
  "PemEncodedCertificate": "string",
  "GroupCertificateAuthorityArn": "string",
  "GroupCertificateAuthorityId": "string"
}
```
GetGroupCertificateAuthorityResponse

Information about a certificate authority for a group.

- **PemEncodedCertificate**
  - The PEM encoded certificate for the group.
  - type: string
- **GroupCertificateAuthorityArn**
  - The ARN of the certificate authority for the group.
  - type: string
- **GroupCertificateAuthorityId**
  - The ID of the certificate authority for the group.
  - type: string

400

Invalid request.

**GeneralError (p. 614)**

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

**GeneralError**

- General error information.
  - type: object
  - required: ["Message"]

**Message**

- A message that contains information about the error.
  - type: string

**ErrorDetails**

- A list of error details.
  - type: array
  - items: **ErrorDetail (p. 600)**

**ErrorDetail**

- Details about the error.
type: object
DetailedErrorCode
    A detailed error code.
    type: string
DetailedErrorMessage
    A detailed error message.
    type: string

500
Server error.

GeneralError (p. 614)

{
    "Message": "string",
    "ErrorDetails": [
    {
        "DetailedErrorCode": "string",
        "DetailedErrorMessage": "string"
    }
    ]
}

GeneralError
General error information.

type: object
required: ["Message"]

ErrorMessage
A message that contains information about the error.

type: string

ErrorDetails
A list of error details.

type: array
items: ErrorDetail (p. 600)

ErrorDetail
Details about the error.

type: object

DetailedErrorCode
A detailed error code.

type: string

DetailedErrorMessage
A detailed error message.
GetGroupCertificateConfiguration

Retrieves the current configuration for the CA used by the group.

URI: GET /greengrass/groups/GroupId/certificateauthorities/configuration/expiry

Produces: application/json

CLI:

```bash
aws greengrass get-group-certificate-configuration \
  --group-id <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "GroupId": "string"
}
```

Parameters:

**GroupId** *(p. 571)*

The ID of the Greengrass group.

- where used: path; required: true
- type: string

Responses:

**200**

Success. The response body contains the PKI configuration.

**GroupCertificateConfiguration** *(p. 639)*

```json
{
  "GroupId": "string",
  "CertificateAuthorityExpiryInMilliseconds": "string",
  "CertificateExpiryInMilliseconds": "string"
}
```

**GroupCertificateConfiguration**

Information about a group certificate configuration.

- type: object
- GroupId
  - The ID of the group certificate configuration.
CertificateAuthorityExpiryInMilliseconds

The amount of time, in milliseconds, before the certificate authority expires.

certificate authority expires.

certificate expires.

certificate expires.

400

Invalid request.

GeneralError (p. 614)

{
   "Message": "string",
   "ErrorDetails": [
   {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
   }
   ]
}

GeneralError

General error information.

certificate expires.

certificate expires.

400

Invalid request.

GeneralError (p. 614)

{
   "Message": "string",
   "ErrorDetails": [
   {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
   }
   ]
}

GeneralError

General error information.

certificate expires.

certificate expires.

400

Invalid request.
type: string

500

Server error.

GeneralError (p. 614)

{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}

GeneralError
  General error information.
  type: object
  required: ["Message"]
Message
  A message that contains information about the error.
  type: string
ErrorDetails
  A list of error details.
  type: array
  items: ErrorDetail (p. 600)
ErrorDetail
  Details about the error.
  type: object
DetailedErrorCode
  A detailed error code.
  type: string
DetailedErrorMessage
  A detailed error message.
  type: string

GetGroupVersion

Retrieves information about a group version.

URI: GET /greengrass/groups/GroupId/versions/GroupVersionId
PRODUCES: application/json

CLI:

```
aws greengrass get-group-version \
  --group-id <value> \
  --group-version-id <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "GroupId": "string",
  "GroupVersionId": "string"
}
```

PARAMETERS:

**GroupId** (p. 571)

The ID of the Greengrass group.

where used: path; required: true

type: string

**GroupVersionId** (p. 572)

The ID of the group version. This value maps to the Version property of the corresponding VersionInformation object, which is returned by ListGroupVersions requests. If the version is the last one that was associated with a group, the value also maps to the LatestVersion property of the corresponding GroupInformation object.

where used: path; required: true

type: string

RESPONSES:

**200**

Success.

GetGroupVersionResponse (p. 629)

```
{
  "Arn": "string",
  "Id": "string",
  "Version": "string",
  "CreationTimestamp": "string",
  "Definition": {
    "CoreDefinitionVersionArn": "string",
    "DeviceDefinitionVersionArn": "string",
    "FunctionDefinitionVersionArn": "string",
    "SubscriptionDefinitionVersionArn": "string",
    "LoggerDefinitionVersionArn": "string",
    "ResourceDefinitionVersionArn": "string",
```

157
GetGroupVersion

GetGroupVersionResponse
  Information about a group version.
  type: object
  Arn
    The ARN of the group version.
    type: string
  Id
    The ID of the group that the version is associated with.
    type: string
  Version
    The ID of the group version.
    type: string
  CreationTimestamp
    The time, in milliseconds since the epoch, when the group version was created.
    type: string
  Definition
    Information about a group version.
    type: object
    CoreDefinitionVersionArn
      The ARN of the core definition version for this group.
      type: string
    DeviceDefinitionVersionArn
      The ARN of the device definition version for this group.
      type: string
    FunctionDefinitionVersionArn
      The ARN of the function definition version for this group.
      type: string
    SubscriptionDefinitionVersionArn
      The ARN of the subscription definition version for this group.
      type: string
    LoggerDefinitionVersionArn
      The ARN of the logger definition version for this group.
      type: string
ResourceDefinitionVersionArn

    The ARN of the resource definition version for this group.

    type: string

ConnectorDefinitionVersionArn

    The ARN of the connector definition version for this group.

    type: string

400

Invalid request.

**GeneralError (p. 614)**

```json
{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}
```

**GeneralError**

General error information.

    type: object

    required: ["Message"]

**Message**

A message that contains information about the error.

    type: string

**ErrorDetails**

A list of error details.

    type: array

    items: **ErrorDetail (p. 600)**

**ErrorDetail**

Details about the error.

    type: object

**DetailedErrorCode**

A detailed error code.

    type: string

**DetailedErrorMessage**

A detailed error message.

    type: string
GetLoggerDefinition

Retrieves information about a logger definition.

URI: GET /greengrass/definition/loggers/LoggerDefinitionId

Produces: application/json

CLI:

```
aws greengrass get-logger-definition
   --logger-definition-id <value> \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```
{
   "LoggerDefinitionId": "string"
}
```

Parameters:

`LoggerDefinitionId` (p. 572)

The ID of the logger definition.

where used: path; required: true

type: string

Responses:

200 (GetLoggerDefinitionResponse)

```
DefinitionInformation (p. 594)
```

```
{
   "Name": "string",
   "Id": "string",
   "Arn": "string",
   "tags": {
      "additionalProperty0": "string",
      "additionalProperty1": "string",
      "additionalProperty2": "string"
   },
   "LastUpdatedTimestamp": "string",
   "CreationTimestamp": "string",
   "LatestVersion": "string",
   "LatestVersionArn": "string"
}
```

DefinitionInformation

Information about a definition.

type: object
Name
  The name of the definition.
  type: string

Id
  The ID of the definition.
  type: string

Arn
  The ARN of the definition.
  type: string
tags
  The resource tags.
  type: object
  additionalProperties: The key-value pair for the resource tag. Type: string

LastUpdatedTimestamp
  The time, in milliseconds since the epoch, when the definition was last updated.
  type: string

CreationTimestamp
  The time, in milliseconds since the epoch, when the definition was created.
  type: string

LatestVersion
  The ID of the latest version associated with the definition.
  type: string

LatestVersionArn
  The ARN of the latest version associated with the definition.
  type: string

400
Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```
GetLoggerDefinitionVersion

Retrieves information about a logger definition version.

URI: GET /greengrass/definition/loggers/LoggerDefinitionId/versions/LoggerDefinitionVersionId

Produces: application/json

CLI:

```
aws greengrass get-logger-definition-version \ 
   --logger-definition-version-id <value> \ 
   --logger-definition-id <value> \ 
   [--next-token <value>] \ 
   [--cli-input-json <value>] \ 
   [--generate-cli-skeleton]
```

cli-input-json format:

```
{
```
GetLoggerDefinitionVersion

```
"LoggerDefinitionVersionId": "string",
"LoggerDefinitionId": "string",
"NextToken": "string"
}
```

**Parameters:**

**LoggerDefinitionVersionId** *(p. 572)*

The ID of the logger definition version. This value maps to the `Version` property of the corresponding `VersionInformation` object, which is returned by `ListLoggerDefinitionVersions` requests. If the version is the last one that was associated with a logger definition, the value also maps to the `LatestVersion` property of the corresponding `DefinitionInformation` object.

- where used: path; required: true
- type: string

**LoggerDefinitionId** *(p. 572)*

The ID of the logger definition.

- where used: path; required: true
- type: string

**NextToken** *(p. 573)*

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

- where used: query; required: false
- type: string

**Responses:**

**200** *(GetLoggerDefinitionVersionResponse)*

Success.

**GetLoggerDefinitionVersionResponse** *(p. 630)*

```
{
  "Arn": "string",
  "Id": "string",
  "Version": "string",
  "CreationTimestamp": "string",
  "Definition": {
    "Loggers": [
      {
        "Id": "string",
        "Type": "FileSystem|AWSCloudWatch",
        "Component": "GreengrassSystem|Lambda",
        "Level": "DEBUG/INFO/WARN/ERROR/FATAL",
        "Space": 0
      }
    ]
  }
}
```
GetLoggerDefinitionVersionResponse

Information about a logger definition version.

type: object

Arn

The ARN of the logger definition version.

type: string

Id

The ID of the logger definition version.

type: string

Version

The version of the logger definition version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the logger definition version was created.

type: string

Definition

Information about a logger definition version.

type: object

Loggers

A list of loggers.

type: array

items: Logger (p. 652)

Logger

Information about a logger

type: object

required: ["Id", "Type", "Component", "Level"]

Id

A descriptive or arbitrary ID for the logger. This value must be unique within the logger definition version. Maximum length is 128 characters with the pattern \[a-zA-Z0-9_:.-] +\).

type: string

Type


type: string

enum: ["FileSystem", "AWSCloudWatch"]

Component


type: string
enum: ["GreengrassSystem", "Lambda"]

Level

type: string

enum: ["DEBUG", "INFO", "WARN", "ERROR", "FATAL"]

Space

The amount of file space, in KB, to use if the local file system is used for logging purposes.

type: integer

400

Invalid request.

**GeneralError (p. 614)**

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

**GeneralError**

General error information.

type: object

required: ["Message"]

**Message**

A message that contains information about the error.

type: string

**ErrorDetails**

A list of error details.

type: array

items: **ErrorDetail (p. 600)**

**ErrorDetail**

Details about the error.

type: object

**DetailedErrorCode**

A detailed error code.

type: string

**DetailedErrorMessage**

A detailed error message.
type: string

**GetResourceDefinition**

Retrieves information about a resource definition, including its creation time and latest version.

**URL:** GET  /greengrass/definition/resources/*ResourceDefinitionId*

**Produces:** application/json

**CLI:**

```bash
aws greengrass get-resource-definition \
  --resource-definition-id <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "ResourceDefinitionId": "string"
}
```

**Parameters:**

**ResourceDefinitionId (p. 574)**

The ID of the resource definition.

where used: path; required: true

type: string

**Responses:**

**200 (GetResourceDefinitionResponse)**

Success.

**DefinitionInformation (p. 594)**

```json
{
  "Name": "string",
  "Id": "string",
  "Arn": "string",
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  },
  "LastUpdatedTimestamp": "string",
  "CreationTimestamp": "string",
  "LatestVersion": "string",
  "LatestVersionArn": "string"
}
```
DefinitionInformation
  Information about a definition.
  type: object
Name
  The name of the definition.
  type: string
Id
  The ID of the definition.
  type: string
Arn
  The ARN of the definition.
  type: string
tags
  The resource tags.
  type: object
  additionalProperties: The key-value pair for the resource tag. Type: string
LastUpdatedTimestamp
  The time, in milliseconds since the epoch, when the definition was last updated.
  type: string
CreationTimestamp
  The time, in milliseconds since the epoch, when the definition was created.
  type: string
LatestVersion
  The ID of the latest version associated with the definition.
  type: string
LatestVersionArn
  The ARN of the latest version associated with the definition.
  type: string

400
  Invalid request.

GeneralError (p. 614)

{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]}
GetResourceDefinitionVersion

Retrieves information about a resource definition version, including which resources are included in the version.

URL: GET /greengrass/definition/resources/ResourceDefinitionId/versions/ResourceDefinitionVersionId

Produces: application/json

CLI:

```
aws greengrass get-resource-definition-version
  --resource-definition-id <value>
  --resource-definition-version-id <value>
  [--cli-input-json <value>]
  [--generate-cli-skeleton]
```
**GetResourceDefinitionVersion**

**cli-input-json format:**

```json
{
  "ResourceDefinitionId": "string",
  "ResourceDefinitionVersionId": "string"
}
```

**Parameters:**

**ResourceDefinitionId (p. 574)**

The ID of the resource definition.

where used: path; required: true

type: string

**ResourceDefinitionVersionId (p. 574)**

The ID of the resource definition version. This value maps to the `Version` property of the corresponding `VersionInformation` object, which is returned by `ListResourceDefinitionVersions` requests. If the version is the last one that was associated with a resource definition, the value also maps to the `LatestVersion` property of the corresponding `DefinitionInformation` object.

where used: path; required: true

type: string

**Responses:**

**200**

Success.

**GetResourceDefinitionVersionResponse (p. 632)**

```json
{
  "Arn": "string",
  "Id": "string",
  "Version": "string",
  "CreationTimestamp": "string",
  "Definition": {
    "Resources": [
      {
        "Id": "string",
        "Name": "string",
        "ResourceDataContainer": {
          "--- NOTE:" : "choose one of the following ---",
          "LocalDeviceResourceData": {
            "SourcePath": "string",
            "GroupOwnerSetting": {
              "AutoAddGroupOwner": true,
              "GroupOwner": "string"
            }
          },
          "LocalVolumeResourceData": {
            "SourcePath": "string",
            "DestinationPath": "string",
            "GroupOwnerSetting": {
```

169
"AutoAddGroupOwner": true,
"GroupOwner": "string"
}
},
"SageMakerMachineLearningModelResourceData": {
"SageMakerJobArn": "string",
"DestinationPath": "string",
"OwnerSetting": {
  "GroupOwner": "string",
  "GroupPermission": "ro|rw"
}
},
"S3MachineLearningModelResourceData": {
"S3Uri": "string",
"DestinationPath": "string",
"OwnerSetting": {
  "GroupOwner": "string",
  "GroupPermission": "ro|rw"
}
},
"SecretsManagerSecretResourceData": {
"ARN": "string",
"AdditionalStagingLabelsToDownload": [
  "string"
]
}
]}
}
}

GetResourceDefinitionVersionResponse

Information about a resource definition version.

type: object

Arn

The ARN of the resource definition version.

type: string

Id

The ID of the resource definition version.

type: string

Version

The version of the resource definition version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the resource definition version was created.

type: string

Definition

Information about a resource definition version.

type: object
Resources

A list of resources.

type: array

definition:

- items: Resource (p. 656)

Resource

Information about a resource.

type: object

required: ["Id", "Name", "ResourceDataContainer"]

Id

The resource ID, used to refer to a resource in the Lambda function configuration. Maximum length is 128 characters with the pattern \[a-zA-Z0-9:_-]+\]. This must be unique within a Greengrass group.

type: string

Name

The descriptive resource name, which is displayed on the AWS IoT Greengrass console. Maximum length is 128 characters with the pattern \[a-zA-Z0-9:_-]+\]. This must be unique within a Greengrass group.

required: Name

ResourceDataContainer

A container for resource data. The container takes only one of the following supported resource data types: LocalDeviceResourceData, LocalVolumeResourceData, SageMakerMachineLearningModelResourceData, S3MachineLearningModelResourceData, SecretsManagerSecretResourceData.

type: object

NOTE: oneOf

LocalDeviceResourceData

Attributes that define a local device resource.

required: SourcePath

SourcePath

The local absolute path of the device resource. The source path for a device resource can refer only to a character device or block device under /dev.

type: string

GroupOwnerSetting

Group owner related settings for local resources.

required: AutoAddGroupOwner

AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.
type: boolean

GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.

type: string

LocalVolumeResourceData

Attributes that define a local volume resource.

type: object

SourcePath

The local absolute path of the volume resource on the host. The source path for a volume resource type cannot start with /sys.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

GroupOwnerSetting

Group owner related settings for local resources.

type: object

AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.

type: boolean

GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.

type: string

SageMakerMachineLearningModelResourceData

Attributes that define an Amazon SageMaker machine learning resource.

type: object

SageMakerJobArn

The ARN of the Amazon SageMaker training job that represents the source model.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.
type: object

required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are `rw` (read-write) or `ro` (read-only).

type: string

enum: ["ro", "rw"]

S3MachineLearningModelResourceData

Attributes that define an Amazon S3 machine learning resource.

type: object

S3Uri

The URI of the source model in an S3 bucket. The model package must be in `tar.gz` or `.zip` format.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.

type: object

required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are `rw` (read-write) or `ro` (read-only).

type: string

enum: ["ro", "rw"]

SecretsManagerSecretResourceData

Attributes that define a secret resource, which references a secret from AWS Secrets Manager. AWS IoT Greengrass stores a local, encrypted copy of the secret on the Greengrass core, where it can be securely accessed by connectors and Lambda functions.
type: object

ARN

The ARN of the Secrets Manager secret to make available on the core. The value of the secret's latest version (represented by the AWSCURRENT staging label) is included by default.

type: string

AdditionalStagingLabelsToDownload

Optional. The staging labels in addition to AWSCURRENT whose values you want to make available on the core.

type: array

400

Invalid request.

GeneralError (p. 614)

```
{
  "Message": "string",
  "ErrorDetails": [
  {
    "DetailedErrorCode": "string",
    "DetailedErrorMessage": "string"
  }
  ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string
DetailedErrorMessage

A detailed error message.
type: string

GetServiceRoleForAccount

Retrieves the service role that is attached to your account.

URI: GET /greengrass/servicerole

CLI:

```
aws greengrass get-service-role-for-account
  [--cli-input-json <value>]
  [--generate-cli-skeleton]
```

Responses:

200

Success.

GetServiceRoleForAccountResponse (p. 636)

```json
{
  "AssociatedAt": "string",
  "RoleArn": "string"
}
```

GetServiceRoleForAccountResponse

type: object

AssociatedAt

The time when the service role was associated with the account.
type: string

RoleArn

The ARN of the role associated with the account.
type: string

500

Server error.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```
GetSubscriptionDefinition

Retrieves information about a subscription definition.

URI: GET /greengrass/definition/subscriptions/SubscriptionDefinitionId

Produces: application/json

CLI:

```
aws greengrass get-subscription-definition
   --subscription-definition-id <value> \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```
{
   "SubscriptionDefinitionId": "string"
}
```
Parameters:

SubscriptionDefinitionId (p. 576)

The ID of the subscription definition.

where used: path; required: true
type: string

Responses:

200 (GetSubscriptionDefinitionResponse)

DefinitionInformation (p. 594)

```json
{
  "Name": "string",
  "Id": "string",
  "Arn": "string",
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  },
  "LastUpdatedTimestamp": "string",
  "CreationTimestamp": "string",
  "LatestVersion": "string",
  "LatestVersionArn": "string"
}
```

DefinitionInformation

Information about a definition.
type: object

Name

The name of the definition.
type: string

Id

The ID of the definition.
type: string

Arn

The ARN of the definition.
type: string

tags

The resource tags.
type: object
additionalProperties: The key-value pair for the resource tag. Type: string

LastUpdatedTimestamp
The time, in milliseconds since the epoch, when the definition was last updated.
type: string

CreationTimestamp
The time, in milliseconds since the epoch, when the definition was created.
type: string

LatestVersion
The ID of the latest version associated with the definition.
type: string

LatestVersionArn
The ARN of the latest version associated with the definition.
type: string

400
Invalid request.

GeneralError (p. 614)

```json
{
   "Message": "string",
   "ErrorDetails": [
       {
           "DetailedErrorCode": "string",
           "DetailedErrorMessage": "string"
       }
   ]
}
```

GeneralError
General error information.
type: object
required: ["Message"]

Message
A message that contains information about the error.
type: string

ErrorDetails
A list of error details.
type: array
items: ErrorDetail (p. 600)

ErrorDetail
Details about the error.
GetSubscriptionDefinitionVersion

Retrieves information about a subscription definition version.

URI: GET /greengrass/definition/subscriptions/{SubscriptionDefinitionId}/versions/{SubscriptionDefinitionVersionId}

Produces: application/json

CLI:

```bash
aws greengrass get-subscription-definition-version
  --subscription-definition-id <value>
  --subscription-definition-version-id <value>
  [--next-token <value>]
  [--cli-input-json <value>]
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "SubscriptionDefinitionId": "string",
  "SubscriptionDefinitionVersionId": "string",
  "NextToken": "string"
}
```

Parameters:

- **SubscriptionDefinitionId** *(p. 576)*
  
  The ID of the subscription definition.
  
  where used: path; required: true
  
  type: string

- **SubscriptionDefinitionVersionId** *(p. 576)*
  
  The ID of the subscription definition version. This value maps to the `Version` property of the corresponding `VersionInformation` object, which is returned by `ListSubscriptionDefinitionVersions` requests. If the version is the last one that was associated with a subscription definition, the value also maps to the `LatestVersion` property of the corresponding `DefinitionInformation` object.
  
  where used: path; required: true
type: string

**NextToken (p. 573)**

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

where used: query; required: false

type: string

**Responses:**

**200 (GetSubscriptionDefinitionVersionResponse)**

GetSubscriptionDefinitionVersionResponse (p. 637)

```
{
  "Arn": "string",
  "Id": "string",
  "Version": "string",
  "CreationTimestamp": "string",
  "Definition": {
    "Subscriptions": [
      {
        "Id": "string",
        "Source": "string",
        "Subject": "string",
        "Target": "string"
      }
    ],
  },
  "NextToken": "string"
}
```

GetSubscriptionDefinitionVersionResponse

Information about a subscription definition version.

type: object

**Arn**

The ARN of the subscription definition version.

type: string

**Id**

The ID of the subscription definition version.

type: string

**Version**

The version of the subscription definition version.

type: string

**CreationTimestamp**

The time, in milliseconds since the epoch, when the subscription definition version was created.

type: string
Definition

Information about a subscription definition version.

type: object

Subscriptions

A list of subscriptions.

type: array

items: Subscription (p. 672)

Subscription

Information about a subscription.

type: object

required: ["Id", "Source", "Subject", "Target"]

Id

A descriptive or arbitrary ID for the subscription. This value must be unique within the subscription definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-]+.

type: string

Source

The source of the subscription. Can be a thing ARN, the ARN of a Lambda function alias (recommended) or version, a connector ARN, 'cloud' (which represents AWS IoT), or 'GGShadowService'. If you specify a Lambda function, this ARN should match the ARN used to add the function to the Greengrass group.

type: string

Subject

The MQTT topic used to route the message.

type: string

Target

Where the message is sent. Can be a thing ARN, the ARN of a Lambda function alias (recommended) or version, a connector ARN, 'cloud' (which represents AWS IoT), or 'GGShadowService'. If you specify a Lambda function, this ARN should match the ARN used to add the function to the Greengrass group.

type: string

NextToken

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

type: string

400

Invalid request.

GeneralError (p. 614)
ListBulkDeploymentDetailedReports

Gets a paginated list of the deployments that have been started in a bulk deployment operation, and their current deployment status.

URI: GET /greengrass/bulk/deployments/BulkDeploymentId/detailed-reports

Produces: application/json

**CLI:**

```bash
cmd aws greengrass list-bulk-deployment-detailed-reports \\   --bulk-deployment-id <value> \\   [--max-results <value>] \\```
ListBulkDeploymentDetailedReports

[--next-token <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]

cli-input-json format:

{
  "BulkDeploymentId": "string",
  "MaxResults": "integer",
  "NextToken": "string"
}

Parameters:

**BulkDeploymentId (p. 545)**

The ID of the bulk deployment.

where used: path; required: true

type: string

**MaxResults (p. 573)**

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

**NextToken (p. 573)**

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

where used: query; required: false

type: string

Responses:

200

Success. The response body contains the list of deployments for the given group.

ListBulkDeploymentDetailedReportsResponse (p. 642)

{
  "Deployments": [ 
    {
      "DeploymentId": "string",
      "DeploymentArn": "string",
      "GroupArn": "string",
      "DeploymentType": "NewDeployment|Redeployment|ResetDeployment|ForceResetDeployment",
      "CreatedAt": "string",
      "DeploymentStatus": "string",
      "ErrorMessage": "string",
      "ErrorDetails": [ 
        {
          "DetailedErrorCode": "string",
          "DetailedErrorMessage": "string"
        }
      ]
    }
  ]
ListBulkDeploymentDetailedReports

ListBulkDeploymentDetailedReportsResponse

  type: object

Deployments

  type: array

  items: BulkDeploymentResult (p. 584)

BulkDeploymentResult

  Information about an individual group deployment in a bulk deployment operation.

  type: object

DeploymentId

  The ID of the group deployment.

  type: string

DeploymentArn

  The ARN of the group deployment.

  type: string

GroupArn

  The ARN of the Greengrass group.

  type: string

DeploymentType

  The type of deployment. When used for CreateDeployment, only NewDeployment and Redeployment are valid.

  type: string

  enum: ["NewDeployment", "Redeployment", "ResetDeployment", "ForceResetDeployment"]

CreatedAt

  The time, in ISO format, when the deployment was created.

  type: string

DeploymentStatus

  The current status of the group deployment: `Building`, `InProgress`, `Success`, or `Failure`.

  type: string

ErrorMessage

  The error message for a failed deployment.

  type: string
ErrorDetails
A list of error details.
type: array
items: ErrorDetail (p. 600)

ErrorDetail
Details about the error.
type: object

DetailedErrorCode
A detailed error code.
type: string

DetailedErrorMessage
A detailed error message.
type: string

NextToken
The token for the next set of results, or null if there are no more results.
in: query
type: string

400
Invalid request.

GeneralError (p. 614)

{  "Message": "string",
   "ErrorDetails": [
      {
         "DetailedErrorCode": "string",
         "DetailedErrorMessage": "string"
      }
   ]
}

GeneralError
General error information.
type: object
required: ["Message"]

Message
A message that contains information about the error.
type: string

ErrorDetails
A list of error details.
ListBulkDeployments

Returns a list of bulk deployments.

URI: GET /greengrass/bulk/deployments

Produces: application/json

CLI:

```bash
aws greengrass list-bulk-deployments
    [--max-results <value>] \
    [--next-token <value>] \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
    "MaxResults": "integer",
    "NextToken": "string"
}
```

Parameters:

**MaxResults (p. 573)**

The maximum number of results to be returned per request.

where used: query; required: false

 type: integer

**NextToken (p. 573)**

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

where used: query; required: false

 type: string
type: string

**Responses:**

**200**

Success. The response body contains the list of bulk deployments.

[ListBulkDeploymentsResponse (p. 644)]

```json
{
   "BulkDeployments": [
      {
         "BulkDeploymentId": "string",
         "BulkDeploymentArn": "string",
         "CreatedAt": "string"
      },
      {
         "BulkDeploymentId": "string",
         "BulkDeploymentArn": "string",
         "CreatedAt": "string"
      }
   ],
   "NextToken": "string"
}
```

[ListBulkDeploymentsResponse](

**BulkDeployments**

**BulkDeployment**

Information about a bulk deployment. You cannot start a new bulk deployment while another one is still running or in a non-terminal state.

**BulkDeploymentId**

The ID of the bulk deployment.

**BulkDeploymentArn**

The ARN of the bulk deployment.

**CreatedAt**

The time, in ISO format, when the deployment was created.

**NextToken**

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

in: query

type: string
400
Invalid request.

GeneralError (p. 614)

```
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

**GeneralError**

General error information.

- type: object
- required: ["Message"]

**Message**

A message that contains information about the error.

- type: string

**ErrorDetails**

A list of error details.

- type: array

  - **items**: ErrorDetail (p. 600)

**ErrorDetail**

Details about the error.

- type: object

  - **DetailedErrorCode**
  - type: string

  - **DetailedErrorMessage**
  - type: string

---

**ListConnectorDefinitionVersions**

Lists the versions of a connector definition, which are containers for connectors. Connectors run on the Greengrass core and contain built-in integration with local infrastructure, device protocols, AWS, and other cloud services.

**URI**: `GET /greengrass/definition/connectors/ConnectorDefinitionId/versions`
Produce: application/json

CLI:

```
aws greengrass list-connector-definition-versions \
  --connector-definition-id <value> \
  [--next-token <value>] \
  [--max-results <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

Cli-input-json format:

```
{
  "ConnectorDefinitionId": "string",
  "NextToken": "string",
  "MaxResults": "integer"
}
```

Parameters:

**ConnectorDefinitionId** *(p. 546)*

The ID of the connector definition.

where used: path; required: true
type: string

**NextToken** *(p. 573)*

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

where used: query; required: false
type: string

**MaxResults** *(p. 573)*

The maximum number of results to be returned per request.

where used: query; required: false
type: integer

Responses:

**200** *(ListConnectorDefinitionVersionsResponse)*

```
ListVersionsResponse (p. 650)
```

```
{
  "Versions": [
    {
      "Arn": "string",
      "Id": "string",
      "Version": "string",
      "CreationTimestamp": "string"
    }
  ]
}
```
ListVersionsResponse

A list of versions.

type: object

Versions

Information about a version.

type: array

items: VersionInformation (p. 676)

VersionInformation

Information about a version.

type: object

Arn

The ARN of the version.

type: string

Id

The ID of the parent definition that the version is associated with.

type: string

Version

The ID of the version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the version was created.

type: string

NextToken

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

type: string

400

Invalid request.

GeneralError (p. 614)
GeneralError
  General error information.
  type: object
  required: ["Message"]
Message
  A message that contains information about the error.
  type: string
ErrorDetails
  A list of error details.
  type: array
  items: ErrorDetail (p. 600)
ErrorDetail
  Details about the error.
  type: object
DetailedErrorCode
  A detailed error code.
  type: string
DetailedErrorMessage
  A detailed error message.
  type: string

ListConnectorDefinitions
Retrieves a list of connector definitions.

URI: GET /greengrass/definition/connectors

Produces: application/json

CLI:

```bash
aws greengrass list-connector-definitions \
  [--max-results <value>] \
  [--next-token <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  
```
"MaxResults": "integer",
"NextToken": "string"
}

Parameters:

MaxResults (p. 573)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

NextToken (p. 573)

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

where used: query; required: false

type: string

Responses:

200 (ListConnectorDefinitionsResponse)

ListDefinitionsResponse (p. 645)

{
"Definitions": [
 {
 "Name": "string",
 "Id": "string",
 "Arn": "string",
 "tags": {
 "additionalProperty0": "string",
 "additionalProperty1": "string",
 "additionalProperty2": "string"
 },
 "LastUpdatedTimestamp": "string",
 "CreationTimestamp": "string",
 "LatestVersion": "string",
 "LatestVersionArn": "string"
 }
],
"NextToken": "string"
}

ListDefinitionsResponse

A list of definitions.

type: object

Definitions

Information about a definition.

type: array

items: DefinitionInformation (p. 594)
DefinitionInformation
  Information about a definition.
  type: object
Name
  The name of the definition.
  type: string
Id
  The ID of the definition.
  type: string
Arn
  The ARN of the definition.
  type: string
tags
  The resource tags.
  type: object
    additionalProperties: The key-value pair for the resource tag. Type: string
LastUpdatedTimestamp
  The time, in milliseconds since the epoch, when the definition was last updated.
  type: string
CreationTimestamp
  The time, in milliseconds since the epoch, when the definition was created.
  type: string
LatestVersion
  The ID of the latest version associated with the definition.
  type: string
LatestVersionArn
  The ARN of the latest version associated with the definition.
  type: string
NextToken
  The token for the next set of results, or null if there are no more results.
  type: string

ListCoreDefinitionVersions
Lists the versions of a core definition.

URI: GET /greengrass/definition/cores/CoreDefinitionId/versions
**CLI:**

```
aws greengrass list-core-definition-versions \
  --core-definition-id <value> \
  [--next-token <value>] \
  [--max-results <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "CoreDefinitionId": "string",
  "NextToken": "string",
  "MaxResults": "integer"
}
```

**Parameters:**

**CoreDefinitionId (p. 547)**

The ID of the core definition.

Where used: path; required: true

type: string

**NextToken (p. 573)**

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

Where used: query; required: false

type: string

**MaxResults (p. 573)**

The maximum number of results to be returned per request.

Where used: query; required: false

type: integer

**Responses:**

**200 (ListCoreDefinitionVersionsResponse)**

```
ListVersionsResponse (p. 650)
```

```
{
  "Versions": [
    {
      "Arn": "string",
      "Id": "string",
      "Version": "string",
      "CreationTimestamp": "string"
    }
  ]
}
ListCoreDefinitionVersions

A list of versions.

- **NextToken**: string

**ListVersionsResponse**

- **Versions**
  - Information about a version.
  - **Type**: array
  - **Items**: `VersionInformation` (p. 676)

**VersionInformation**

  - Information about a version.
  - **Type**: object

**Arn**

  - The ARN of the version.
  - **Type**: string

**Id**

  - The ID of the parent definition that the version is associated with.
  - **Type**: string

**Version**

  - The ID of the version.
  - **Type**: string

**CreationTimestamp**

  - The time, in milliseconds since the epoch, when the version was created.
  - **Type**: string

**NextToken**

  - The token for the next set of results, or `null` if there are no more results.
  - **Type**: string

**400**

Invalid request.

**GeneralError** (p. 614)

- **Message**: string
- **ErrorDetails**: [ ]
GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

ListCoreDefinitions

Retrieves a list of core definitions.

URI: GET /greengrass/definition/cores

Produces: application/json

CLI:

```
aws greengrass list-core-definitions \    
[--max-results <value>] \   
[--next-token <value>] \   
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

cli-input-json format:

```
{
```
"MaxResults": "integer",
"NextToken": "string"
}

Parameters:

**MaxResults (p. 573)**

The maximum number of results to be returned per request.

where used: query; required: false
type: integer

**NextToken (p. 573)**

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

where used: query; required: false
type: string

Responses:

**200 (ListCoreDefinitionsResponse)**

ListDefinitionsResponse (p. 645)

```json
{
  "Definitions": [
    {
      "Name": "string",
      "Id": "string",
      "Arn": "string",
      "tags": {
        "additionalProperty0": "string",
        "additionalProperty1": "string",
        "additionalProperty2": "string"
      },
      "LastUpdatedTimestamp": "string",
      "CreationTimestamp": "string",
      "LatestVersion": "string",
      "LatestVersionArn": "string"
    }
  ],
  "NextToken": "string"
}
```

ListDefinitionsResponse

A list of definitions.
type: object

Definitions

Information about a definition.
type: array

items: DefinitionInformation (p. 594)
DefinitionInformation

Information about a definition.
type: object

Name

The name of the definition.
type: string

Id

The ID of the definition.
type: string

Arn

The ARN of the definition.
type: string

tags

The resource tags.
type: object

additionalProperties: The key-value pair for the resource tag. Type: string

LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.
type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.
type: string

LatestVersion

The ID of the latest version associated with the definition.
type: string

LatestVersionArn

The ARN of the latest version associated with the definition.
type: string

NextToken

The token for the next set of results, or null if there are no more results.
type: string

ListDeployments

Returns a history of deployments for the group.
AWS Greengrass API Reference
ListDeployments

**URI:** GET /greengrass/groups/**GroupId**/deployments

**Produces:** application/json

**CLI:**

```
aws greengrass list-deployments \
  --group-id <value> \ 
  [ --max-results <value> ] \ 
  [ --next-token <value> ] \ 
  [ --cli-input-json <value> ] \ 
  [ --generate-cli-skeleton ]
```

cli-input-json format:

```
{
  "GroupId": "string",
  "MaxResults": "integer",
  "NextToken": "string"
}
```

**Parameters:**

**GroupId (p. 571)**

The ID of the Greengrass group.

where used: path; required: true

type: string

**MaxResults (p. 573)**

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

**NextToken (p. 573)**

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

where used: query; required: false

type: string

**Responses:**

**200**

Success. The response body contains the list of deployments for the given group.

ListDeploymentsResponse (p. 647)

```
{
  "Deployments": [
  
```
ListDeployments

```
"GroupArn": "string",
"DeploymentId": "string",
"DeploymentArn": "string",
"DeploymentType": "NewDeployment|Redeployment|ResetDeployment|ForceResetDeployment",
"CreatedAt": "string"
```

ListDeploymentsResponse

type: object

Deployments

type: array

items: Deployment (p. 596)

Deployment

Information about a deployment.

type: object

GroupArn

The ARN of the group for this deployment.

type: string

DeploymentId

The ID of the deployment.

type: string

DeploymentArn

The ARN of the deployment.

type: string

DeploymentType

The type of deployment. When used for CreateDeployment, only NewDeployment and Redeployment are valid.

type: string

enum: ["NewDeployment", "Redeployment", "ResetDeployment", "ForceResetDeployment"]

CreatedAt

The time, in milliseconds since the epoch, when the deployment was created.

type: string

NextToken

The token for the next set of results, or null if there are no more results.
in: query
type: string
400

Invalid request.

**GeneralError (p. 614)**

```json
{
   "Message": "string",
   "ErrorDetails": [
       {
           "DetailedErrorCode": "string",
           "DetailedErrorMessage": "string"
       }
   ]
}
```

**GeneralError**

General error information.

- **type**: object
- **required**: ["Message"]

**Message**

A message that contains information about the error.

- **type**: string

**ErrorDetails**

A list of error details.

- **type**: array

  - **items**: **ErrorDetail (p. 600)**

**ErrorDetail**

Details about the error.

- **type**: object

  - **DetailedErrorCode**
    
    A detailed error code.
    
    - **type**: string

  - **DetailedErrorMessage**
    
    A detailed error message.
    
    - **type**: string

**ListDeviceDefinitionVersions**

Lists the versions of a device definition.

**URI**: GET /greengrass/definition/devices/*DeviceDefinitionId*/versions

**Produces**: application/json
CLI:

aws greengrass list-device-definition-versions \ 
--device-definition-id <value> \ 
[--next-token <value>] \ 
[--max-results <value>] \ 
[--cli-input-json <value>] \ 
[--generate-cli-skeleton]

cli-input-json format:

{  "DeviceDefinitionId": "string",  "NextToken": "string",  "MaxResults": "integer"}

Parameters:

DeviceDefinitionId (p. 570)

The ID of the device definition.

where used: path; required: true
type: string

NextToken (p. 573)

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

where used: query; required: false
type: string

MaxResults (p. 573)

The maximum number of results to be returned per request.

where used: query; required: false
type: integer

Responses:

200 (ListDeviceDefinitionVersionsResponse)

ListVersionsResponse (p. 650)

{  "Versions": [  {   "Arn": "string",   "Id": "string",   "Version": "string",   "CreationTimestamp": "string"  }  ],  "NextToken": "string"
ListDeviceDefinitionVersions

A list of versions.

type: object

Versions

Information about a version.

type: array

items: VersionInformation (p. 676)

VersionInformation

Information about a version.

type: object

Arn

The ARN of the version.

type: string

Id

The ID of the parent definition that the version is associated with.

type: string

Version

The ID of the version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the version was created.

type: string

NextToken

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

type: string

400

Invalid request.

GeneralError (p. 614)
ListDeviceDefinitions

Retrieves a list of device definitions.

URI: GET /greengrass/definition/devices

Produces: application/json

CLI:

```bash
aws greengrass list-device-definitions \
  [--max-results <value>] \
  [--next-token <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string
"MaxResults": "integer",
"NextToken": "string"
}

Parameters:

MaxResults (p. 573)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

NextToken (p. 573)

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

where used: query; required: false

type: string

Responses:

200 (ListDeviceDefinitionsResponse)

ListDefinitionsResponse (p. 645)

```json
{
  "Definitions": [
    {
      "Name": "string",
      "Id": "string",
      "Arn": "string",
      "tags": {
        "additionalProperty0": "string",
        "additionalProperty1": "string",
        "additionalProperty2": "string"
      },
      "LastUpdatedTimestamp": "string",
      "CreationTimestamp": "string",
      "LatestVersion": "string",
      "LatestVersionArn": "string"
    }
  ],
  "NextToken": "string"
}
```

ListDefinitionsResponse

A list of definitions.

type: object

Definitions

Information about a definition.

type: array

items: DefinitionInformation (p. 594)
DefinitionInformation
  Information about a definition.
  type: object
Name
  The name of the definition.
  type: string
Id
  The ID of the definition.
  type: string
Arn
  The ARN of the definition.
  type: string
tags
  The resource tags.
  type: object
  additionalProperties: The key-value pair for the resource tag. Type: string
LastUpdatedTimestamp
  The time, in milliseconds since the epoch, when the definition was last updated.
  type: string
CreationTimestamp
  The time, in milliseconds since the epoch, when the definition was created.
  type: string
LatestVersion
  The ID of the latest version associated with the definition.
  type: string
LatestVersionArn
  The ARN of the latest version associated with the definition.
  type: string
NextToken
  The token for the next set of results, or null if there are no more results.
  type: string

ListFunctionDefinitionVersions
Lists the versions of a Lambda function definition.

URL: GET /greengrass/definition/functions/FunctionDefinitionId/versions
CLI:

```
aws greengrass list-function-definition-versions \
   --function-definition-id <value> \
   [--next-token <value>] \
   [--max-results <value>] \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```
{
    "FunctionDefinitionId": "string",
    "NextToken": "string",
    "MaxResults": "integer"
}
```

Parameters:

**FunctionDefinitionId** (p. 571)

The ID of the function definition.

where used: path; required: true

type: string

**NextToken** (p. 573)

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

where used: query; required: false

type: string

**MaxResults** (p. 573)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

Responses:

**200** (ListFunctionDefinitionVersionsResponse)

```
{
    "Versions": [
        {
            "Arn": "string",
            "Id": "string",
            "Version": "string",
            "CreationTimestamp": "string"
        }
    ]
}
```
AWS Greengrass API Reference
ListFunctionDefinitionVersions

},
   "NextToken": "string"
}

ListVersionsResponse

A list of versions.

type: object

Versions

Information about a version.

type: array

items: VersionInformation (p. 676)

VersionInformation

Information about a version.

type: object

Arn

The ARN of the version.

type: string

Id

The ID of the parent definition that the version is associated with.

type: string

Version

The ID of the version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the version was created.

type: string

NextToken

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

type: string

400

Invalid request.

GeneralError (p. 614)
ListFunctionDefinitions

Retrieves a list of Lambda function definitions.

URI: GET /greengrass/definition/functions

Produce: application/json

CLI:

```
aws greengrass list-function-definitions
   [--max-results <value>] \ 
   [--next-token <value>] \ 
   [--cli-input-json <value>] \ 
   [--generate-cli-skeleton]
```

cli-input-json format:

```
{
    "MaxResults": "integer",
```
"NextToken": "string"
}

**Parameters:**

**MaxResults (p. 573)**

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

**NextToken (p. 573)**

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

where used: query; required: false

type: string

**Responses:**

**200** (ListFunctionDefinitionsResponse)

Success. The response contains the IDs of all of the Greengrass Lambda function definitions in this account.

**ListDefinitionsResponse (p. 645)**

```json
{
  "Definitions": [
    {
      "Name": "string",
      "Id": "string",
      "Arn": "string",
      "tags": {
        "additionalProperty0": "string",
        "additionalProperty1": "string",
        "additionalProperty2": "string"
      },
      "LastUpdatedTimestamp": "string",
      "CreationTimestamp": "string",
      "LatestVersion": "string",
      "LatestVersionArn": "string"
    }
  ],
  "NextToken": "string"
}
```

ListDefinitionsResponse

A list of definitions.

type: object

Definitions

Information about a definition.

type: array
items: `DefinitionInformation (p. 594)`

`DefinitionInformation`

Information about a definition.

`type: object`

`Name`

The name of the definition.

`type: string`

`Id`

The ID of the definition.

`type: string`

`Arn`

The ARN of the definition.

`type: string`

`tage`

The resource tags.

`type: object`

`additionalProperties: The key-value pair for the resource tag. Type: string`

`LastUpdatedTimestamp`

The time, in milliseconds since the epoch, when the definition was last updated.

`type: string`

`CreationTimestamp`

The time, in milliseconds since the epoch, when the definition was created.

`type: string`

`LatestVersion`

The ID of the latest version associated with the definition.

`type: string`

`LatestVersionArn`

The ARN of the latest version associated with the definition.

`type: string`

`NextToken`

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

`type: string`

**ListGroupCertificateAuthorities**

Retrieves the current CAs for a group.
List Group Certificate Authorities

**URI:** GET /greengrass/groups/{GroupId}/certificateauthorities

**Produce:** application/json

**CLI:**

```
aws greengrass list-group-certificate-authorities
    --group-id <value> \
    [--cli-input-json <value>] \n    [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
    "GroupId": "string"
}
```

**Parameters:**

**GroupId (p. 571)**

The ID of the Greengrass group.

where used: path; required: true

```text
type: string
```

**Responses:**

**200**

Success. The response body contains the PKI configuration.

**ListGroupCertificateAuthoritiesResponse (p. 648)**

```json
{
    "GroupCertificateAuthorities": [
        {
            "GroupCertificateAuthorityId": "string",
            "GroupCertificateAuthorityArn": "string"
        }
    ]
}
```

**ListGroupCertificateAuthoritiesResponse**

```text
type: object
```

**GroupCertificateAuthorities**

A list of certificate authorities associated with the group.

```text
type: array
```

**items:** **GroupCertificateAuthorityProperties (p. 639)**

**Group Certificate Authority Properties**

Information about a certificate authority for a group.
type: object
GroupCertificateAuthorityId
  The ID of the certificate authority for the group.
  type: string
GroupCertificateAuthorityArn
  The ARN of the certificate authority for the group.
  type: string

400
Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError
  General error information.
  type: object
  required: ["Message"]
Message
  A message that contains information about the error.
  type: string
ErrorDetails
  A list of error details.
  type: array
  items: ErrorDetail (p. 600)
ErrorDetail
  Details about the error.
  type: object
DetailedErrorCode
  A detailed error code.
  type: string
DetailedErrorMessage
  A detailed error message.
type: string

500

Server error.

**GeneralError (p. 614)**

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

**GeneralError**

General error information.

- **type**: object
  - **required**: ["Message"]
  - **Message**: A message that contains information about the error.
    - **type**: string
  - **ErrorDetails**: A list of error details.
    - **type**: array
      - **items**: **ErrorDetail (p. 600)**
  - **ErrorDetail**: Details about the error.
    - **type**: object
      - **DetailedErrorCode**: A detailed error code.
        - **type**: string
      - **DetailedErrorMessage**: A detailed error message.
        - **type**: string

**ListGroupVersions**

Lists the versions of a group.

**URI**: GET /greengrass/groups/GroupID/versions

**Produces**: application/json
CLI:

```bash
aws greengrass list-group-versions \\
--group-id <value> \\
[--next-token <value>] \\
[--max-results <value>] \\
[--cli-input-json <value>] \\
[--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "GroupId": "string",
  "NextToken": "string",
  "MaxResults": "integer"
}
```

Parameters:

**GroupId (p. 571)**

The ID of the Greengrass group.

where used: path; required: true

type: string

**NextToken (p. 573)**

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

where used: query; required: false

type: string

**MaxResults (p. 573)**

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

Responses:

**200 (ListGroupVersionsResponse)**

Success. The response contains the list of versions and metadata for the given group.

**ListVersionsResponse (p. 650)**

```json
{
  "Versions": [
    {
      "Arn": "string",
      "Id": "string",
      "Version": "string",
      "CreationTimestamp": "string"
    }
  ]
}
```
ListVersionsResponse

A list of versions.

type: object

Versions

Information about a version.

type: array

items: VersionInformation (p. 676)

VersionInformation

Information about a version.

type: object

Arn

The ARN of the version.

type: string

Id

The ID of the parent definition that the version is associated with.

type: string

Version

The ID of the version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the version was created.

type: string

NextToken

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

type: string

400

Invalid request.

GeneralError (p. 614)
GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

ListGroups

Retrieves a list of groups.

URI: GET /greengrass/groups

Produces: application/json

CLI:

aws greengrass list-groups \ 
  [--max-results <value>] \ 
  [--next-token <value>] \ 
  [--cli-input-json <value>] \ 
  [--generate-cli-skeleton]

cli-input-json format:

{
  "MaxResults": "integer"},
"NextToken": "string"
}

**Parameters:**

**MaxResults (p. 573)**

The maximum number of results to be returned per request.
where used: query; required: false
type: integer

**NextToken (p. 573)**

The token for the next set of results, or null if there are no more results.
where used: query; required: false
type: string

**Responses:**

**200 (ListGroupsResponse)**

ListGroupsResponse (p. 649)

{
  "Groups": [
    {
      "Name": "string",
      "Id": "string",
      "Arn": "string",
      "LastUpdatedTimestamp": "string",
      "CreationTimestamp": "string",
      "LatestVersion": "string",
      "LatestVersionArn": "string"
    }
  ],
  "NextToken": "string"
}

ListGroupsResponse
type: object

Groups
type: array

items: GroupInformation (p. 640)

GroupInformation
Information about a group.
type: object

Name
The name of the group.
ListLoggerDefinitionVersions

Lists the versions of a logger definition.

URI: GET /greengrass/definition/loggers/LoggerDefinitionId/versions

Produces: application/json

CLI:

```
aws greengrass list-logger-definition-versions \
  --logger-definition-id <value> \
  [--next-token <value>] \
  [--max-results <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{

}
```
"LoggerDefinitionId": "string",
"NextToken": "string",
"MaxResults": "integer"
}

Parameters:

**LoggerDefinitionId (p. 572)**

The ID of the logger definition.

where used: path; required: true

type: string

**NextToken (p. 573)**

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

where used: query; required: false

type: string

**MaxResults (p. 573)**

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

Responses:

**200 (ListLoggerDefinitionVersionsResponse)**

ListVersionsResponse (p. 650)

```json
{
   "Versions": [
      {
         "Arn": "string",
         "Id": "string",
         "Version": "string",
         "CreationTimestamp": "string"
      }
   ],
   "NextToken": "string"
}
```

ListVersionsResponse

A list of versions.

type: object

Versions

Information about a version.

type: array

items: VersionInformation (p. 676)
VersionInformation
  Information about a version.
  type: object
Arn
  The ARN of the version.
  type: string
Id
  The ID of the parent definition that the version is associated with.
  type: string
Version
  The ID of the version.
  type: string
CreationTimestamp
  The time, in milliseconds since the epoch, when the version was created.
  type: string
NextToken
  The token for the next set of results, or null if there are no more results.
  type: string

400
Invalid request.

GeneralError (p. 614)

{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}

GeneralError
  General error information.
  type: object
  required: ["Message"]
Message
  A message that contains information about the error.
  type: string
ErrorDetails
  A list of error details.
ListLoggerDefinitions

Retrieves a list of logger definitions.

URI: GET /greengrass/definition/loggers

Produces: application/json

CLI:

```bash
aws greengrass list-logger-definitions \
   [ --max-results <value> ] \
   [ --next-token <value> ] \
   [ --cli-input-json <value> ] \
   [ --generate-cli-skeleton ]
```

cli-input-json format:

```json
{
   "MaxResults": "integer",
   "NextToken": "string"
}
```

Parameters:

**MaxResults (p. 573)**

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

**NextToken (p. 573)**

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

where used: query; required: false
type: string

Responses:

200 (ListLoggerDefinitionsResponse)

ListDefinitionsResponse (p. 645)

```
{
  "Definitions": [
    {
      "Name": "string",
      "Id": "string",
      "Arn": "string",
      "tags": {
        "additionalProperty0": "string",
        "additionalProperty1": "string",
        "additionalProperty2": "string"
      },
      "LastUpdatedTimestamp": "string",
      "CreationTimestamp": "string",
      "LatestVersion": "string",
      "LatestVersionArn": "string"
    }
  ],
  "NextToken": "string"
}
```

ListDefinitionsResponse

A list of definitions.

Definitions

Information about a definition.

type: array

items: DefinitionInformation (p. 594)

DefinitionInformation

Information about a definition.

type: object

Name

The name of the definition.

type: string

Id

The ID of the definition.

type: string

Arn

The ARN of the definition.

type: string
tags
   The resource tags.
   type: object
   additionalProperties: The key-value pair for the resource tag. Type: string

LastUpdatedTimestamp
   The time, in milliseconds since the epoch, when the definition was last updated.
   type: string

CreationTimestamp
   The time, in milliseconds since the epoch, when the definition was created.
   type: string

LatestVersion
   The ID of the latest version associated with the definition.
   type: string

LatestVersionArn
   The ARN of the latest version associated with the definition.
   type: string

NextToken
   The token for the next set of results, or null if there are no more results.
   type: string

**ListResourceDefinitionVersions**

Lists the versions of a resource definition.

**URL**: GET /greengrass/definition/resources/ResourceDefinitionId/versions

**Produces**: application/json

**CLI**:

```bash
aws greengrass list-resource-definition-versions \ 
  --resource-definition-id <value> \ 
  [--next-token <value>] \ 
  [--max-results <value>] \ 
  [--cli-input-json <value>] \ 
  [--generate-cli-skeleton]
```

**cli-input-json format**:

```json
{
    "ResourceDefinitionId": "string",
    "NextToken": "string",
    "MaxResults": "integer"
}
```
Parameters:

**ResourceDefinitionId (p. 574)**

The ID of the resource definition.

where used: path; required: true

type: string

**NextToken (p. 573)**

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

where used: query; required: false

type: string

**MaxResults (p. 573)**

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

Responses:

**200 (ListResourceDefinitionVersionsResponse)**

ListVersionsResponse (p. 650)

```json
{
   "Versions": [
      {
         "Arn": "string",
         "Id": "string",
         "Version": "string",
         "CreationTimestamp": "string"
      },
      "NextToken": "string"
   }
}
```

ListVersionsResponse

A list of versions.

type: object

**Versions**

Information about a version.

type: array

items: VersionInformation (p. 676)

**VersionInformation**

Information about a version.

type: object
Arn
The ARN of the version.
type: string
Id
The ID of the parent definition that the version is associated with.
type: string
Version
The ID of the version.
type: string
CreationTimestamp
The time, in milliseconds since the epoch, when the version was created.
type: string
NextToken
The token for the next set of results, or null if there are no more results.
type: string

400
Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError
General error information.
type: object
required: ["Message"]
Message
A message that contains information about the error.
type: string
ErrorDetails
A list of error details.
type: array
ListResourceDefinitions

Retrieves a list of resource definitions.

URI: GET /greengrass/definition/resources

Produces: application/json

CLI:

```bash
aws greengrass list-resource-definitions \
    [--max-results <value>] \
    [--next-token <value>] \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
    "MaxResults": "integer",
    "NextToken": "string"
}
```

Parameters:

**MaxResults (p. 573)**

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

**NextToken (p. 573)**

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

where used: query; required: false

type: string
**Responses:**

**200** (ListResourceDefinitionsResponse)

The IDs of all the Greengrass resource definitions in this account.

ListDefinitionsResponse (p. 645)

```json
{
    "Definitions": [  
        {
            "Name": "string",
            "Id": "string",
            "Arn": "string",
            "tags": {
                "additionalProperty0": "string",
                "additionalProperty1": "string",
                "additionalProperty2": "string"
            },
            "LastUpdatedTimestamp": "string",
            "CreationTimestamp": "string",
            "LatestVersion": "string",
            "LatestVersionArn": "string"
        }
    ],
    "NextToken": "string"
}
```

ListDefinitionsResponse

A list of definitions.

- **type**: object

Definitions

Information about a definition.

- **type**: array

items: DefinitionInformation (p. 594)

DefinitionInformation

Information about a definition.

- **type**: object

Name

The name of the definition.

- **type**: string

Id

The ID of the definition.

- **type**: string

Arn

The ARN of the definition.

- **type**: string
tags
The resource tags.
type: object
additionalProperties: The key-value pair for the resource tag. Type: string

LastUpdatedTimestamp
The time, in milliseconds since the epoch, when the definition was last updated.
type: string

CreationTimestamp
The time, in milliseconds since the epoch, when the definition was created.
type: string

LatestVersion
The ID of the latest version associated with the definition.
type: string

LatestVersionArn
The ARN of the latest version associated with the definition.
type: string

NextToken
The token for the next set of results, or null if there are no more results.
type: string

ListSubscriptionDefinitionVersions
Lists the versions of a subscription definition.

URI: GET /greengrass/definition/subscriptions/SubscriptionDefinitionId/versions

Produces: application/json

CLI:

```bash
aws greengrass list-subscription-definition-versions \   --subscription-definition-id <value> \   [--next-token <value>] \   [--max-results <value>] \   [--cli-input-json <value>] \   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "SubscriptionDefinitionId": "string",
   "NextToken": "string",
   "MaxResults": "integer"
}
```
Parameters:

SubscriptionDefinitionId (p. 576)

The ID of the subscription definition.
where used: path; required: true
type: string

NextToken (p. 573)

The token for the next set of results, or null if there are no more results.
where used: query; required: false
type: string

MaxResults (p. 573)

The maximum number of results to be returned per request.
where used: query; required: false
type: integer

Responses:

200 (ListSubscriptionDefinitionVersionsResponse)

ListVersionsResponse (p. 650)

{
   "Versions": [
      {
         "Arn": "string",
         "Id": "string",
         "Version": "string",
         "CreationTimestamp": "string"
      },
      "NextToken": "string"
   ]
}

ListVersionsResponse

A list of versions.
type: object

Versions

Information about a version.
type: array

items: VersionInformation (p. 676)

VersionInformation

Information about a version.
type: object

Arn

The ARN of the version.

type: string

Id

The ID of the parent definition that the version is associated with.

type: string

Version

The ID of the version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the version was created.

type: string

NextToken

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

type: string

400

Invalid request.

GeneralError (p. 614)

{
    "Message": "string",
    "ErrorDetails": [ 
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.
ListSubscriptionDefinitions

Retrieves a list of subscription definitions.

URI: GET /greengrass/definition/subscriptions

Produces: application/json

CLI:

```bash
aws greengrass list-subscription-definitions \
   [--max-results <value>] \
   [--next-token <value>] \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "MaxResults": "integer",
   "NextToken": "string"
}
```

Parameters:

**MaxResults** (p. 573)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

**NextToken** (p. 573)

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

where used: query; required: false
type: string

**Responses:**

200 (ListSubscriptionDefinitionsResponse)

ListDefinitionsResponse (p. 645)

```json
{
  "Definitions": [
    {
      "Name": "string",
      "Id": "string",
      "Arn": "string",
      "tags": {
        "additionalProperty0": "string",
        "additionalProperty1": "string",
        "additionalProperty2": "string"
      },
      "LastUpdatedTimestamp": "string",
      "CreationTimestamp": "string",
      "LatestVersion": "string",
      "LatestVersionArn": "string"
    }
  ],
  "NextToken": "string"
}
```

ListDefinitionsResponse

A list of definitions.

**Definitions**

Information about a definition.

**type**: object

**Definitions**

Information about a definition.

**type**: array

**items**: DefinitionInformation (p. 594)

**DefinitionInformation**

Information about a definition.

**type**: object

**Name**

The name of the definition.

**type**: string

**Id**

The ID of the definition.

**type**: string

**Arn**

The ARN of the definition.

**type**: string
tags
  
  The resource tags.

type: object

  additionalProperties: The key-value pair for the resource tag. Type: string

LastUpdatedTimestamp

  The time, in milliseconds since the epoch, when the definition was last updated.

type: string

CreationTimestamp

  The time, in milliseconds since the epoch, when the definition was created.

type: string

LatestVersion

  The ID of the latest version associated with the definition.

type: string

LatestVersionArn

  The ARN of the latest version associated with the definition.

type: string

NextToken

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

  type: string

ListTagsForResource

Lists tags for a Greengrass resource. Valid resources are Group, ConnectorDefinition, CoreDefinition, DeviceDefinition, FunctionDefinition, LoggerDefinition, ResourceDefinition, SubscriptionDefinition, and BulkDeployment.

URI: GET /tags/resource-arn

Produces: application/json

CLI:

```
aws greengrass list-tags-for-resource  
  [--cli-input-json <value>] 
  [--generate-cli-skeleton]
```

Parameters:

ResourceArn (p. 574)

  The Amazon Resource Name (ARN) of the resource whose tags you want to retrieve.

  where used: path; required: true
type: string

Responses:

200

HTTP Status Code 200: OK.

tags (p. 678)

{
    "tags": {
        "keyName0": "value0",
        "keyName1": "value1",
        "keyName2": "value2"
    }
}

The resource tags.

type: object

additionalProperties: The key-value pair for the resource tag. Type: string

400

Invalid request.

GeneralError (p. 614)

{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)
ErrorDetail
Details about the error.
type: object
DetailedErrorCode
A detailed error code.
type: string
DetailedErrorMessage
A detailed error message.
type: string

ResetDeployments
Resets a group's deployments.

URI: POST /greengrass/groups/GroupId/deployments/$reset

Produces: application/json

CLI:

```bash
aws greengrass reset-deployments \
  --group-id <value> \n  [--force | --no-force] \n  [--amzn-client-token <value>] \n  [--cli-input-json <value>] \n  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "GroupId": "string",
  "Force": "boolean",
  "AmznClientToken": "string"
}
```

Parameters:

GroupId (p. 571)
The ID of the Greengrass group.
where used: path; required: true
type: string

ResetDeploymentsRequestBody (p. 573)
Information required to reset deployments.
where used: body; required: true

{  
}
"Force": true
}

schema:
ResetDeploymentsRequest
  Information about a group reset request.
  type: object
  Force
  If true, performs a best-effort only core reset.
  type: boolean

X-Amzn-Client-Token (p. 579)
  A client token used to correlate requests and responses.
  where used: header; required: false
  type: string

Responses:

200
  Success. The group's deployments were reset.
  ResetDeploymentsResponse (p. 655)

  
  { 
    "DeploymentArn": "string",
    "DeploymentId": "string"
  }

  ResetDeploymentsResponse
    type: object
  DeploymentArn
    The ARN of the deployment.
    type: string
  DeploymentId
    The ID of the deployment.
    type: string

400
  Invalid request.
  GeneralError (p. 614)

  
  { 
    "Message": "string",
    "ErrorDetails": [
    
  }
StartBulkDeployment

Deploys multiple groups in one operation. This action starts the bulk deployment of a specified set of group versions. Each group version deployment is triggered with an adaptive rate that has a fixed upper limit. We recommend that you include an X-Amzn-Client-Token token in every StartBulkDeployment request. These requests are idempotent with respect to the token and the request parameters.

URI: POST /greengrass/bulk/deployments

Produces: application/json

CLI:

```
aws greengrass start-bulk-deployment \
    [--input-file-uri <value>] \
    [--execution-role-arn <value>]
```
StartBulkDeployment

[--tags <value>] 
[--amzn-client-token <value>] 
[--cli-input-json <value>] 
[--generate-cli-skeleton]

cli-input-json format:

```json
{
  "InputFileUri": "string",
  "ExecutionRoleArn": "string",
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  },
  "AmznClientToken": "string"
}
```

Parameters:

StartBulkDeploymentRequestBody (p. 575)

where used: body; required: true

```json
{
  "InputFileUri": "string",
  "ExecutionRoleArn": "string",
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  }
}
```

schema:
StartBulkDeploymentRequest

Information about a bulk deployment. You cannot start a new bulk deployment while another one is still running or in a non-terminal state.

type: object

required: ["InputFileUri", "ExecutionRoleArn"]

InputFileUri

The URI of the input file in the S3 bucket. The execution role must have `getObject` permissions on this bucket to access the input file. The input file is a JSON-serialized, line-delimited file with UTF-8 encoding that provides a list of group and version IDs and the deployment type. This file must be less than 100 MB. Currently, AWS IoT Greengrass supports only `NewDeployment` deployment types.

```json
[{"GroupId":"groupId1", "GroupVersionId":"groupVersionId1", "DeploymentType":"NewDeployment"}
{"GroupId":"groupId2", "GroupVersionId":"groupVersionId2", "DeploymentType":"NewDeployment"}
{"GroupId":"groupId3", "GroupVersionId":"groupVersionId3", "DeploymentType":"NewDeployment"}
...
```
For more information, including the input file format, see Create Bulk Deployments For Groups.

type: string

ExecutionRoleArn

The ARN of the execution role to associate with the bulk deployment operation. This IAM role must allow the greengrass:CreateDeployment action for all group versions that are listed in the input file. This IAM role must have access to the S3 bucket that contains the input file.

type: string
tagss

The resource tags.

type: object

additionalProperties: The key-value pair for the resource tag. Type: string

X-Amzn-Client-Token (p. 579)

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200

Success.

StartBulkDeploymentResponse (p. 672)

`{    "BulkDeploymentId": "string",    "BulkDeploymentArn": "string"  }

StartBulkDeploymentResponse
type: object
BulkDeploymentId

The ID of the bulk deployment.

type: string
BulkDeploymentArn

The ARN of the bulk deployment.

type: string

400

Invalid request.

GeneralError (p. 614)

`{
"Message": "string",
"ErrorDetails": [
{
  "DetailedErrorCode": "string",
  "DetailedErrorMessage": "string"
}
]
}

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

StopBulkDeployment

Stops the execution of a bulk deployment. This action returns a status of Stopping until the deployment is stopped. You cannot start a new bulk deployment while a previous deployment is in the Stopping state. This action doesn't roll back completed deployments or cancel pending deployments.

URI: PUT /greengrass/bulk/deployments/BulkDeploymentId/$stop

Produces: application/json

CLI:

```
aws greengrass stop-bulk-deployment \
  --bulk-deployment-id <value> \
```
StopBulkDeployment

[--cli-input-json <value>] \
[--generate-cli-skeleton]

cli-input-json format:

```
{
   "BulkDeploymentId": "string"
}
```

Parameters:

**BulkDeploymentId** *(p. 545)*

The ID of the bulk deployment.

where used: path; required: true
type: string

Responses:

**200**

Success. The bulk deployment is being stopped.

**400**

Invalid request.

**GeneralError** *(p. 614)*

```
{
   "Message": "string",
   "ErrorDetails": [
   {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
   }
   ]
}
```

GeneralError

General error information.
type: object
required: ["Message"]

**Message**

A message that contains information about the error.
type: string

**ErrorDetails**

A list of error details.
type: array
TagResource

Adds tags to a Greengrass resource. Valid resources are Group, ConnectorDefinition, CoreDefinition, DeviceDefinition, FunctionDefinition, LoggerDefinition, ResourceDefinition, SubscriptionDefinition, and BulkDeployment.

URI: POST /tags/resource-arn

Produces: application/json

CLI:

```
aws greengrass tag-resource \
  --resource-arn <value> \
  [--tags <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "tags": {
    "keyName0": "string",
    "keyName1": "string",
    "keyName2": "string"
  }
}
```

Parameters:

ResourceArn (p. 574)

The Amazon Resource Name (ARN) of the resource to attach the tags to.

where used: path; required: true

  type: string

tags (p. 576)

A map of the key-value pairs for the resource tag.
where used: body; required: true
type: tags

Responses:

204

HTTP Status Code 204: Successful response.

400

Invalid request.

GeneralError (p. 614)

```json
{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}
```

GeneralError

General error information.
type: object
required: ["Message"]

Message

A message that contains information about the error.
type: string

ErrorDetails

A list of error details.
type: array
items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.
type: object

DetailedErrorCode

A detailed error code.
type: string

DetailedErrorMessage

A detailed error message.
type: string
UntagResource

Removes tags from a Greengrass resource. Valid resources are Group, ConnectorDefinition, CoreDefinition, DeviceDefinition, FunctionDefinition, LoggerDefinition, ResourceDefinition, SubscriptionDefinition, and BulkDeployment.

URI: DELETE /tags/resource-arn

Produces: application/json

CLI:

```
aws greengrass untag-resource  
  --resource-arn <value> 
  --tag-keys <value>  
  [--cli-input-json <value>] 
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "ResourceArn": "string",
  "TagKeys": [
    "keyName0"
  ]
}
```

Parameters:

**TagKeys (p. 577)**

An array of tag keys to delete.

where used: query; required: true
type: array of strings

**ResourceArn (p. 574)**

The Amazon Resource Name (ARN) of the resource to remove the tags from.

where used: path; required: true
type: string

Responses:

**204**

HTTP Status Code 204: Successful response.

**400**

Invalid request.

**GeneralError (p. 614)**
"Message": "string",
"ErrorDetails": [
{
   "DetailedErrorCode": "string",
   "DetailedErrorMessage": "string"
}
]

GeneralError

General error information.
type: object
required: ["Message"]

Message

A message that contains information about the error.
type: string

ErrorDetails

A list of error details.
type: array
items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.
type: object

DetailedErrorCode

A detailed error code.
type: string

DetailedErrorMessage

A detailed error message.
type: string

**UpdateConnectivityInfo**

Updates the connectivity information for the core. Devices that belong to the group that has this core receive this information to find the location of the core and connect to it.

URI: PUT /greengrass/things/ThingName/connectivityInfo

Produces: application/json

**CLI:**

```
aws greengrass update-connectivity-info 
   --thing-name <value> 
   [--connectivity-info <value>] 
   [--cli-input-json <value>] 
```
cli-input-json format:

```json
{
    "ThingName": "string",
    "ConnectivityInfo": [
        {
            "Id": "string",
            "HostAddress": "string",
            "PortNumber": "int32-integer",
            "Metadata": "string"
        }
    ]
}
```

**Parameters:**

**ThingName (p. 577)**

The thing name.

- where used: path; required: true
- type: string

**UpdateConnectivityInfoRequestBody (p. 577)**

The connectivity information.

- where used: body; required:

```json
{
    "ConnectivityInfo": [
        {
            "Id": "string",
            "HostAddress": "string",
            "PortNumber": 0x01,
            "Metadata": "string"
        }
    ]
}
```

**ConnectivityInfo**

A list of connectivity info.

- type: array
- items: ConnectivityInfo (p. 588)

Information about a Greengrass core's connectivity.

- required: ["Id", "HostAddress"]
- Id

The ID of the connectivity information.
**UpdateConnectivityInfo**

**type**: string

**HostAddress**

The endpoint for the Greengrass core. Can be an IP address or DNS address.

**type**: string

**PortNumber**

The port of the Greengrass core, usually 8883.

**type**: integer

**format**: int32

**Metadata**

Metadata for this endpoint.

**type**: string

### Responses:

**200**

Success.

**UpdateConnectivityInfoResponse (p. 675)**

```json
{
    "message": "string",
    "Version": "string"
}
```

**message**

A message about the connectivity info update request.

**type**: string

**Version**

The new version of the connectivity info.

**type**: string

**400**

Invalid request.

**GeneralError (p. 614)**

```json
{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}
```
GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

500

Server error.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError

General error information.

  type: object

required: ["Message"]

Message

A message that contains information about the error.
UpdateConnectorDefinition

Updates the name of a connector definition. To update the list of available connectors, use CreateConnectorDefinitionVersion.

**URI:** PUT /greengrass/definition/connectors/ConnectorDefinitionId

**Produces:** application/json

**CLI:**

```bash
aws greengrass update-connector-definition \
  --connector-definition-id <value> \
  [--name <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

**cli-input-json format:**

```json
{
  "ConnectorDefinitionId": "string",
  "Name": "string"
}
```

**Parameters:**

**ConnectorDefinitionId** *(p. 546)*

The ID of the connector definition.

where used: path; required: true
UpdateConnectorDefinition

UpdateDefinitionRequestBody (p. 578)

where used: body; required: true

```
{
  "Name": "string"
}
```

Name

The name of the definition.

required: true

type: string

Responses:

200

Success.

Empty (p. 600)

```
{
}
```

Empty Schema

Empty

type: object

400

Invalid request.

GeneralError (p. 614)

```
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]
UpdateCoreDefinition

Updates the name of a core definition. To update the list of available cores, use CreateCoreDefinitionVersion.

URI: PUT /greengrass/definition/cores/CoreDefinitionId

Produces: application/json

CLI:

```
aws greengrass update-core-definition \n  --core-definition-id <value> \n  [--name <value>] \n  [--cli-input-json <value>] \n  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "CoreDefinitionId": "string",
  "Name": "string"
}
```

Parameters:

CoreDefinitionId (p. 547)

The ID of the core definition.
where used: path; required: true
type: string

**UpdateDefinitionRequestBody (p. 578)**
where used: body; required: true

```
{
   "Name": "string"
}
```
schema:
Name
The name of the definition.
required: true
type: string

**Responses:**

**200**
Success.

**Empty (p. 600)**

```
{
}
```

Empty Schema
Empty
type: object

**400**
Invalid request.

**GeneralError (p. 614)**

```
{
   "Message": "string",
   "ErrorDetails": [
      {
         "DetailedErrorCode": "string",
         "DetailedErrorMessage": "string"
      }
   ]
}
```

GeneralError
General error information.
type: object
required: ["Message"]
UpdateDeviceDefinition

Updates the name of a device definition. To update the list of available devices, use CreateDeviceDefinitionVersion.

URI: PUT /greengrass/definition/devices/DeviceDefinitionId

Produces: application/json

CLI:

```
aws greengrass update-device-definition \
  --device-definition-id <value> \
  [--name <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "DeviceDefinitionId": "string",
  "Name": "string"
}
```

Parameters:

**DeviceDefinitionId** *(p. 570)*

The ID of the device definition.
where used: path; required: true

type: string

**UpdateDefinitionRequestBody (p. 578)**

where used: body; required: true

```json
{
   "Name": "string"
}
```

**Responses:**

**200**

Success.

**Empty (p. 600)**

```json
{}
```

**Empty Schema**

Empty

type: object

**400**

Invalid request.

**GeneralError (p. 614)**

```json
{
   "Message": "string",
   "ErrorDetails": [
   {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
   }
   ]
}
```

**GeneralError**

General error information.

type: object

required: ["Message"]
UpdateFunctionDefinition

Updates the name of a function definition. To update the list of available functions, use CreateFunctionDefinitionVersion.

**URI:** PUT /greengrass/definition/functions/\*FunctionDefinitionId*

**Produces:** application/json

**CLI:**

```bash
aws greengrass update-function-definition \
   --function-definition-id <value> \
   [--name <value>] \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "FunctionDefinitionId": "string",
   "Name": "string"
}
```

**Parameters:**

*FunctionDefinitionId* (p. 571)

The ID of the function definition.
where used: path; required: true

type: string

`UpdateDefinitionRequestBody (p. 578)`

where used: body; required: true

```json
{
   "Name": "string"
}
```

schema:
Name

The name of the definition.

required: true

type: string

Responses:

200

Success.

Empty (p. 600)

```json
{
}
```

Empty Schema

Empty
type: object

400

Invalid request.

`GeneralError (p. 614)`

```json
{
   "Message": "string",
   "ErrorDetails": [
   {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
   }
   ]
}
```

GeneralError

General error information.

type: object
UpdateGroup

Updates the name of a group. To update group components, use CreateGroupVersion.

URI: PUT /greengrass/groups/{GroupId}

Produces: application/json

CLI:

```bash
aws greengrass update-group \
  --group-id <value> \
  [--name <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "GroupId": "string",
  "Name": "string"
}
```

Parameters:

**GroupId** *(p. 571)*

The ID of the Greengrass group.

---

required: ["Message"]

**Message**

A message that contains information about the error.

type: string

**ErrorDetails**

A list of error details.

type: array

items: [ErrorDetail](p. 600)

**ErrorDetail**

Details about the error.

type: object

**DetailedErrorCode**

A detailed error code.

type: string

**DetailedErrorMessage**

A detailed error message.

type: string
where used: path; required: true
type: string

**UpdateDefinitionRequestBody** (p. 578)

where used: body; required: true

```
{
  "Name": "string"
}
```

schema:
Name
  The name of the definition.
  required: true
  type: string

**Responses:**

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Success.</td>
</tr>
<tr>
<td></td>
<td>Empty (p. 600)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td></td>
<td>Empty Schema</td>
</tr>
<tr>
<td></td>
<td>Empty</td>
</tr>
<tr>
<td></td>
<td>type: object</td>
</tr>
<tr>
<td>400</td>
<td>Invalid request.</td>
</tr>
<tr>
<td></td>
<td>GeneralError (p. 614)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;Message&quot;: &quot;string&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;ErrorDetails&quot;: [</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;DetailedErrorCode&quot;: &quot;string&quot;,</td>
</tr>
</tbody>
</table>
|             |       "DetailedErrorMessage": "string"
|             |     ]                      |
|             | }                          |
|             | GeneralError                |
|             | General error information.  |
|             | type: object                 |
required: ["Message"]
Message
A message that contains information about the error.
  type: string
ErrorDetails
A list of error details.
  type: array
  items: ErrorDetail (p. 600)
ErrorDetail
Details about the error.
  type: object
DetailedErrorCode
A detailed error code.
  type: string
DetailedErrorMessage
A detailed error message.
  type: string

**UpdateGroupCertificateConfiguration**

Updates the certificate expiry time for a group.

**URI:** PUT /greengrass/groups/\*GroupId*/*certificateauthorities/configuration/expiry

**Produces:** application/json

**CLI:**

```bash
aws greengrass update-group-certificate-configuration \
  --group-id <value> \
  [--certificate-expiry-in-milliseconds <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "GroupId": "string",
  "CertificateExpIryInMilliseconds": "string"
}
```

**Parameters:**

**GroupId (p. 571)**

The ID of the Greengrass group.
where used: path; required: true

type: string

**UpdateGroupCertificateConfigurationRequestBody** *(p. 579)*

where used: body; required: true

```json
{
   "CertificateExpiryInMilliseconds": "string"
}
```

schema:
updateGroupCertificateConfigurationRequest
type: object
required: ["CertificateExpiryInMilliseconds"]

**CertificateExpiryInMilliseconds**
The amount of time, in milliseconds, before the certificate expires.
type: string

**Responses:**

**200**

Success. The response body contains the PKI configuration.

**GroupCertificateConfiguration** *(p. 639)*

```json
{
   "GroupId": "string",
   "CertificateAuthorityExpiryInMilliseconds": "string",
   "CertificateExpiryInMilliseconds": "string"
}
```

**GroupCertificateConfiguration**
Information about a group certificate configuration.
type: object

**GroupId**
The ID of the group certificate configuration.
type: string

**CertificateAuthorityExpiryInMilliseconds**
The amount of time, in milliseconds, before the certificate authority expires.
type: string

**CertificateExpiryInMilliseconds**
The amount of time, in milliseconds, before the certificate expires.
type: string
400
Invalid request.

GeneralError (p. 614)

```json
{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}
```

**GeneralError**

General error information.

type: object

required: [*Message*]

**Message**

A message that contains information about the error.

type: string

**ErrorDetails**

A list of error details.

**ErrorDetail**

Details about the error.

**DetailedErrorCode**

A detailed error code.

type: string

**DetailedErrorMessage**

A detailed error message.

type: string

500
Server error.

GeneralError (p. 614)

```json
{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}
```
UpdateLoggerDefinition

Updates the name of a logger definition. To update the list of available loggers, use CreateLoggerDefinitionVersion.

URI: PUT /greengrass/definition/loggers/LoggerDefinitionId

Produces: application/json

CLI:

```
aws greengrass update-logger-definition \     
  --logger-definition-id <value> \     
  [--name <value>] \     
  [--cli-input-json <value>] \     
  [--generate-cli-skeleton]
```
cli-input-json format:

```json
{
  "LoggerDefinitionId": "string",
  "Name": "string"
}
```

**Parameters:**

**LoggerDefinitionId** *(p. 572)*

The ID of the logger definition.

where used: path; required: true
type: string

**UpdateDefinitionRequestBody** *(p. 578)*

where used: body; required: true

```json
{
  "Name": "string"
}
```

schema:
Name

The name of the definition.

required: true
type: string

**Responses:**

**200**

Success.

**Empty (p. 600)**

```json
{}
```

Empty Schema

Empty
type: object

**400**

Invalid request.

**GeneralError (p. 614)**

```json
{
  "Message": "string",
```
"ErrorDetails": [  {
    "DetailedErrorCode": "string",
    "DetailedErrorMessage": "string"
  }
]

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

UpdateResourceDefinition

Updates the name of a resource definition. To update the list of available resources, use CreateResourceDefinitionVersion.

URI: PUT /greengrass/definition/resources/ResourceDefinitionId

Produces: application/json

CLI:

```bash
aws greengrass update-resource-definition \
  --resource-definition-id <value> \n  [--name <value>] \n  [--cli-input-json <value>]
```
cli-input-json format:

```json
{
  "ResourceDefinitionId": "string",
  "Name": "string"
}
```

**Parameters:**

**ResourceId** *(p. 574)*

The ID of the resource definition.

where used: path; required: true
type: string

**UpdateDefinitionRequestBody** *(p. 578)*

where used: body; required: true

```json
{
  "Name": "string"
}
```

schema:
Name

The name of the definition.

required: true
type: string

**Responses:**

200

Success.

Empty *(p. 600)*

```json
{}
```

Empty Schema
Empty
type: object

400

Invalid request.

GeneralError *(p. 614)*
UpdateSubscriptionDefinition

Updates the name of a subscription definition. To update the list of available subscriptions, use CreateSubscriptionDefinitionVersion.

URI: PUT /greengrass/definition/subscriptions/SubscriptionDefinitionId

Produces: application/json

CLI:

aws greengrass update-subscription-definition \
UpdateSubscriptionDefinition

```bash
--subscription-definition-id <value> \
[--name <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "SubscriptionDefinitionId": "string",
   "Name": "string"
}
```

**Parameters:**

**SubscriptionDefinitionId (p. 576)**

The ID of the subscription definition.

- where used: path; required: true
- type: string

**UpdateDefinitionRequestBody (p. 578)**

- where used: body; required: true

```json
{
   "Name": "string"
}
```

**Responses:**

**200**

Success.

**Empty (p. 600)**

```json
{}
```

**Empty Schema**

- Empty
- type: object

**400**

Invalid request.
Endpoints

The following endpoints are supported:

- /greengrass/bulk/deployments (p. 271)
- /greengrass/bulk/deployments/BulkDeploymentId/$stop (p. 276)
- /greengrass/bulk/deployments/BulkDeploymentId/detailed-reports (p. 278)
- /greengrass/bulk/deployments/BulkDeploymentId/status (p. 282)
- /greengrass/definition/connectors (p. 285)
• /greengrass/definition/connectors/ConnectorDefinitionId (p. 291)
• /greengrass/definition/connectors/ConnectorDefinitionId/versions (p. 298)
• /greengrass/definition/connectors/ConnectorDefinitionId/versions/ConnectorDefinitionVersionId (p. 304)
• /greengrass/definition/cores (p. 308)
• /greengrass/definition/cores/CoreDefinitionId (p. 314)
• /greengrass/definition/cores/CoreDefinitionId/versions (p. 320)
• /greengrass/definition/cores/CoreDefinitionId/versions/CoreDefinitionVersionId (p. 327)
• /greengrass/definition/devices (p. 330)
• /greengrass/definition/devices/DeviceDefinitionId (p. 336)
• /greengrass/definition/devices/DeviceDefinitionId/versions (p. 343)
• /greengrass/definition/devices/DeviceDefinitionId/versions/DeviceDefinitionVersionId (p. 349)
• /greengrass/definition/functions (p. 353)
• /greengrass/definition/functions/FunctionDefinitionId (p. 362)
• /greengrass/definition/functions/FunctionDefinitionId/versions (p. 369)
• /greengrass/definition/functions/FunctionDefinitionId/versions/FunctionDefinitionVersionId (p. 378)
• /greengrass/definition/loggers (p. 384)
• /greengrass/definition/loggers/LoggerDefinitionId (p. 391)
• /greengrass/definition/loggers/LoggerDefinitionId/versions (p. 397)
• /greengrass/definition/loggers/LoggerDefinitionId/versions/LoggerDefinitionVersionId (p. 404)
• /greengrass/definition/resources (p. 407)
• /greengrass/definition/resources/ResourceDefinitionId (p. 417)
• /greengrass/definition/resources/ResourceDefinitionId/versions (p. 424)
• /greengrass/definition/resources/ResourceDefinitionId/versions/ResourceDefinitionVersionId (p. 434)
• /greengrass/definition/subscriptions (p. 441)
• /greengrass/definition/subscriptions/SubscriptionDefinitionId (p. 447)
• /greengrass/definition/subscriptions/SubscriptionDefinitionId/versions (p. 454)
• /greengrass/definition/subscriptions/SubscriptionDefinitionId/versions/SubscriptionDefinitionVersionId (p. 460)
• /greengrass/groups (p. 464)
• /greengrass/groups/GroupId (p. 470)
• /greengrass/groups/GroupId/certificateauthorities (p. 476)
• /greengrass/groups/GroupId/certificateauthorities/configuration/expiry (p. 482)
• /greengrass/groups/GroupId/certificateauthorities/CertificateAuthorityId (p. 488)
• /greengrass/groups/GroupId/deployments (p. 491)
• /greengrass/groups/GroupId/deployments/$reset (p. 497)
• /greengrass/groups/GroupId/deployments/DeploymentId/status (p. 500)
• /greengrass/groups/GroupId/role (p. 503)
• /greengrass/groups/GroupId/versions (p. 511)
• /greengrass/groups/GroupId/versions/GroupVersionId (p. 518)
• /greengrass/servicerole (p. 521)
• /greengrass/things/ThingName/connectivityInfo (p. 527)
• /greengrass/updates (p. 534)
• /tags/resource-arn (p. 538)
/greengrass/bulk/deployments

POST

POST /greengrass/bulk/deployments

Operation ID: StartBulkDeployment (p. 238)

Deploys multiple groups in one operation. This action starts the bulk deployment of a specified set of group versions. Each group version deployment is triggered with an adaptive rate that has a fixed upper limit. We recommend that you include an X-Amzn-Client-Token token in every StartBulkDeployment request. These requests are idempotent with respect to the token and the request parameters.

Produces: application/json

Body Parameters

StartBulkDeploymentRequestBody (p. 575)

where used: body; required: true

```
{
  "InputFileUri": "string",
  "ExecutionRoleArn": "string",
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  }
}
```

StartBulkDeploymentRequest

Information about a bulk deployment. You cannot start a new bulk deployment while another one is still running or in a non-terminal state.

type: object

required: ["InputFileUri", "ExecutionRoleArn"]

InputFileUri

The URI of the input file in the S3 bucket. The execution role must have get object permissions on this bucket to access the input file. The input file is a JSON-serialized, line-delimited file with UTF-8 encoding that provides a list of group and version IDs and the deployment type. This file must be less than 100 MB. Currently, AWS IoT Greengrass supports only NewDeployment deployment types.

```
{"GroupId":"groupId1", "GroupVersionId":"groupVersionId1", "DeploymentType":"NewDeployment"}
{"GroupId":"groupId2", "GroupVersionId":"groupVersionId2", "DeploymentType":"NewDeployment"}
{"GroupId":"groupId3", "GroupVersionId":"groupVersionId3", "DeploymentType":"NewDeployment"}
...
```

For more information, including the input file format, see Create Bulk Deployments For Groups.
type: string

ExecutionRoleArn
The ARN of the execution role to associate with the bulk deployment operation. This IAM role must allow the greengrass:CreateDeployment action for all group versions that are listed in the input file. This IAM role must have access to the S3 bucket that contains the input file.

type: string
tags
The resource tags.
type: object
additionalProperties: The key-value pair for the resource tag. Type: string

Header Parameters

X-Amzn-Client-Token (p. 579)
A client token used to correlate requests and responses.
where used: header; required: false
type: string

CLI

```
aws greengrass start-bulk-deployment \
  [--input-file-uri <value>] \
  [--execution-role-arn <value>] \
  [--tags <value>] \
  [--amzn-client-token <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "InputFileUri": "string",
  "ExecutionRoleArn": "string",
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  },
  "AmznClientToken": "string"
}
```

Responses

200
Success.

StartBulkDeploymentResponse (p. 672)

```
StartBulkDeploymentResponse

  type: object

BulkDeploymentId

  The ID of the bulk deployment.

  type: string

BulkDeploymentArn

  The ARN of the bulk deployment.

  type: string

400

Invalid request.

GeneralError (p. 614)

  
  {
    "Message": "string",
    "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]

GeneralError

  General error information.

  type: object

  required: ["Message"]

Message

  A message that contains information about the error.

  type: string

ErrorDetails

  A list of error details.

  type: array

  items: ErrorDetail (p. 600)

ErrorDetail

  Details about the error.

  type: object

DetailedErrorCode

  A detailed error code.
GET /greengrass/bulk/deployments

Operation ID: ListBulkDeployments (p. 186)
Returns a list of bulk deployments.
Produces: application/json

Query Parameters

MaxResults (p. 573)
The maximum number of results to be returned per request.
where used: query; required: false
type: integer

NextToken (p. 573)
The token for the next set of results, or null if there are no more results.
where used: query; required: false
type: string

CLI

```bash
aws greengrass list-bulk-deployments \
   [--max-results <value>] \
   [--next-token <value>] \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "MaxResults": "integer",
   "NextToken": "string"
}
```

Responses

200
Success. The response body contains the list of bulk deployments.
ListBulkDeploymentsResponse (p. 644)
ListBulkDeploymentsResponse

type: object

BulkDeployments

type: array

items: BulkDeployment (p. 583)

BulkDeployment

Information about a bulk deployment. You cannot start a new bulk deployment while another one is still running or in a non-terminal state.

type: object

BulkDeploymentId

The ID of the bulk deployment.

type: string

BulkDeploymentArn

The ARN of the bulk deployment.

type: string

CreatedAt

The time, in ISO format, when the deployment was created.

type: string

NextToken

The token for the next set of results, or null if there are no more results.
in: query

type: string

400

Invalid request.

GeneralError (p. 614)

{}
/greengrass/bulk/deployments/BulkDeploymentId/$stop

PUT

PUT /greengrass/bulk/deployments/BulkDeploymentId/$stop

Operation ID: StopBulkDeployment (p. 241)

Stops the execution of a bulk deployment. This action returns a status of Stopping until the deployment is stopped. You cannot start a new bulk deployment while a previous deployment is in the Stopping state. This action doesn't roll back completed deployments or cancel pending deployments.

Produces: application/json

Path Parameters

BulkDeploymentId (p. 545)

The ID of the bulk deployment.
where used: path; required: true

type: string

**CLI**

```
aws greengrass stop-bulk-deployment
   --bulk-deployment-id <value>
   [--cli-input-json <value>]
   [--generate-cli-skeleton]
```

cli-input-json format:

```
{
   "BulkDeploymentId": "string"
}
```

**Responses**

**200**

Success. The bulk deployment is being stopped.

**400**

Invalid request.

**GeneralError (p. 614)**

```
{
   "Message": "string",
   "ErrorDetails": [
   {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
   }
   ]
}
```

GeneralError

General error information.

- **type**: object
- **required**: ["Message"]

Message

A message that contains information about the error.

- **type**: string

ErrorDetails

A list of error details.

- **type**: array
  - **items**: ErrorDetail (p. 600)
ErrorDetail
    Details about the error.
    type: object
DetailedErrorCode
    A detailed error code.
    type: string
DetailedErrorMessage
    A detailed error message.
    type: string

/greengrass/bulk/deployments/BulkDeploymentId/detailed-reports

GET

GET /greengrass/bulk/deployments/BulkDeploymentId/detailed-reports

Operation ID: ListBulkDeploymentDetailedReports (p. 182)

Gets a paginated list of the deployments that have been started in a bulk deployment operation, and their current deployment status.

Produces: application/json

Path Parameters

BulkDeploymentId (p. 545)
    The ID of the bulk deployment.
    where used: path; required: true
    type: string

Query Parameters

MaxResults (p. 573)
    The maximum number of results to be returned per request.
    where used: query; required: false
    type: integer

NextToken (p. 573)
    The token for the next set of results, or null if there are no more results.
    where used: query; required: false
    type: string
CLI

```
aws greengrass list-bulk-deployment-detailed-reports \
  --bulk-deployment-id <value> \
  [--max-results <value>] \
  [--next-token <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "BulkDeploymentId": "string",
  "MaxResults": "integer",
  "NextToken": "string"
}
```

Responses

200

Success. The response body contains the list of deployments for the given group.

`ListBulkDeploymentDetailedReportsResponse` (p. 642)

```
{
  "Deployments": [ 
    {
      "DeploymentId": "string",
      "DeploymentArn": "string",
      "GroupArn": "string",
      "DeploymentType": "NewDeployment|Redeployment|ResetDeployment|ForceResetDeployment",
      "CreatedAt": "string",
      "DeploymentStatus": "string",
      "ErrorMessage": "string",
      "ErrorDetails": [ 
        {
          "DetailedErrorCode": "string",
          "DetailedErrorMessage": "string"
        }
      ]
    }
  ],
  "NextToken": "string"
}
```

`ListBulkDeploymentDetailedReportsResponse`

type: object

`Deployments`

type: array

`items`: `BulkDeploymentResult` (p. 584)

`BulkDeploymentResult`

Information about an individual group deployment in a bulk deployment operation.

type: object
DeploymentId
The ID of the group deployment.
type: string

DeploymentArn
The ARN of the group deployment.
type: string

GroupArn
The ARN of the Greengrass group.
type: string

DeploymentType
The type of deployment. When used for CreateDeployment, only NewDeployment and Redeployment are valid.
type: string
elem: string
enum: ["NewDeployment", "Redeployment", "ResetDeployment", "ForceResetDeployment"]

CreatedAt
The time, in ISO format, when the deployment was created.
type: string

DeploymentStatus
The current status of the group deployment: `Building`, `InProgress`, `Success`, or `Failure`.
type: string

ErrorMessage
The error message for a failed deployment.
type: string

ErrorDetails
A list of error details.
type: array
items: ErrorDetail (p. 600)

ErrorDetail
Details about the error.
type: object

DetailedErrorCode
A detailed error code.
type: string

DetailedErrorMessage
A detailed error message.
NextToken

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

in: query
type: string

400

Invalid request.

**GeneralError (p. 614)**

```json
{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}
```

**GeneralError**

General error information.

- **Message**
  - type: string
  - A message that contains information about the error.

- **ErrorDetails**
  - A list of error details.
    - type: array
    - items: **ErrorDetail (p. 600)**

**ErrorDetail**

Details about the error.

- **DetailedErrorCode**
  - type: string
  - A detailed error code.

- **DetailedErrorMessage**
  - type: string
  - A detailed error message.
/greengrass/bulk/deployments/BulkDeploymentId/status

GET

GET /greengrass/bulk/deployments/BulkDeploymentId/status

Operation ID: GetBulkDeploymentStatus (p. 112)

Returns the status of a bulk deployment.

Produces: application/json

Path Parameters

BulkDeploymentId (p. 545)

The ID of the bulk deployment.

where used: path; required: true

type: string

CLI

```
aws greengrass get-bulk-deployment-status \
  --bulk-deployment-id <value>  \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "BulkDeploymentId": "string"
}
```

Responses

200

Success. The response body contains the status of the bulk deployment.

GetBulkDeploymentStatusResponse (p. 616)

```json
{
  "BulkDeploymentStatus": ["Initializing","Running","Completed","Stopping","Stopped","Failed"],
  "BulkDeploymentMetrics": {
    "RecordsProcessed": 0,
    "InvalidInputRecords": 0,
    "RetryAttempts": 0
  },
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  },
  "CreatedAt": "string",
```
GetBulkDeploymentStatusResponse

Information about the status of a bulk deployment at the time of the request.

type: object

BulkDeploymentStatus

The current status of the bulk deployment.

type: string


BulkDeploymentMetrics

Relevant metrics on input records processed during bulk deployment.

type: object

RecordsProcessed

The total number of group records from the input file that have been processed or attempted so far.

type: integer

InvalidInputRecords

The total number of records that returned a non-retryable error. For example, this can occur if a group record from the input file uses an invalid format or specifies a nonexistent group version, or if the execution role doesn't grant permission to deploy a group or group version.

type: integer

RetryAttempts

The total number of deployment attempts that returned a retryable error. For example, a retry is triggered if the attempt to deploy a group returns a throttling error. StartBulkDeployment retries a group deployment up to five times.

type: integer

tags

The resource tags.

type: object

additionalProperties: The key-value pair for the resource tag. Type: string

CreatedAt

The time, in ISO format, when the deployment was created.

type: string

ErrorMessage

Error message
type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

400

Invalid request.

GeneralError (p. 614)

{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)
ErrorDetail
  Details about the error.
  type: object
DetailedErrorCode
  A detailed error code.
  type: string
DetailedErrorMessage
  A detailed error message.
  type: string

/greengrass/definition/connectors

GET

GET /greengrass/definition/connectors
Operation ID: ListConnectorDefinitions (p. 191)
Retrieves a list of connector definitions.
Produces: application/json
Query Parameters
MaxResults (p. 573)
  The maximum number of results to be returned per request.
  where used: query; required: false
  type: integer
NextToken (p. 573)
  The token for the next set of results, or null if there are no more results.
  where used: query; required: false
  type: string

CLI

aws greengrass list-connector-definitions \
  [--max-results <value>] \
  [--next-token <value>] \
  [--cli-input-json <value>] \n  [--generate-cli-skeleton]

cli-input-json format:

```json
{
  "MaxResults": "integer",
```
Responses

200 (ListConnectorDefinitionsResponse)

ListDefinitionsResponse (p. 645)

```json
{
    "Definitions": [
    {
        "Name": "string",
        "Id": "string",
        "Arn": "string",
        "tags": {
            "additionalProperty0": "string",
            "additionalProperty1": "string",
            "additionalProperty2": "string"
        },
        "LastUpdatedTimestamp": "string",
        "CreationTimestamp": "string",
        "LatestVersion": "string",
        "LatestVersionArn": "string"
    }
    ],
    "NextToken": "string"
}
```

ListDefinitionsResponse

A list of definitions.

`type: object`

Definitions

Information about a definition.

`type: array`

`items: DefinitionInformation (p. 594)`

DefinitionInformation

Information about a definition.

`type: object`

Name

The name of the definition.

`type: string`

Id

The ID of the definition.

`type: string`

Arn

The ARN of the definition.

`type: string`
tags
The resource tags.
type: object
additionalProperties: The key-value pair for the resource tag. Type: string

LastUpdatedTimestamp
The time, in milliseconds since the epoch, when the definition was last updated.
type: string

CreationTimestamp
The time, in milliseconds since the epoch, when the definition was created.
type: string

LatestVersion
The ID of the latest version associated with the definition.
type: string

LatestVersionArn
The ARN of the latest version associated with the definition.
type: string

NextToken
The token for the next set of results, or null if there are no more results.
type: string

POST
POST /greengrass/definition/connectors

Operation ID: CreateConnectorDefinition (p. 9)

Creates a connector definition. You can provide the initial version of the connector definition now or use CreateConnectorDefinitionVersion later.

Produces: application/json

Body Parameters
CreateConnectorDefinitionRequestBody (p. 547)

where used: body; required: true

```json
{
    "Name": "string",
    "InitialVersion": {
        "Connectors": [
            {
                "Id": "string",
                "ConnectorArn": "string",
                "Parameters": {
                    "additionalProperty0": "string",
                    "additionalProperty1": "string",
```
Name
The name of the connector definition.
type: string

InitialVersion
Information about the connector definition version, which is a container for connectors.
type: object

Connectors
A list of references to connectors in this version, with their corresponding configuration settings.
type: array
items: Connector (p. 588)

Connector
Information about a connector. Connectors run on the Greengrass core and contain built-in integration with local infrastructure, device protocols, AWS, and other cloud services.
type: object
required: ["Id", "ConnectorArn"]

Id
A descriptive or arbitrary ID for the connector. This value must be unique within the connector definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-]+.
type: string

ConnectorArn
The ARN of the connector.
type: string

Parameters
The parameters or configuration used by the connector.
type: object
additionalProperties: An object with properties of type string that represent the connector configuration.

tags
The resource tags.
type: object
additionalProperties: The key-value pair for the resource tag. Type: string

Header Parameters

X-Amzn-Client-Token (p. 579)

A client token used to enforce the idempotency of this API.

where used: header; required: false
type: string

CLI

```bash
aws greengrass create-connector-definition
   [--name <value>] 
   [--initial-version <value>] 
   [--tags <value>] 
   [--amzn-client-token <value>] 
   [--cli-input-json <value>] 
   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "Name": "string",
   "InitialVersion": {
      "Connectors": [
         {
            "Id": "string",
            "ConnectorArn": "string",
            "Parameters": {
               "additionalProperty0": "string",
               "additionalProperty1": "string",
               "additionalProperty2": "string"
            }
         }
      ],
   "tags": {
      "additionalProperty0": "string",
      "additionalProperty1": "string",
      "additionalProperty2": "string"
   },
   "AmznClientToken": "string"
}
```

Responses

200 (CreateConnectorDefinitionResponse)

DefinitionInformation (p. 594)

```json
{
   "Name": "string",
   "Id": "string",
   "Arn": "string",
   "LastUpdatedTimestamp": "string",
   "CreationTimestamp": "string",
   "LastUpdatedVersion": "string",
   "InitialVersion": "string"
}
```
DefinitionInformation

Information about a definition.

type: object

Name

The name of the definition.

type: string

Id

The ID of the definition.

type: string

Arn

The ARN of the definition.

type: string

LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.

type: string

LatestVersion

The ID of the latest version associated with the definition.

type: string

LatestVersionArn

The ARN of the latest version associated with the definition.

type: string

400

Invalid request.

GeneralError (p. 614)
GeneralError
  General error information.
  type: object
  required: ["Message"]
Message
  A message that contains information about the error.
  type: string
ErrorDetails
  A list of error details.
  type: array
  items: ErrorDetail (p. 600)
ErrorDetail
  Details about the error.
  type: object
DetailedErrorCode
  A detailed error code.
  type: string
DetailedErrorMessage
  A detailed error message.
  type: string

/getreengrass/definition/connectors/ConnectorDefinitionId

GET

GET /getreengrass/definition/connectors/ConnectorDefinitionId

Operation ID: GetConnectorDefinition (p. 119)

Retrieves information about a connector definition.

Produces: application/json

Path Parameters

ConnectorDefinitionId (p. 546)
  The ID of the connector definition.
  where used: path; required: true
  type: string
CLI

```bash
aws greengrass get-connector-definition
   --connector-definition-id <value> \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "ConnectorDefinitionId": "string"
}
```

Responses

**200 (GetConnectorDefinitionResponse)**

DefinitionInformation (p. 594)

```json
{
   "Name": "string",
   "Id": "string",
   "Arn": "string",
   "tags": {
      "additionalProperty0": "string",
      "additionalProperty1": "string",
      "additionalProperty2": "string"
   },
   "LastUpdatedTimestamp": "string",
   "CreationTimestamp": "string",
   "LatestVersion": "string",
   "LatestVersionArn": "string"
}
```

DefinitionInformation

- Information about a definition.

  type: object

Name

- The name of the definition.

  type: string

Id

- The ID of the definition.

  type: string

Arn

- The ARN of the definition.

  type: string

tags

- The resource tags.

  type: object
additionalProperties: The key-value pair for the resource tag. Type: string

**LastUpdatedTimestamp**

The time, in milliseconds since the epoch, when the definition was last updated.

type: string

**CreationTimestamp**

The time, in milliseconds since the epoch, when the definition was created.

type: string

**LatestVersion**

The ID of the latest version associated with the definition.

type: string

**LatestVersionArn**

The ARN of the latest version associated with the definition.

type: string

400

Invalid request.

**GeneralError (p. 614)**

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

**GeneralError**

General error information.

```json
type: object
required: ["Message"]
```

**Message**

A message that contains information about the error.

```json
type: string
```

**ErrorDetails**

A list of error details.

```json
type: array
items: ErrorDetail (p. 600)
```

**ErrorDetail**

Details about the error.
DELETE

DELETE /greengrass/definition/connectors/ConnectorDefinitionId

Operation ID: DeleteConnectorDefinition (p. 92)

Deletes a connector definition.

Produces: application/json

Path Parameters

ConnectorDefinitionId (p. 546)

The ID of the connector definition.

where used: path; required: true

type: string

CLI

aws greengrass delete-connector-definition \
   --connector-definition-id <value> \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]

cli-input-json format:

{
   "ConnectorDefinitionId": "string"
}

Responses

200

Success.

Empty (p. 600)
Empty Schema

Empty
type: object

400

Invalid request.

GeneralError (p. 614)

{
    "Message": "string",
    "ErrorDetails": [
    {
        "DetailedErrorCode": "string",
        "DetailedErrorMessage": "string"
    }
    ]
}

GeneralError

General error information.
type: object
required: ["Message"]

Message

A message that contains information about the error.
type: string

ErrorDetails

A list of error details.
type: array
items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.
type: object

DetailedErrorCode

A detailed error code.
type: string

DetailedErrorMessage

A detailed error message.
type: string

PUT

PUT /greengrass/definition/connectors/ConnectorDefinitionId
Operation ID: UpdateConnectorDefinition (p. 250)

Updates the name of a connector definition. To update the list of available connectors, use CreateConnectorDefinitionVersion.

Produces: application/json

Body Parameters

**UpdateDefinitionRequestBody** (p. 578)

where used: body; required: true

```json
{
  "Name": "string"
}
```

Name

The name of the definition.

required: true
type: string

Path Parameters

**ConnectorDefinitionId** (p. 546)

The ID of the connector definition.

where used: path; required: true
type: string

CLI

```bash
aws greengrass update-connector-definition \
  --connector-definition-id <value> \n  [--name <value>] \n  [--cli-input-json <value>] \n  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "ConnectorDefinitionId": "string",
  "Name": "string"
}
```

Responses

200

Success.

Empty (p. 600)
Empty Schema
Empty
type: object

400
Invalid request.

GeneralError (p. 614)

{
   "Message": "string",
   "ErrorDetails": [
   {
   "DetailedErrorCode": "string",
   "DetailedErrorMessage": "string"
   }
   ]
}

GeneralError
General error information.
type: object
required: ["Message"]

Message
A message that contains information about the error.
type: string

ErrorDetails
A list of error details.
type: array
items: ErrorDetail (p. 600)

ErrorDetail
Details about the error.
type: object

DetailedErrorCode
A detailed error code.
type: string

DetailedErrorMessage
A detailed error message.
type: string
GET /greengrass/definition/connectors/ConnectorDefinitionId/versions

GET /greengrass/definition/connectors/ConnectorDefinitionId/versions

Operation ID: ListConnectorDefinitionVersions (p. 188)

Lists the versions of a connector definition, which are containers for connectors. Connectors run on the
Greengrass core and contain built-in integration with local infrastructure, device protocols, AWS, and
other cloud services.

Produces: application/json

Path Parameters

ConnectorDefinitionId (p. 546)

The ID of the connector definition.

where used: path; required: true

type: string

Query Parameters

NextToken (p. 573)

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

where used: query; required: false

type: string

MaxResults (p. 573)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

CLI

aws greengrass list-connector-definition-versions \n   --connector-definition-id <value> \n   [--next-token <value>] \n   [--max-results <value>] \n   [--cli-input-json <value>] \n   [--generate-cli-skeleton]

cli-input-json format:

{  "ConnectorDefinitionId": "string",
"NextToken": "string",
"MaxResults": "integer"
}

Responses

200 (ListConnectorDefinitionVersionsResponse)

ListVersionsResponse (p. 650)

{
   "Versions": [
      {
         "Arn": "string",
         "Id": "string",
         "Version": "string",
         "CreationTimestamp": "string"
      }
   ],
   "NextToken": "string"
}

ListVersionsResponse

A list of versions.

   type: object

Versions

   Information about a version.

   type: array

   items: VersionInformation (p. 676)

VersionInformation

   Information about a version.

   type: object

Arn

   The ARN of the version.

   type: string

Id

   The ID of the parent definition that the version is associated with.

   type: string

Version

   The ID of the version.

   type: string

CreationTimestamp

   The time, in milliseconds since the epoch, when the version was created.

   type: string
NextToken

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

type: string

400

Invalid request.

GeneralError (p. 614)

{
   "Message": "string",
   "ErrorDetails": [
      {
         "DetailedErrorCode": "string",
         "DetailedErrorMessage": "string"
      }
   ]
}

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

POST

POST /greengrass/definition/connectors/ConnectorDefinitionId/versions
Operation ID: CreateConnectorDefinitionVersion (p. 13)

Creates a version of a connector definition that has already been defined.

Produces: application/json

Body Parameters

CreateConnectorDefinitionVersionRequestBody (p. 548)

where used: body; required: true

```json
{
  "Connectors": [
    {
      "Id": "string",
      "ConnectorArn": "string",
      "Parameters": {
        "additionalProperty0": "string",
        "additionalProperty1": "string",
        "additionalProperty2": "string"
      }
    }
  ]
}
```

ConnectorDefinitionVersion

Information about the connector definition version, which is a container for connectors.

type: object

Connectors

A list of references to connectors in this version, with their corresponding configuration settings.

type: array

items: Connector (p. 588)

Connector

Information about a connector. Connectors run on the Greengrass core and contain built-in integration with local infrastructure, device protocols, AWS, and other cloud services.

type: object

required: ["Id", "ConnectorArn"]

Id

A descriptive or arbitrary ID for the connector. This value must be unique within the connector definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9_:.-]+.

type: string

ConnectorArn

The ARN of the connector.

type: string

Parameters

The parameters or configuration used by the connector.
type: object

additionalProperties: An object with properties of type string that represent the connector configuration.

Header Parameters

X-Amzn-Client-Token (p. 579)

A client token used to enforce the idempotency of this API.

where used: header; required: false
type: string

Path Parameters

ConnectorDefinitionId (p. 546)

The ID of the connector definition.

where used: path; required: true
type: string

CLI

aws greengrass create-connector-definition-version \
   --connector-definition-id <value> \
   [--connectors <value>] \
   [--amzn-client-token <value>] \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]

cli-input-json format:

```
{
   "ConnectorDefinitionId": "string",
   "Connectors": [
   {
      "Id": "string",
      "ConnectorArn": "string",
      "Parameters": {
         "additionalProperty0": "string",
         "additionalProperty1": "string",
         "additionalProperty2": "string"
      }
   },
   "AmznClientToken": "string"
}
```

Responses

200 (CreateConnectorDefinitionVersionResponse)

VersionInformation (p. 676)
VersionInformation

Information about a version.

type: object

Arn

The ARN of the version.

type: string

Id

The ID of the parent definition that the version is associated with.

type: string

Version

The ID of the version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the version was created.

type: string

400

Invalid request.

GeneralError (p. 614)

{
   "Message": "string",
   "ErrorDetails": [
      {
         "DetailedErrorCode": "string",
         "DetailedErrorMessage": "string"
      }
   ]
}

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.
/aws greengrass/definition/connectors/
ConnectorDefinitionId/versions/
ConnectorDefinitionVersionId

    type: string
  ErrorDetails
    A list of error details.
    type: array
    items: ErrorDetail (p. 600)
  ErrorDetail
    Details about the error.
    type: object
  DetailedErrorCode
    A detailed error code.
    type: string
  DetailedErrorMessage
    A detailed error message.
    type: string

/greengrass/definition/connectors/
ConnectorDefinitionId/versions/
ConnectorDefinitionVersionId

GET

GET /greengrass/definition/connectors/ConnectorDefinitionId/
   versions/ConnectorDefinitionVersionId

Operation ID: GetConnectorDefinitionVersion (p. 121)

Retrieves information about a connector definition version, including the connectors that the version
contains. Connectors are prebuilt modules that interact with local infrastructure, device protocols, AWS,
and other cloud services.

Produces: application/json

Path Parameters

ConnectorDefinitionId (p. 546)
  The ID of the connector definition.
  where used: path; required: true
  type: string

ConnectorDefinitionVersionId (p. 546)
  The ID of the connector definition version. This value maps to the Version
  property of the corresponding VersionInformation object, which is returned by
  ListConnectorDefinitionVersions requests. If the version is the last one that was associated

with a connector definition, the value also maps to the `LatestVersion` property of the corresponding `DefinitionInformation` object.

where used: path; required: true
type: string

**Query Parameters**

**NextToken (p. 573)**

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

where used: query; required: false
type: string

**CLI**

```
aws greengrass get-connector-definition-version
  --connector-definition-id <value> \
  --connector-definition-version-id <value> \n  [--next-token <value>] \n  [--cli-input-json <value>] \n  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "ConnectorDefinitionId": "string",
  "ConnectorDefinitionVersionId": "string",
  "NextToken": "string"
}
```

**Responses**

**200 (GetConnectorDefinitionVersionResponse)**

GetConnectorDefinitionVersionResponse (p. 618)

```
{
  "Arn": "string",
  "Id": "string",
  "Version": "string",
  "CreationTimestamp": "string",
  "Definition": {
    "Connectors": [
      {
        "Id": "string",
        "ConnectorArn": "string",
        "Parameters": {
          "additionalProperty0": "string",
          "additionalProperty1": "string",
          "additionalProperty2": "string"
        }
      }
    ],
    "NextToken": "string"
  }
}
```
GetConnectorDefinitionVersionResponse

Information about a connector definition version.

type: object

Arn

The ARN of the connector definition version.

type: string

Id

The ID of the connector definition version.

type: string

Version

The version of the connector definition version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the connector definition version was created.

type: string

Definition

Information about the connector definition version, which is a container for connectors.

type: object

Connectors

A list of references to connectors in this version, with their corresponding configuration settings.

type: array

items: Connector (p. 588)

Connector

Information about a connector. Connectors run on the Greengrass core and contain built-in integration with local infrastructure, device protocols, AWS, and other cloud services.

type: object

required: ["Id", "ConnectorArn"]

Id

A descriptive or arbitrary ID for the connector. This value must be unique within the connector definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-]+.

type: string

ConnectorArn

The ARN of the connector.

type: string
Parameters

The parameters or configuration used by the connector.

type: object

additionalProperties: An object with properties of type string that represent the connector configuration.

NextToken

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

type: string

400

Invalid request.

GeneralError (p. 614)

```json
{
    "Message": "string",
    "ErrorDetails": [
    {
        "DetailedErrorCode": "string",
        "DetailedErrorMessage": "string"
    }
    ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.
type: string

/greengrass/definition/cores

GET

GET /greengrass/definition/cores

Operation ID: ListCoreDefinitions (p. 196)

Retrieves a list of core definitions.

Produces: application/json

Query Parameters

MaxResults (p. 573)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

NextToken (p. 573)

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

where used: query; required: false

type: string

CLI

aws greengrass list-core-definitions
   [--max-results <value>]
   [--next-token <value>]
   [--cli-input-json <value>]
   [--generate-cli-skeleton]

cli-input-json format:

```
{
   "MaxResults": "integer",
   "NextToken": "string"
}
```

Responses

200 (ListCoreDefinitionsResponse)

ListDefinitionsResponse (p. 645)

```
{
   "Definitions": [
      {
         "Name": "string",
```
"Id": "string",
"Arn": "string",
"tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
},
"LastUpdatedTimestamp": "string",
"CreationTimestamp": "string",
"LatestVersion": "string",
"LatestVersionArn": "string"
},
"NextToken": "string"
}

ListDefinitionsResponse

A list of definitions.

type: object

Definitions

Information about a definition.

type: array

items: DefinitionInformation (p. 594)

DefinitionInformation

Information about a definition.

type: object

Name

The name of the definition.

type: string

Id

The ID of the definition.

type: string

Arn

The ARN of the definition.

type: string

tags

The resource tags.

type: object

additionalProperties: The key-value pair for the resource tag. Type: string

LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.

type: string
CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.

type: string

LatestVersion

The ID of the latest version associated with the definition.

type: string

LatestVersionArn

The ARN of the latest version associated with the definition.

type: string

NextToken

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

type: string

**POST**

**POST** /greengrass/definition/cores

**Operation ID:** CreateCoreDefinition (p. 16)

Creates a core definition. You can provide the initial version of the core definition now or use CreateCoreDefinitionVersion later. Greengrass groups must each contain exactly one Greengrass core.

Produces: application/json

**Body Parameters**

CreateCoreDefinitionRequestBody (p. 549)

Information required to create a core definition.

where used: body; required: true

```json
{
  "Name": "string",
  "InitialVersion": {
    "Cores": [
    {
      "Id": "string",
      "ThingArn": "string",
      "CertificateArn": "string",
      "SyncShadow": true
    }
    ],
  },
  "tags": {
  "additionalProperty0": "string",
  "additionalProperty1": "string",
  "additionalProperty2": "string"
  }
}
```
Name
The name of the core definition.
type: string

InitialVersion
Information about a core definition version.
type: object

Cores
A list of cores in the core definition version.
type: array
items: Core (p. 590)

Core
Information about a core.
type: object
required: ["Id", "ThingArn", "CertificateArn"]

Id
A descriptive or arbitrary ID for the core. This value must be unique within the core definition version. Maximum length is 128 characters with the pattern \[a-zA-Z0-9:_-\]+. 
type: string

ThingArn
The ARN of the thing that is the core.
type: string

CertificateArn
The ARN of the certificate associated with the core.
type: string

SyncShadow
If true, the core's local shadow is synced with the cloud automatically.
type: boolean

tags
The resource tags.
type: object
additionalProperties: The key-value pair for the resource tag. Type: string

Header Parameters

X-Amzn-Client-Token (p. 579)
A client token used to correlate requests and responses.
CLI

```bash
cli-input-json format:
{
  "Name": "string",
  "InitialVersion": {
    "Cores": [
      {
        "Id": "string",
        "ThingArn": "string",
        "CertificateArn": "string",
        "SyncShadow": "boolean"
      }
    ],
  },
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  },
  "AmznClientToken": "string"
}
```

Responses

**200** *(CreateCoreDefinitionResponse)*

```json
DefinitionInformation (p. 594)
{
  "Name": "string",
  "Id": "string",
  "Arn": "string",
  "LastUpdatedTimestamp": "string",
  "CreationTimestamp": "string",
  "LatestVersion": "string",
  "LatestVersionArn": "string"
}
```
type: string
Id
The ID of the definition.
type: string
Arn
The ARN of the definition.
type: string
LastUpdatedTimestamp
The time, in milliseconds since the epoch, when the definition was last updated.
type: string
CreationTimestamp
The time, in milliseconds since the epoch, when the definition was created.
type: string
LatestVersion
The ID of the latest version associated with the definition.
type: string
LatestVersionArn
The ARN of the latest version associated with the definition.
type: string

400
Invalid request.

**GeneralError (p. 614)**

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

**GeneralError**

General error information.
type: object

required: ["Message"]

**Message**

A message that contains information about the error.
type: string
/greengrass/definition/cores/CoreDefinitionId

GET

GET /greengrass/definition/cores/CoreDefinitionId

Operation ID: GetCoreDefinition (p. 125)

Retrieves information about a core definition version.

Produces: application/json

Path Parameters

CoreDefinitionId (p. 547)

The ID of the core definition.

where used: path; required: true
type: string

CLI

```
aws greengrass get-core-definition \
  --core-definition-id <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "CoreDefinitionId": "string"
}
```
Responses

200 (GetCoreDefinitionResponse)

DefinitionInformation (p. 594)

```
{
  "Name": "string",
  "Id": "string",
  "Arn": "string",
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  },
  "LastUpdatedTimestamp": "string",
  "CreationTimestamp": "string",
  "LatestVersion": "string",
  "LatestVersionArn": "string"
}
```

DefinitionInformation

Information about a definition.

type: object

Name

The name of the definition.

type: string

Id

The ID of the definition.

type: string

Arn

The ARN of the definition.

type: string

tags

The resource tags.

type: object

  additionalProperties: The key-value pair for the resource tag. Type: string

LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.

type: string
LatestVersion

The ID of the latest version associated with the definition.

type: string

LatestVersionArn

The ARN of the latest version associated with the definition.

type: string

400

Invalid request.

**GeneralError (p. 614)**

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

**GeneralError**

General error information.

type: object

required: ["Message"]

**Message**

A message that contains information about the error.

type: string

**ErrorDetails**

A list of error details.

type: array

items: **ErrorDetail (p. 600)**

**ErrorDetail**

Details about the error.

type: object

**DetailedErrorCode**

A detailed error code.

type: string

**DetailedErrorMessage**

A detailed error message.

type: string
DELETE

DELETE /greengrass/definition/cores/CoreDefinitionId

Operation ID: DeleteCoreDefinition (p. 93)

Deletes a core definition.

Produces: application/json

Path Parameters

CoreDefinitionId (p. 547)

The ID of the core definition.

where used: path; required: true

type: string

CLI

```
aws greengrass delete-core-definition \
  --core-definition-id <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "CoreDefinitionId": "string"
}
```

Responses

200

Success.

Empty (p. 600)

```
{
}
```

Empty Schema

Empty

type: object

400

Invalid request.

GeneralError (p. 614)

```
{
  "Message": "string",
```
"ErrorDetails": [
  {
    "DetailedErrorCode": "string",
    "DetailedErrorMessage": "string"
  }
]
}

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

**PUT**

PUT /greengrass/definition/cores/CoreDefinitionId

Operation ID: UpdateCoreDefinition (p. 252)

Updates the name of a core definition. To update the list of available cores, use CreateCoreDefinitionVersion.

Produces: application/json

Body Parameters

*UpdateDefinitionRequestBody (p. 578)*

where used: body; required: true
{  
  "Name": "string"
}

Name

The name of the definition.

required: true

type: string

Path Parameters

CoreDefinitionId (p. 547)

The ID of the core definition.

where used: path; required: true

type: string

CLI

`aws greengrass update-core-definition \  
  --core-definition-id <value> \  
  [--name <value>] \  
  [--cli-input-json <value>] \  
  [--generate-cli-skeleton]`

cli-input-json format:

```
{
  "CoreDefinitionId": "string",
  "Name": "string"
}
```

Responses

200

Success.

Empty (p. 600)

```
{
}
```

Empty Schema

Empty

type: object

400

Invalid request.
GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

**GeneralError**

General error information.

type: object

required: ["Message"]

**Message**

A message that contains information about the error.

type: string

**ErrorDetails**

A list of error details.

type: array

items: ErrorDetail (p. 600)

**ErrorDetail**

Details about the error.

type: object

**DetailedErrorCode**

A detailed error code.

type: string

**DetailedErrorMessage**

A detailed error message.

type: string

//!greengrass/definition/cores/CoreDefinitionId/versions

**POST**

**POST** /greengrass/definition/cores/CoreDefinitionId/versions

Operation ID: CreateCoreDefinitionVersion (p. 20)
Creates a version of a core definition that has already been defined. Greengrass groups must each contain exactly one Greengrass core.

Produce: application/json

### Body Parameters

CreateCoreDefinitionVersionRequestBody (p. 550)

where used: body; required: true

```json
{
   "Cores": [
      {
         "Id": "string",
         "ThingArn": "string",
         "CertificateArn": "string",
         "SyncShadow": true
      }
   ]
}
```

CoreDefinitionVersion

Information about a core definition version.

**type:** object

Cores

A list of cores in the core definition version.

**type:** array

items: Core (p. 590)

Core

Information about a core.

**type:** object

- **Id**: A descriptive or arbitrary ID for the core. This value must be unique within the core definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-]+.

  **type:** string

- **ThingArn**: The ARN of the thing that is the core.

  **type:** string

- **CertificateArn**: The ARN of the certificate associated with the core.

  **type:** string

- **SyncShadow**: If true, the core’s local shadow is synced with the cloud automatically.
type: boolean

Header Parameters

X-Amzn-Client-Token (p. 579)

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Path Parameters

CoreDefinitionId (p. 547)

The ID of the core definition.

where used: path; required: true

type: string

CLI

```bash
aws greengrass create-core-definition-version \
  --core-definition-id <value> \
  [--cores <value>] \
  [--amzn-client-token <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "CoreDefinitionId": "string",
  "Cores": [
    {
      "Id": "string",
      "ThingArn": "string",
      "CertificateArn": "string",
      "SyncShadow": "boolean"
    },
  ],
  "AmznClientToken": "string"
}
```

Responses

200 (CreateCoreDefinitionVersionResponse)

```json
{
  "Arn": "string",
  "Id": "string",
  "Version": "string",
  "CreationTimestamp": "string"
}
```
VersionInformation

Information about a version.

- **Arn**: The ARN of the version.
  - type: string
- **Id**: The ID of the parent definition that the version is associated with.
  - type: string
- **Version**: The ID of the version.
  - type: string
- **CreationTimestamp**: The time, in milliseconds since the epoch, when the version was created.
  - type: string

400

Invalid request.

**GeneralError (p. 614)**

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

**GeneralError**

General error information.

- type: object
  - required: ["Message"]
- **Message**: A message that contains information about the error.
  - type: string
- **ErrorDetails**: A list of error details.
  - type: array
ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

**GET**

GET /greengrass/definition/cores/CoreDefinitionId/versions

Operation ID: ListCoreDefinitionVersions (p. 193)

Lists the versions of a core definition.

Produces: application/json

**Path Parameters**

CoreDefinitionId (p. 547)

The ID of the core definition.

where used: path; required: true

type: string

**Query Parameters**

NextToken (p. 573)

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

where used: query; required: false

type: string

MaxResults (p. 573)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

**CLI**

aws greengrass list-core-definition-versions \

```
--core-definition-id <value> \
[--next-token <value>] \
[--max-results <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "CoreDefinitionId": "string",
  "NextToken": "string",
  "MaxResults": "integer"
}
```

**Responses**

**200** *(ListCoreDefinitionVersionsResponse)*

ListVersionsResponse *(p. 650)*

```
{
  "Versions": [
    {
      "Arn": "string",
      "Id": "string",
      "Version": "string",
      "CreationTimestamp": "string"
    }
  ],
  "NextToken": "string"
}
```

ListVersionsResponse

A list of versions.

type: object

Versions

Information about a version.

type: array

items: VersionInformation *(p. 676)*

VersionInformation

Information about a version.

type: object

Arn

The ARN of the version.

type: string

Id

The ID of the parent definition that the version is associated with.

type: string
Version
The ID of the version.
type: string

CreationTimestamp
The time, in milliseconds since the epoch, when the version was created.
type: string

NextToken
The token for the next set of results, or null if there are no more results.
type: string

400
Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError
General error information.
type: object
required: ["Message"]

Message
A message that contains information about the error.
type: string

ErrorDetails
A list of error details.
type: array
items: ErrorDetail (p. 600)

ErrorDetail
Details about the error.
type: object

DetailedErrorCode
A detailed error code.
type: string
DetaileError Message

A detailed error message.

type: string

/greengrass/definition/cores/CoreDefinitionId/versions/CoreDefinitionVersionId

GET

GET /greengrass/definition/cores/CoreDefinitionId/versions/CoreDefinitionVersionId

Operation ID: GetCoreDefinitionVersion (p. 127)

Retrieves information about a core definition version.

Produces: application/json

Path Parameters

CoreDefinitionId (p. 547)

The ID of the core definition.

where used: path; required: true

type: string

CoreDefinitionVersionId (p. 547)

The ID of the core definition version. This value maps to the Version property of the corresponding VersionInformation object, which is returned by ListCoreDefinitionVersions requests. If the version is the last one that was associated with a core definition, the value also maps to the LatestVersion property of the corresponding DefinitionInformation object.

where used: path; required: true

type: string

CLI

aws greengrass get-core-definition-version \
  --core-definition-id <value> \
  --core-definition-version-id <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]

cli-input-json format:

{  
  "CoreDefinitionId": "string",  
  "CoreDefinitionVersionId": "string"  
}
Responses

200

Success.

GetCoreDefinitionVersionResponse (p. 620)

```json
{
  "Arn": "string",
  "Id": "string",
  "Version": "string",
  "CreationTimestamp": "string",
  "Definition": {
    "Cores": [
      {
        "Id": "string",
        "ThingArn": "string",
        "CertificateArn": "string",
        "SyncShadow": true
      }
    ],
  },
  "NextToken": "string"
}
```

GetCoreDefinitionVersionResponse

type: object

Arn

The ARN of the core definition version.

type: string

Id

The ID of the core definition version.

type: string

Version

The version of the core definition version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the core definition version was created.

type: string

Definition

Information about a core definition version.

type: object

Cores

A list of cores in the core definition version.

type: array

items: Core (p. 590)
Core

Information about a core.

type: object

required: ["Id", "ThingArn", "CertificateArn"]

Id

A descriptive or arbitrary ID for the core. This value must be unique within the core definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-]+.

type: string

ThingArn

The ARN of the thing that is the core.

type: string

CertificateArn

The ARN of the certificate associated with the core.

type: string

SyncShadow

If true, the core's local shadow is synced with the cloud automatically.

type: boolean

NextToken

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

type: string

400

Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string
ErrorDetails
   A list of error details.
   type: array
   items: ErrorDetail (p. 600)
ErrorDetail
   Details about the error.
   type: object
DetailedErrorCode
   A detailed error code.
   type: string
DetailedErrorMessage
   A detailed error message.
   type: string

/greengrass/definition/devices
GET
GET /greengrass/definition/devices
Operation ID: ListDeviceDefinitions (p. 204)
Retrieves a list of device definitions.
Produces: application/json
Query Parameters
MaxResults (p. 573)
   The maximum number of results to be returned per request.
   where used: query; required: false
   type: integer
NextToken (p. 573)
   The token for the next set of results, or null if there are no more results.
   where used: query; required: false
   type: string

CLI
aws greengrass list-device-definitions \
   [--max-results <value>] \
   [--next-token <value>] \
   [--cli-input-json <value>] \n
cli-input-json format:

```json
{
  "MaxResults": "integer",
  "NextToken": "string"
}
```

Responses

**200 (ListDeviceDefinitionsResponse)**

ListDefinitionsResponse (p. 645)

```json
{
  "Definitions": [
    {
      "Name": "string",
      "Id": "string",
      "Arn": "string",
      "tags": {
        "additionalProperty0": "string",
        "additionalProperty1": "string",
        "additionalProperty2": "string"
      },
      "LastUpdatedTimestamp": "string",
      "CreationTimestamp": "string",
      "LatestVersion": "string",
      "LatestVersionArn": "string"
    }
  ],
  "NextToken": "string"
}
```

ListDefinitionsResponse

A list of definitions.

type: object

Definitions

Information about a definition.

type: array

items: DefinitionInformation (p. 594)

DefinitionInformation

Information about a definition.

type: object

Name

The name of the definition.

type: string

Id

The ID of the definition.
Post

Post /greengrass/definition/devices

Operation ID: CreateDeviceDefinition (p. 26)

Creates a device definition. You can provide the initial version of the device definition now or use CreateDeviceDefinitionVersion later.

Produces: application/json

Body Parameters

CreateDeviceDefinitionRequestBody (p. 551)

Information required to create a device definition.

where used: body; required: true

{
"Name": "string",
"InitialVersion": {
  "Devices": [
    {
      "Id": "string",
      "ThingArn": "string",
      "CertificateArn": "string",
      "SyncShadow": true
    }
  ],
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  }
},

Name

The name of the device definition.

  type: string

InitialVersion

Information about a device definition version.

  type: object

Devices

A list of devices in the definition version.

  type: array

  items: Device (p. 597)

Device

Information about a device.

  type: object

  required: ["Id", "ThingArn", "CertificateArn"]

Id

A descriptive or arbitrary ID for the device. This value must be unique within the device definition version. Maximum length is 128 characters with the pattern `[a-zA-Z0-9:_-]+`. 

  type: string

ThingArn

The thing ARN of the device.

  type: string

CertificateArn

The ARN of the certificate associated with the device.

  type: string

SyncShadow

If true, the device's local shadow is synced with the cloud automatically.
type: boolean

tags

The resource tags.

type: object

additionalProperties: The key-value pair for the resource tag. Type: string

Header Parameters

X-Amzn-Client-Token (p. 579)

A client token used to correlate requests and responses.

where used: header; required: false

type: string

CLI

aws greengrass create-device-definition
[--name <value>] 
[--initial-version <value>] 
[--tags <value>] 
[--amzn-client-token <value>] 
[--cli-input-json <value>] 
[--generate-cli-skeleton]

cli-input-json format:

{
    "Name": "string",
    "InitialVersion": {
        "Devices": [
            {
                "Id": "string",
                "ThingArn": "string",
                "CertificateArn": "string",
                "SyncShadow": "boolean"
            }
        ],
        "tags": {
            "additionalProperty0": "string",
            "additionalProperty1": "string",
            "additionalProperty2": "string"
        },
        "AmznClientToken": "string"
    }
}

Responses

200 (CreateDeviceDefinitionResponse)

DefinitionInformation (p. 594)

{

"Name": "string",
"Id": "string",
"Arn": "string",
"LastUpdatedTimestamp": "string",
"CreationTimestamp": "string",
"LatestVersion": "string",
"LatestVersionArn": "string"
}

**DefinitionInformation**

Information about a definition.

- **Name**
  - type: string
  - The name of the definition.

- **Id**
  - type: string
  - The ID of the definition.

- **Arn**
  - type: string
  - The ARN of the definition.

- **LastUpdatedTimestamp**
  - type: string
  - The time, in milliseconds since the epoch, when the definition was last updated.

- **CreationTimestamp**
  - type: string
  - The time, in milliseconds since the epoch, when the definition was created.

- **LatestVersion**
  - type: string
  - The ID of the latest version associated with the definition.

- **LatestVersionArn**
  - type: string
  - The ARN of the latest version associated with the definition.

**400**

Invalid request.

**GeneralError (p. 614)**

{
  "Message": "string",
  "ErrorDetails": [
  {
  
}
"DetailedErrorCode": "string",
"DetailedErrorMessage": "string"
]
}

GeneralError

General error information.
type: object
required: ["Message"]

Message

A message that contains information about the error.
type: string

ErrorDetails

A list of error details.
type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.
type: object

DetailedErrorCode

A detailed error code.
type: string

DetailedErrorMessage

A detailed error message.
type: string

/greengrass/definition/devices/DeviceDefinitionId

GET

GET /greengrass/definition/devices/{DeviceDefinitionId}

Operation ID: GetDeviceDefinition (p. 133)

Retrieves information about a device definition.

Produces: application/json

Path Parameters

DeviceDefinitionId (p. 570)

The ID of the device definition.
where used: path; required: true
type: string

CLI

aws greengrass get-device-definition \
  --device-definition-id <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]

cli-input-json format:

{
  "DeviceDefinitionId": "string"
}

Responses

200 (GetDeviceDefinitionResponse)

DefinitionInformation (p. 594)

{
  "Name": "string",
  "Id": "string",
  "Arn": "string",
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  },
  "LastUpdatedTimestamp": "string",
  "CreationTimestamp": "string",
  "LatestVersion": "string",
  "LatestVersionArn": "string"
}

DefinitionInformation
Information about a definition.
type: object
Name
The name of the definition.
type: string
Id
The ID of the definition.
type: string
Arn
The ARN of the definition.
type: string
tags
  The resource tags.
  type: object
  additionalProperties: The key-value pair for the resource tag. Type: string

LastUpdatedTimestamp
  The time, in milliseconds since the epoch, when the definition was last updated.
  type: string

CreationTimestamp
  The time, in milliseconds since the epoch, when the definition was created.
  type: string

LatestVersion
  The ID of the latest version associated with the definition.
  type: string

LatestVersionArn
  The ARN of the latest version associated with the definition.
  type: string

400
Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError
  General error information.
  type: object
  required: ["Message"]

Message
  A message that contains information about the error.
  type: string

ErrorDetails
  A list of error details.
  type: array
items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

DELETE

DELETE /greengrass/definition/devices/DeviceDefinitionId

Operation ID: DeleteDeviceDefinition (p. 95)

Deletes a device definition.

Produces: application/json

Path Parameters

DeviceDefinitionId (p. 570)

The ID of the device definition.

where used: path; required: true

type: string

CLI

```
aws greengrass delete-device-definition \ 
  --device-definition-id <value> \ 
  [--cli-input-json <value>] \ 
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "DeviceDefinitionId": "string"
}
```

Responses

200

Success.
Empty (p. 600)

{  
  
}

Empty Schema

Empty
type: object

400

Invalid request.

GeneralError (p. 614)

{  
  "Message": "string",
  "ErrorDetails": [
    
    
  ]
}

GeneralError

General error information.
type: object

required: ["Message"]

Message

A message that contains information about the error.
type: string

ErrorDetails

A list of error details.
type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.
type: object

DetailedErrorCode

A detailed error code.
type: string

DetailedErrorMessage

A detailed error message.
type: string

PUT

PUT /greengrass/definition/devices/DeviceDefinitionId

Operation ID: UpdateDeviceDefinition (p. 254)

Updates the name of a device definition. To update the list of available devices, use CreateDeviceDefinitionVersion.

Produce: application/json

Body Parameters

UpdateDefinitionRequestBody (p. 578)

where used: body; required: true

```
{
   "Name": "string"
}
```

Name

The name of the definition.

required: true

type: string

Path Parameters

DeviceDefinitionId (p. 570)

The ID of the device definition.

where used: path; required: true

type: string

CLI

```
aws greengrass update-device-definition \
   --device-definition-id <value> \
   [--name <value>] \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```
{
   "DeviceDefinitionId": "string",
   "Name": "string"
}
```
Responses

200

Success.

Empty (p. 600)

```json
{
}
```

Empty Schema

Empty

type: object

400

Invalid request.

GeneralError (p. 614)

```json
{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.
/greengrass/definition/devices/DeviceDefinitionId/versions

POST

POST /greengrass/definition/devices/DeviceDefinitionId/versions

Operation ID: CreateDeviceDefinitionVersion (p. 30)

Creates a version of a device definition that has already been defined.

Produces: application/json

Body Parameters

CreateDeviceDefinitionVersionRequestBody (p. 552)

where used: body; required: true

```json
{
    "Devices": [
        {
            "Id": "string",
            "ThingArn": "string",
            "CertificateArn": "string",
            "SyncShadow": true
        }
    ]
}
```

DeviceDefinitionVersion

Information about a device definition version.

type: object

Devices

A list of devices in the definition version.

type: array

items: Device (p. 597)

Device

Information about a device.

type: object

required: ["Id", "ThingArn", "CertificateArn"]
Id

A descriptive or arbitrary ID for the device. This value must be unique within the device definition version. Maximum length is 128 characters with the pattern \[a-zA-Z0-9_:\-\]+. 

type: string

ThingArn

The thing ARN of the device. 

type: string

CertificateArn

The ARN of the certificate associated with the device. 

type: string

SyncShadow

If true, the device's local shadow is synced with the cloud automatically. 

type: boolean

Header Parameters

X-Amzn-Client-Token (p. 579)

A client token used to correlate requests and responses. 

where used: header; required: false

type: string

Path Parameters

DeviceDefinitionId (p. 570)

The ID of the device definition. 

where used: path; required: true

type: string

CLI

```bash
aws greengrass create-device-definition-version \
   --device-definition-id <value> \
   [--devices <value>] \
   [--amzn-client-token <value>] \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "DeviceDefinitionId": "string",
   "Devices": [
```


```json
{
    "Id": "string",
    "ThingArn": "string",
    "CertificateArn": "string",
    "SyncShadow": "boolean"
}

"AmznClientToken": "string"
}
```

### Responses

#### 200

**CreateDeviceDefinitionVersionResponse**

VersionInformation (p. 676)

```json
{
    "Arn": "string",
    "Id": "string",
    "Version": "string",
    "CreationTimestamp": "string"
}
```

VersionInformation

Information about a version.

**Arn**

The ARN of the version.

**Id**

The ID of the parent definition that the version is associated with.

**Version**

The ID of the version.

**CreationTimestamp**

The time, in milliseconds since the epoch, when the version was created.

#### 400

Invalid request.

**GeneralError** (p. 614)

```json
{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
```

345
GET /greengrass/definition/devices/DeviceDefinitionId/versions

**Operation ID:** ListDeviceDefinitionVersions (p. 201)

Lists the versions of a device definition.

**Produces:** application/json

**Path Parameters**

**DeviceDefinitionId** (p. 570)

The ID of the device definition.

where used: path; required: true
type: string

Query Parameters

NextToken (p. 573)

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

where used: query; required: false

type: string

MaxResults (p. 573)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

CLI

```
aws greengrass list-device-definition-versions \
  --device-definition-id <value> \
  [--next-token <value>] \
  [--max-results <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "DeviceDefinitionId": "string",
  "NextToken": "string",
  "MaxResults": "integer"
}
```

Responses

**200** (ListDeviceDefinitionVersionsResponse)

ListVersionsResponse (p. 650)

```
{
  "Versions": [
    {
      "Arn": "string",
      "Id": "string",
      "Version": "string",
      "CreationTimestamp": "string"
    }
  ],
  "NextToken": "string"
}
```

ListVersionsResponse

A list of versions.
type: object

Versions

Information about a version.

type: array

items: VersionInformation (p. 676)

VersionInformation

Information about a version.

type: object

Arn

The ARN of the version.

type: string

Id

The ID of the parent definition that the version is associated with.

type: string

Version

The ID of the version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the version was created.

type: string

NextToken

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

type: string

400

Invalid request.

GeneralError (p. 614)

{
   "Message": "string",
   "ErrorDetails": [
      {
         "DetailedErrorCode": "string",
         "DetailedErrorMessage": "string"
      }
   ]
}

GeneralError

General error information.

type: object
Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

GET /greengrass/definition/devices/DeviceDefinitionId/versions/DeviceDefinitionVersionId

Operation ID: GetDeviceDefinitionVersion (p. 136)

Retrieves information about a device definition version.

Produces: application/json

Path Parameters

DeviceDefinitionId (p. 570)

The ID of the device definition.

where used: path; required: true

type: string

DeviceDefinitionVersionId (p. 570)

The ID of the device definition version. This value maps to the Version property of the corresponding VersionInformation object, which is returned by
ListDeviceDefinitionVersions requests. If the version is the last one that was associated with a device definition, the value also maps to the LatestVersion property of the corresponding DefinitionInformation object.

where used: path; required: true
type: string

Query Parameters

**NextToken (p. 573)**

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

where used: query; required: false
type: string

CLI

```bash
aws greengrass get-device-definition-version \
  --device-definition-id <value> \
  --device-definition-version-id <value> \
  [--next-token <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "DeviceDefinitionId": "string",
  "DeviceDefinitionVersionId": "string",
  "NextToken": "string"
}
```

Responses

**200 (GetDeviceDefinitionVersionResponse)**

GetDeviceDefinitionVersionResponse (p. 623)

```json
{
  "Arn": "string",
  "Id": "string",
  "Version": "string",
  "CreationTimestamp": "string",
  "Definition": {
    "Devices": [
      {
        "Id": "string",
        "ThingArn": "string",
        "CertificateArn": "string",
        "SyncShadow": true
      }
    ],
    "NextToken": "string"
  }
}
```
GetDeviceDefinitionVersionResponse

   type: object

Arn

   The ARN of the device definition version.

   type: string

Id

   The ID of the device definition version.

   type: string

Version

   The version of the device definition version.

   type: string

CreationTimestamp

   The time, in milliseconds since the epoch, when the device definition version was created.

   type: string

Definition

   Information about a device definition version.

   type: object

Devices

   A list of devices in the definition version.

   type: array

   items: Device (p. 597)

Device

   Information about a device.

   type: object

   required: ["Id", "ThingArn", "CertificateArn"]

Id

   A descriptive or arbitrary ID for the device. This value must be unique within the device definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-]+.

   type: string

ThingArn

   The thing ARN of the device.

   type: string

CertificateArn

   The ARN of the certificate associated with the device.

   type: string
SyncShadow

If true, the device's local shadow is synced with the cloud automatically.

type: boolean

NextToken

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

type: string

400

Invalid request.

GeneralError (p. 614)

{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}

GeneralError

General error information.

type: object

  required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

  items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.
GET /greengrass/definition/functions

Operation ID: ListFunctionDefinitions (p. 209)
Retrieves a list of Lambda function definitions.
Produces: application/json

Query Parameters

MaxResults (p. 573)
The maximum number of results to be returned per request.
where used: query; required: false

type: integer

NextToken (p. 573)
The token for the next set of results, or null if there are no more results.
where used: query; required: false

type: string

CLI

| awsgreengrass list-function-definitions | 
| [--max-results <value>] | 
| [--next-token <value>] | 
| [--cli-input-json <value>] | 
| [--generate-cli-skeleton] |

cli-input-json format:

```json
{
  "MaxResults": "integer",
  "NextToken": "string"
}
```

Responses

200 (ListFunctionDefinitionsResponse)

Success. The response contains the IDs of all of the Greengrass Lambda function definitions in this account.

ListDefinitionsResponse (p. 645)

```json
{
}
```
"Definitions": [ 
  { 
    "Name": "string",
    "Id": "string",
    "Arn": "string",
    "tags": { 
      "additionalProperty0": "string",
      "additionalProperty1": "string",
      "additionalProperty2": "string"
    },
    "LastUpdatedTimestamp": "string",
    "CreationTimestamp": "string",
    "LatestVersion": "string",
    "LatestVersionArn": "string"
  }
],
"NextToken": "string"

ListDefinitionsResponse

A list of definitions.

type: object

Definitions

Information about a definition.

type: array

items: DefinitionInformation (p. 594)

DefinitionInformation

Information about a definition.

type: object

Name

The name of the definition.

type: string

Id

The ID of the definition.

type: string

Arn

The ARN of the definition.

type: string

tags

The resource tags.

type: object

  additionalProperties: The key-value pair for the resource tag. Type: string

LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.
POST

POST /greengrass/definition/functions

Operation ID: CreateFunctionDefinition (p. 34)

Creates a function definition that contains a list of Lambda functions and their configurations to use in a
group. You can create an initial version of the definition now by providing a list of Lambda functions and
their configurations, or later by using CreateFunctionDefinitionVersion.

Produces: application/json

Body Parameters

CreateFunctionDefinitionRequestBody (p. 553)

where used: body; required: true

```json
{
  "Name": "string",
  "InitialVersion": {
    "DefaultConfig": {
      "Execution": {
        "IsolationMode": "GreengrassContainer|NoContainer",
        "RunAs": {
          "Uid": 1001,
          "Gid": 1002
        }
      }
    },
    "Functions": [
      {
        "Id": "string",
        "FunctionArn": "string",
        "FunctionConfiguration": {
          "Pinned": true,
          "Executable": "string"
        }
      }
    ]
  }
}
```
"ExecArgs": "string",
"MemorySize": 0,
"Timeout": 0,
"EncodingType": "binary|json",
"Environment": {
  "Variables": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  },
  "ResourceAccessPolicies": [
    {
      "ResourceId": "string",
      "Permission": "ro|rw"
    }
  ],
  "AccessSysfs": true,
  "Execution": {
    "IsolationMode": "GreengrassContainer|NoContainer",
    "RunAs": {
      "Uid": 1001,
      "Gid": 1002
    }
  }
},
"tags": {
  "additionalProperty0": "string",
  "additionalProperty1": "string",
  "additionalProperty2": "string"
}
}

Name

The name of the function definition.

  type: string

InitialVersion

Information about a function definition version.

  type: object

DefaultConfig

The default configuration that applies to all Lambda functions in the group. Individual Lambda functions can override these settings.

  type: object

Execution

Configuration information that specifies how a Lambda function runs.

Functions

A list of Lambda functions in this function definition version.

  type: array

  items: Function (p. 601)
Information about a Lambda function.

type: object

required: ["Id"]

Id

A descriptive or arbitrary ID for the function. This value must be unique within the function definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9_: -]+.

type: string

FunctionArn

The ARN of the alias (recommended) or version of the target Lambda function.

type: string

FunctionConfiguration

The configuration of the Lambda function.

type: object

Pinned

True if the function is pinned. Pinned means the function is long-lived and starts when the core starts.

type: boolean

Executable

The name of the function executable.

type: string

ExecArgs

The execution arguments.

type: string

MemorySize

The memory size, in KB, required by the function. This setting does not apply and should be cleared when you run the Lambda function without containerization.

type: integer

Timeout

The allowed function execution time, after which Lambda should terminate the function. This timeout still applies to pinned Lambda functions for each request.

type: integer

EncodingType

The expected encoding type of the input payload for the function. The default is json.

type: string

enum: ["binary", "json"]
Environment

The environment configuration of the function.

type: object

Variables

Environment variables for the Lambda function's configuration.

type: object

additionalProperties: An object with properties of type string that represent the environment variables.

ResourceAccessPolicies

A list of the resources, with their permissions, to which the Lambda function is granted access. A Lambda function can have at most 10 resources. ResourceAccessPolicies applies only when you run the Lambda function in a Greengrass container.

type: array

items: ResourceAccessPolicy (p. 660)

ResourceAccessPolicy

A policy used by the function to access a resource.

type: object

required: ["ResourceId"]

ResourceId

The ID of the resource. (This ID is assigned to the resource when you create the resource definition.)

type: string

Permission

The type of permission a function has to access a resource.

type: string

enum: ["ro", "rw"]

AccessSysfs

If true, the Lambda function is allowed to access the host's /sys folder. Use this when the Lambda function needs to read device information from /sys. This setting applies only when you run the Lambda function in a Greengrass container.

type: boolean

Execution

Configuration information that specifies how a Lambda function runs.

type: object

IsolationMode

Specifies whether the Lambda function runs in a Greengrass container (default) or without containerization. Unless your scenario requires that you run without containerization, we recommend that you run in a Greengrass container. Omit this value to run the Lambda function with the default containerization for the group.
RunAs

Specifies the user and group whose permissions are used when running the Lambda function. You can specify one or both values to override the default values. To minimize the risk of unintended changes or malicious attacks, we recommend that you avoid running as root unless absolutely necessary. To run as root, you must update config.json in `greengrass-root/config` to set `allowFunctionsToRunAsRoot` to `yes`.

type: object

Uid

The user ID whose permissions are used to run a Lambda function.

type: integer

Gid

The group ID whose permissions are used to run a Lambda function.

type: integer

tags

The resource tags.

type: object

additionalProperties: The key-value pair for the resource tag. Type: string

Header Parameters

X-Amzn-Client-Token (p. 579)

A client token used to correlate requests and responses.

where used: header; required: false

type: string

CLI

```
aws greengrass create-function-definition
  [--name <value>] \n  [--initial-version <value>] \n  [--tags <value>] \n  [--amzn-client-token <value>] \n  [--cli-input-json <value>] \n  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "Name": "string",
  "InitialVersion": {
    "DefaultConfig": {
      "Execution": {
```
"IsolationMode": "GreengrassContainer|NoContainer",
"RunAs": {
  "Uid": "integer",
  "Gid": "integer"
}
},
"Functions": [
{
  "Id": "string",
  "FunctionArn": "string",
  "FunctionConfiguration": {
    "Pinned": "boolean",
    "Executable": "string",
    "ExecArgs": "string",
    "MemorySize": "integer",
    "Timeout": "integer",
    "EncodingType": "binary|json",
    "Environment": {
      "Variables": {
        "additionalProperty0": "string",
        "additionalProperty1": "string",
        "additionalProperty2": "string"
      }
    },
    "ResourceAccessPolicies": [
      {
        "ResourceId": "string",
        "Permission": "ro|rw"
      }
    ],
    "AccessSysfs": "boolean",
    "Execution": {
      "IsolationMode": "GreengrassContainer|NoContainer",
      "RunAs": {
        "Uid": "integer",
        "Gid": "integer"
      }
    }
  }
}
},
"tags": {
  "additionalProperty0": "string",
  "additionalProperty1": "string",
  "additionalProperty2": "string"
},
"AmznClientToken": "string"

Responses

200 (CreateFunctionDefinitionResponse)

**DefinitionInformation (p. 594)**

{
  "Name": "string",
  "Id": "string",
  "Arn": "string",
  "LastUpdatedTimestamp": "string",
  "CreationTimestamp": "string",
  "LatestVersion": "string",
}
DefinitionInformation

Information about a definition.

type: object

Name

The name of the definition.

type: string

Id

The ID of the definition.

type: string

Arn

The ARN of the definition.

type: string

LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.

type: string

LatestVersion

The ID of the latest version associated with the definition.

type: string

LatestVersionArn

The ARN of the latest version associated with the definition.

type: string

400

Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```
GeneralError
  General error information.
  type: object
  required: ["Message"]
Message
  A message that contains information about the error.
  type: string
ErrorDetails
  A list of error details.
  type: array
  items: ErrorDetail (p. 600)
ErrorDetail
  Details about the error.
  type: object
DetailedErrorCode
  A detailed error code.
  type: string
DetailedErrorMessage
  A detailed error message.
  type: string

/greengrass/definition/functions/FunctionDefinitionId

GET

GET /greengrass/definition/functions/FunctionDefinitionId

Operation ID: GetFunctionDefinition (p. 139)

Retrieves information about a Lambda function definition, including its creation time and latest version.

Produces: application/json

Path Parameters

FunctionDefinitionId (p. 571)
  The ID of the function definition.
  where used: path; required: true
  type: string
CLI

```bash
aws greengrass get-function-definition \
  --function-definition-id <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "FunctionDefinitionId": "string"
}
```

Responses

**200 (GetFunctionDefinitionResponse)**

Success.

**DefinitionInformation (p. 594)**

```json
{
  "Name": "string",
  "Id": "string",
  "Arn": "string",
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  },
  "LastUpdatedTimestamp": "string",
  "CreationTimestamp": "string",
  "LatestVersion": "string",
  "LatestVersionArn": "string"
}
```

DefinitionInformation

Information about a definition.

- **Name**
  - type: string
  - The name of the definition.

- **Id**
  - type: string
  - The ID of the definition.

- **Arn**
  - type: string
  - The ARN of the definition.

- **tags**
  - The resource tags.
type: object

additionalProperties: The key-value pair for the resource tag. Type: string

LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.

type: string

LatestVersion

The ID of the latest version associated with the definition.

type: string

LatestVersionArn

The ARN of the latest version associated with the definition.

type: string

400

Invalid request.

**GeneralError (p. 614)**

```json
{
   "Message": "string",
   "ErrorDetails": [
      {
         "DetailedErrorCode": "string",
         "DetailedErrorMessage": "string"
      }
   ]
}
```

**GeneralError**

General error information.

type: object

required: ["Message"]

**Message**

A message that contains information about the error.

type: string

**ErrorDetails**

A list of error details.

type: array

items: **ErrorDetail (p. 600)**

**ErrorDetail**

Details about the error.
**DELETE**

`DELETE /greengrass/definition/functions/FunctionDefinitionId`

**Operation ID:** DeleteFunctionDefinition (p. 97)

Deletes a Lambda function definition.

**Path Parameters**

*FunctionDefinitionId* (p. 571)

The ID of the function definition.

- where used: path; required: true
- type: string

**CLI**

```
aws greengrass delete-function-definition
   --function-definition-id <value> \
   [--cli-input-json <value>] \ 
   [--generate-cli-skeleton]
```

**cli-input-json format:**

```
{
   "FunctionDefinitionId": "string"
}
```

**Responses**

**200**

Success.

**Empty (p. 600)**

```
{
}
```
Empty Schema

Empty
type: object

400

Invalid request.

GeneralError (p. 614)

```
{
  "Message": "string",
  "ErrorDetails": [ 
    { 
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError

General error information.
type: object
required: ["Message"]

Message

A message that contains information about the error.
type: string

ErrorDetails

A list of error details.
type: array
items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.
type: object

DetailedErrorCode

A detailed error code.
type: string

DetailedErrorMessage

A detailed error message.
type: string

PUT

PUT /greengrass/definition/functions/FunctionDefinitionId
Operation ID: UpdateFunctionDefinition (p. 256)

Updates the name of a function definition. To update the list of available functions, use CreateFunctionDefinitionVersion.

Produces: application/json

**Body Parameters**

**UpdateDefinitionRequestBody** (p. 578)

where used: body; required: true

```
{
    "Name": "string"
}
```

**Name**

The name of the definition.

required: true
type: string

**Path Parameters**

**FunctionDefinitionId** (p. 571)

The ID of the function definition.

where used: path; required: true
type: string

**CLI**

```
aws greengrass update-function-definition \
   --function-definition-id <value> \ 
   [--name <value>] \ 
   [--cli-input-json <value>] \ 
   [--generate-cli-skeleton]
```

cli-input-json format:

```
{
    "FunctionDefinitionId": "string",
    "Name": "string"
}
```

**Responses**

**200**

Success.

Empty (p. 600)
Empty Schema

Empty

type: object

400

Invalid request.

GeneralError (p. 614)

{
  "Message": "string",
  "ErrorDetails": [
  {
    "DetailedErrorCode": "string",
    "DetailedErrorMessage": "string"
  }
  ]
}

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string
POST /greengrass/definition/functions/FunctionDefinitionId/versions

POST /greengrass/definition/functions/FunctionDefinitionId/versions

Operation ID: CreateFunctionDefinitionVersion (p. 41)

Creates a version of a Lambda function definition that has already been defined.

Produces: application/json

Body Parameters

CreateFunctionDefinitionVersionRequestBody (p. 554)

Information about the function definition version.

where used: body; required: true

```json
{
  "DefaultConfig": {
    "Execution": {
      "IsolationMode": "GreengrassContainer|NoContainer",
      "RunAs": {
        "Uid": 1001,
        "Gid": 1002
      }
    },
  },
  "Functions": [
    {
      "Id": "string",
      "FunctionArn": "string",
      "FunctionConfiguration": {
        "Pinned": true,
        "Executable": "string",
        "ExecArgs": "string",
        "MemorySize": 0,
        "Timeout": 0,
        "EncodingType": "binary|json",
        "Environment": {
          "Variables": {
            "additionalProperty0": "string",
            "additionalProperty1": "string",
            "additionalProperty2": "string"
          }
        },
        "ResourceAccessPolicies": [
          {
            "ResourceId": "string",
            "Permission": "ro|rw"
          }
        ],
        "AccessSysfs": true,
        "Execution": {
          "IsolationMode": "GreengrassContainer|NoContainer",
          "RunAs": {
            "Uid": 1001,
            "Gid": 1002
          }
        }
```
FunctionDefinitionVersion

Information about a function definition version.

type: object

DefaultConfig

The default configuration that applies to all Lambda functions in the group. Individual Lambda functions can override these settings.

type: object

Execution

Configuration information that specifies how a Lambda function runs.

Functions

A list of Lambda functions in this function definition version.

type: array

items: Function (p. 601)

Information about a Lambda function.

type: object

required: ["Id"]

Id

A descriptive or arbitrary ID for the function. This value must be unique within the function definition version. Maximum length is 128 characters with the pattern `[a-zA-Z0-9_:\_]`.+.

type: string

FunctionArn

The ARN of the alias (recommended) or version of the target Lambda function.

type: string

FunctionConfiguration

The configuration of the Lambda function.

type: object

Pinned

True if the function is pinned. Pinned means the function is long-lived and starts when the core starts.

type: boolean

Executable

The name of the function executable.
type: string

ExecArgs

The execution arguments.

type: string

MemorySize

The memory size, in KB, required by the function. This setting does not apply and should be cleared when you run the Lambda function without containerization.

type: integer

Timeout

The allowed function execution time, after which Lambda should terminate the function. This timeout still applies to pinned Lambda functions for each request.

type: integer

EncodingType

The expected encoding type of the input payload for the function. The default is `json`.

type: string

enum: ["binary", "json"]

Environment

The environment configuration of the function.

type: object

Variables

Environment variables for the Lambda function's configuration.

type: object

additionalProperties: An object with properties of type `string` that represent the environment variables.

ResourceAccessPolicies

A list of the resources, with their permissions, to which the Lambda function is granted access. A Lambda function can have at most 10 resources. ResourceAccessPolicies applies only when you run the Lambda function in a Greengrass container.

type: array

items: ResourceAccessPolicy (p. 660)

ResourceAccessPolicy

A policy used by the function to access a resource.

type: object

required: ["ResourceId"]

ResourceId

The ID of the resource. (This ID is assigned to the resource when you create the resource definition.)
type: string

Permission

The type of permission a function has to access a resource.

type: string

enum: ["ro", "rw"]

AccessSysfs

If true, the Lambda function is allowed to access the host's /sys folder. Use this when the Lambda function needs to read device information from /sys. This setting applies only when you run the Lambda function in a Greengrass container.

type: boolean

Execution

Configuration information that specifies how a Lambda function runs.

type: object

IsolationMode

Specifies whether the Lambda function runs in a Greengrass container (default) or without containerization. Unless your scenario requires that you run without containerization, we recommend that you run in a Greengrass container. Omit this value to run the Lambda function with the default containerization for the group.

type: string

enum: ["GreengrassContainer", "NoContainer"]

RunAs

Specifies the user and group whose permissions are used when running the Lambda function. You can specify one or both values to override the default values. To minimize the risk of unintended changes or malicious attacks, we recommend that you avoid running as root unless absolutely necessary. To run as root, you must update config.json in greengrass-root/config to set allowFunctionsToRunAsRoot to yes.

type: object

Uid

The user ID whose permissions are used to run a Lambda function.

type: integer

Gid

The group ID whose permissions are used to run a Lambda function.

type: integer

Header Parameters

X-Amzn-Client-Token (p. 579)

A client token used to correlate requests and responses.

where used: header; required: false
type: string

Path Parameters

FunctionDefinitionId (p. 571)

The ID of the function definition.

where used: path; required: true

type: string

CLI

aws greengrass create-function-definition-version \
  --function-definition-id <value> \ 
  [--default-config <value>] \ 
  [--functions <value>] \ 
  [--amzn-client-token <value>] \ 
  [--cli-input-json <value>] \ 
  [--generate-cli-skeleton]

cli-input-json format:

```json
{
  "FunctionDefinitionId": "string",
  "DefaultConfig": {
    "Execution": {
      "IsolationMode": "GreengrassContainer|NoContainer",
      "RunAs": {
        "Uid": "integer",
        "Gid": "integer"
      }
    }
  },
  "Functions": [
    {
      "Id": "string",
      "FunctionArn": "string",
      "FunctionConfiguration": {
        "Pinned": "boolean",
        "Executable": "string",
        "ExecArgs": "string",
        "MemorySize": "integer",
        "Timeout": "integer",
        "EncodingType": "binary|json",
        "Environment": {
          "Variables": {
            "additionalProperty0": "string",
            "additionalProperty1": "string",
            "additionalProperty2": "string"
          }
        },
        "ResourceAccessPolicies": [
          {
            "ResourceId": "string",
            "Permission": "ro|rw"
          }
        ],
        "AccessSysfs": "boolean",
        "Execution": {
          "IsolationMode": "GreengrassContainer|NoContainer",
          "RunAs": {
            "Uid": "integer",
            "Gid": "integer"
          }
        }
      }
    }
  ]
}
```
"RunAs": {
    "Uid": "integer",
    "Gid": "integer"
}
}
},
"AmznClientToken": "string"
}

**Responses**

**200 (CreateFunctionDefinitionVersionResponse)**

VersionInformation (p. 676)

```json
{
    "Arn": "string",
    "Id": "string",
    "Version": "string",
    "CreationTimestamp": "string"
}
```

VersionInformation

Information about a version.

type: object

**Arn**

The ARN of the version.

type: string

**Id**

The ID of the parent definition that the version is associated with.

type: string

**Version**

The ID of the version.

type: string

**CreationTimestamp**

The time, in milliseconds since the epoch, when the version was created.

type: string

**400**

Invalid request.

**GeneralError (p. 614)**

```json
{
    "Message": "string",
    "ErrorDetails": []
}
```
GET

GET /greengrass/definition/functions/FunctionDefinitionId/versions

Operation ID: ListFunctionDefinitionVersions (p. 206)

Lists the versions of a Lambda function definition.

Produces: application/json

Path Parameters

FunctionDefinitionId (p. 571)

The ID of the function definition.

where used: path; required: true
**Type**: string

**Query Parameters**

**NextToken (p. 573)**

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

where used: query; required: false

**MaxResults (p. 573)**

The maximum number of results to be returned per request.

where used: query; required: false

**CLI**

```bash
aws greengrass list-function-definition-versions \
  --function-definition-id <value> \
  [--next-token <value>] \
  [--max-results <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "FunctionDefinitionId": "string",
  "NextToken": "string",
  "MaxResults": "integer"
}
```

**Responses**

**200 (ListFunctionDefinitionVersionsResponse)**

**ListVersionsResponse (p. 650)**

```json
{
  "Versions": [
  {
    "Arn": "string",
    "Id": "string",
    "Version": "string",
    "CreationTimestamp": "string"
  },
  ],
  "NextToken": "string"
}
```

**ListVersionsResponse**

A list of versions.
type: object

Versions

Information about a version.

type: array

items: VersionInformation (p. 676)

VersionInformation

Information about a version.

type: object

Arn

The ARN of the version.

type: string

Id

The ID of the parent definition that the version is associated with.

type: string

Version

The ID of the version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the version was created.

type: string

NextToken

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

type: string

400

Invalid request.

GeneralError (p. 614)

{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}

GeneralError

General error information.

type: object
required: ["Message"]

Message
A message that contains information about the error.
type: string

ErrorDetails
A list of error details.
type: array
items: ErrorDetail (p. 600)

ErrorDetail
Details about the error.
type: object

DetailedErrorCode
A detailed error code.
type: string

DetailedErrorMessage
A detailed error message.
type: string

GET /greengrass/definition/functions/FunctionDefinitionId/versions/FunctionDefinitionVersionId

GET /greengrass/definition/functions/FunctionDefinitionId/versions/FunctionDefinitionVersionId

Operation ID: GetFunctionDefinitionVersion (p. 142)

Retrieves information about a Lambda function definition version, including which Lambda functions are included in the version and their configurations.

Produces: application/json

Path Parameters

FunctionDefinitionId (p. 571)
The ID of the function definition.
where used: path; required: true
type: string
**FunctionDefinitionVersionId** (p. 571)

The ID of the function definition version. This value maps to the `Version` property of the corresponding `VersionInformation` object, which is returned by `ListFunctionDefinitionVersions` requests. If the version is the last one that was associated with a function definition, the value also maps to the `LatestVersion` property of the corresponding `DefinitionInformation` object.

where used: path; required: true
type: string

**Query Parameters**

**NextToken** (p. 573)

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

where used: query; required: false
type: string

**CLI**

```bash
aws greengrass get-function-definition-version
   --function-definition-id <value> \
   --function-definition-version-id <value> \
   [--next-token <value>] \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "FunctionDefinitionId": "string",
   "FunctionDefinitionVersionId": "string",
   "NextToken": "string"
}
```

**Responses**

**200**

Success.

**GetFunctionDefinitionVersionResponse** (p. 624)

```json
{
   "Arn": "string",
   "Id": "string",
   "Version": "string",
   "CreationTimestamp": "string",
   "Definition": {
      "DefaultConfig": {
         "Execution": {
            "IsolationMode": "GreengrassContainer|NoContainer",
            "RunAs": {
               "Uid": 1001,
```
"Gid": 1002
},
"Functions": [
{
"Id": "string",
"FunctionArn": "string",
"FunctionConfiguration": {
"Pinned": true,
"Executable": "string",
"ExecArgs": "string",
"MemorySize": 0,
"Timeout": 0,
"EncodingType": "binary|json",
"Environment": {
"Variables": {
"additionalProperty0": "string",
"additionalProperty1": "string",
"additionalProperty2": "string"
},
"ResourceAccessPolicies": [
{
"ResourceId": "string",
"Permission": "ro|rw"
}
],
"AccessSysfs": true,
"Execution": {
"IsolationMode": "GreengrassContainer|NoContainer",
"RunAs": {
"Uid": 1001,
"Gid": 1002
}
}
}
}
"NextToken": "string"
}

GetFunctionDefinitionVersionResponse
Information about a function definition version.
type: object
Arn
The ARN of the function definition version.
type: string
Id
The ID of the function definition version.
type: string
Version
The version of the function definition version.
type: string
CreationTimestamp
   The time, in milliseconds since the epoch, when the function definition version was created.
   type: string

Definition
   Information about a function definition version.
   type: object

DefaultConfig
   The default configuration that applies to all Lambda functions in the group. Individual Lambda
   functions can override these settings.
   type: object

Execution
   Configuration information that specifies how a Lambda function runs.

Functions
   A list of Lambda functions in this function definition version.
   type: array
   items: Function (p. 601)

Id
   A descriptive or arbitrary ID for the function. This value must be unique within the function
   definition version. Maximum length is 128 characters with the pattern `[a-zA-Z0-9_:.-]+`.
   type: string

FunctionArn
   The ARN of the alias (recommended) or version of the target Lambda function.
   type: string

FunctionConfiguration
   The configuration of the Lambda function.
   type: object

Pinned
   True if the function is pinned. Pinned means the function is long-lived and starts when the core
   starts.
   type: boolean

Executable
   The name of the function executable.
type: string

ExecArgs

The execution arguments.

type: string

MemorySize

The memory size, in KB, required by the function. This setting does not apply and should be cleared when you run the Lambda function without containerization.

type: integer

Timeout

The allowed function execution time, after which Lambda should terminate the function. This timeout still applies to pinned Lambda functions for each request.

type: integer

EncodingType

The expected encoding type of the input payload for the function. The default is `json`.

type: string

default: "json"

enum: ["binary", "json"]

Environment

The environment configuration of the function.

type: object

Variables

Environment variables for the Lambda function's configuration.

type: object

additionalProperties: An object with properties of type `string` that represent the environment variables.

ResourceAccessPolicies

A list of the resources, with their permissions, to which the Lambda function is granted access. A Lambda function can have at most 10 resources. ResourceAccessPolicies applies only when you run the Lambda function in a Greengrass container.

type: array

items: ResourceAccessPolicy (p. 660)

ResourceAccessPolicy

A policy used by the function to access a resource.

type: object

required: ["ResourceId"]

ResourceId

The ID of the resource. (This ID is assigned to the resource when you create the resource definition.)
type: string
Permission

The type of permission a function has to access a resource.

type: string
def: ["ro", "rw"]
AccessSysfs

If true, the Lambda function is allowed to access the host's /sys folder. Use this when the Lambda function needs to read device information from /sys. This setting applies only when you run the Lambda function in a Greengrass container.

type: boolean
Execution

Configuration information that specifies how a Lambda function runs.

type: object
IsolationMode

Specifies whether the Lambda function runs in a Greengrass container (default) or without containerization. Unless your scenario requires that you run without containerization, we recommend that you run in a Greengrass container. Omit this value to run the Lambda function with the default containerization for the group.

type: string
def: ["GreengrassContainer", "NoContainer"]
RunAs

Specifies the user and group whose permissions are used when running the Lambda function. You can specify one or both values to override the default values. To minimize the risk of unintended changes or malicious attacks, we recommend that you avoid running as root unless absolutely necessary. To run as root, you must update config.json in greengrass-root/config to set allowFunctionsToRunAsRoot to yes.

type: object
Uid

The user ID whose permissions are used to run a Lambda function.

type: integer
Gid

The group ID whose permissions are used to run a Lambda function.

type: integer
NextToken

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

type: string
400

Invalid request.
GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

**GeneralError**

General error information.

type: object

required: ["Message"]

**Message**

A message that contains information about the error.

type: string

**ErrorDetails**

A list of error details.

type: array

items: ErrorDetail (p. 600)

**ErrorDetail**

Details about the error.

type: object

**DetailedErrorCode**

A detailed error code.

type: string

**DetailedErrorMessage**

A detailed error message.

type: string

/greengrass/definition/loggers

**GET**

GET /greengrass/definition/loggers

Operation ID: ListLoggerDefinitions (p. 222)

Retrieves a list of logger definitions.
Produce: application/json

Query Parameters

MaxResults (p. 573)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

NextToken (p. 573)

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

where used: query; required: false

type: string

CLI

aws greengrass list-logger-definitions 
  [--max-results <value>] 
  [--next-token <value>] 
  [--cli-input-json <value>] 
  [--generate-cli-skeleton]

cli-input-json format:

{
    "MaxResults": "integer",
    "NextToken": "string"
}

Responses

200 (ListLoggerDefinitionsResponse)

ListDefinitionsResponse (p. 645)

{
    "Definitions": [
        {
            "Name": "string",
            "Id": "string",
            "Arn": "string",
            "tags": {
                "additionalProperty0": "string",
                "additionalProperty1": "string",
                "additionalProperty2": "string"
            },
            "LastUpdatedTimestamp": "string",
            "CreationTimestamp": "string",
            "LatestVersion": "string",
            "LatestVersionArn": "string"
        }
    ],
    "NextToken": "string"
}
ListDefinitionsResponse

A list of definitions.

type: object

Definitions

Information about a definition.

type: array

items: DefinitionInformation (p. 594)

DefinitionInformation

Information about a definition.

type: object

Name

The name of the definition.

type: string

Id

The ID of the definition.

type: string

Arn

The ARN of the definition.

type: string

tags

The resource tags.

type: object

additionalProperties: The key-value pair for the resource tag. Type: string

LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.

type: string

LatestVersion

The ID of the latest version associated with the definition.

type: string

LatestVersionArn

The ARN of the latest version associated with the definition.

type: string
NextToken

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

type: string

POST

POST /greengrass/definition/loggers

Operation ID: CreateLoggerDefinition (p. 57)

Creates a logger definition. You can provide the initial version of the logger definition now or use CreateLoggerDefinitionVersion later.

Produces: application/json

Body Parameters

CreateLoggerDefinitionRequestBody (p. 559)

where used: body; required: true

```
{
   "Name": "string",
   "InitialVersion": {
      "Loggers": [
         {
            "Id": "string",
            "Type": "FileSystem|AWSCloudWatch",
            "Component": "GreengrassSystem|Lambda",
            "Level": "DEBUG|INFO|WARN|ERROR|FATAL",
            "Space": 0
         }
      ],
      "tags": {
         "keyName0": "string",
         "keyName1": "string",
         "keyName2": "string"
      }
   }
}
```

Name

The name of the logger definition.

type: string

InitialVersion

Information about a logger definition version.

type: object

Loggers

A list of loggers.

type: array

items: Logger (p. 652)
Logger

Information about a logger
type: object
required: ["Id", "Type", "Component", "Level"]

Id
A descriptive or arbitrary ID for the logger. This value must be unique within the logger
definition version. Maximum length is 128 characters with the pattern \[a-zA-Z0-9:_-]+\.
type: string

Type
type: string
enum: ["FileSystem", "AWSCloudWatch"]

Component
type: string
enum: ["GreengrassSystem", "Lambda"]

Level
type: string
enum: ["DEBUG", "INFO", "WARN", "ERROR", "FATAL"]

Space
The amount of file space, in KB, to use if the local file system is used for logging purposes.
type: integer
tags
The resource tags.
type: object
additionalProperties: The key-value pair for the resource tag. Type: string

Header Parameters

X-Amzn-Client-Token (p. 579)
A client token used to correlate requests and responses.
where used: header; required: false
type: string

CLI

```
aws greengrass create-logger-definition \
    [ --name <value> ] \
    [ --initial-version <value> ] \
    [ --tags <value> ] \
    [ --amzn-client-token <value> ] \
```
cli-input-json format:

```json
{
  "Name": "string",
  "InitialVersion": {
    "Loggers": [
      {
        "Id": "string",
        "Type": "FileSystem|AWSCloudWatch",
        "Component": "GreengrassSystem|Lambda",
        "Level": "DEBUG|INFO|WARN|ERROR|FATAL",
        "Space": "integer"
      }
    ],
    "tags": {
      "additionalProperty0": "string",
      "additionalProperty1": "string",
      "additionalProperty2": "string"
    },
    "AmznClientToken": "string"
  }
}
```

**Responses**

**200** *(CreateLoggerDefinitionResponse)*

**DefinitionInformation (p. 594)**

```json
{
  "Name": "string",
  "Id": "string",
  "Arn": "string",
  "LastUpdatedTimestamp": "string",
  "CreationTimestamp": "string",
  "LatestVersion": "string",
  "LatestVersionArn": "string"
}
```

**DefinitionInformation**

Information about a definition.

- **Name**
  - type: object
  - **Name**
    - The name of the definition.
  - type: string

- **Id**
  - The ID of the definition.
  - type: string

- **Arn**
  - The ARN of the definition.
type: string
LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.

type: string
CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.

type: string
LatestVersion

The ID of the latest version associated with the definition.

type: string
LatestVersionArn

The ARN of the latest version associated with the definition.

type: string

400

Invalid request.

GeneralError (p. 614)

{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.
/greengrass/definition/loggers/LoggerDefinitionId

GET

GET /greengrass/definition/loggers/LoggerDefinitionId

Operation ID: GetLoggerDefinition (p. 160)

Retrieves information about a logger definition.

Produces: application/json

Path Parameters

LoggerDefinitionId (p. 572)

The ID of the logger definition.

where used: path; required: true

type: string

 CLI

aws greengrass get-logger-definition \
  --logger-definition-id <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]

cli-input-json format:

```json
{
  "LoggerDefinitionId": "string"
}
```

Responses

200 (GetLoggerDefinitionResponse)

DefinitionInformation (p. 594)

```json
{
  "Name": "string",
```
"Id": "string",
"Arn": "string",
"tags": {
  "additionalProperty0": "string",
  "additionalProperty1": "string",
  "additionalProperty2": "string"
},
"LastUpdatedTimestamp": "string",
"CreationTimestamp": "string",
"LatestVersion": "string",
"LatestVersionArn": "string"
}

DefinitionInformation

Information about a definition.

type: object

Name

The name of the definition.

type: string

Id

The ID of the definition.

type: string

Arn

The ARN of the definition.

type: string
tags

The resource tags.

type: object

additionalProperties: The key-value pair for the resource tag. Type: string

LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.

type: string

LatestVersion

The ID of the latest version associated with the definition.

type: string

LatestVersionArn

The ARN of the latest version associated with the definition.

type: string
400
Invalid request.

GeneralError (p. 614)

```
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError
General error information.
type: object
required: ["Message"]

Message
A message that contains information about the error.
type: string

ErrorDetails
A list of error details.
type: array
items: ErrorDetail (p. 600)

ErrorDetail
Details about the error.
type: object

DetailedErrorCode
A detailed error code.
type: string

DetailedErrorMessage
A detailed error message.
type: string

DELETE
DELETE /greengrass/definition/loggers/LoggerDefinitionId

Operation ID: DeleteLoggerDefinition (p. 100)

Deletes a logger definition.

Produces: application/json
Path Parameters

**LoggerDefinitionId (p. 572)**

The ID of the logger definition.

where used: path; required: true
type: string

CLI

```bash
aws greengrass delete-logger-definition \
  --logger-definition-id <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "LoggerDefinitionId": "string"
}
```

Responses

200

Success.

Empty (p. 600)

```json
{}
```

Empty Schema

Empty
type: object

400

Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError

General error information.
type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

**PUT**

PUT /greengrass/definition/loggers/`LoggerDefinitionId`

Operation ID: `UpdateLoggerDefinition (p. 263)`

Updates the name of a logger definition. To update the list of available loggers, use `CreateLoggerDefinitionVersion`.

Produces: application/json

**Body Parameters**

`UpdateDefinitionRequestBody (p. 578)`

where used: body; required: true

```
{
   "Name": "string"
}
```

Name

The name of the definition.

required: true

type: string
Path Parameters

**LoggerDefinitionId (p. 572)**

The ID of the logger definition.

where used: path; required: true
type: string

CLI

```bash
aws greengrass update-logger-definition \
   --logger-definition-id <value> \
   [--name <value>]  \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "LoggerDefinitionId": "string",
   "Name": "string"
}
```

Responses

**200**

Success.

Empty (p. 600)

```json
{}
```

Empty Schema

Empty
type: object

**400**

Invalid request.

GeneralError (p. 614)

```json
{
   "Message": "string",
   "ErrorDetails": [
   {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
   }
   ]
}
```
GeneralError

- General error information.
  - type: object
  - required: ["Message"]
Message

- A message that contains information about the error.
  - type: string
ErrorDetails

- A list of error details.
  - type: array
  - items: ErrorDetail (p. 600)
ErrorDetail

- Details about the error.
  - type: object
DetailedErrorCode

- A detailed error code.
  - type: string
DetailedErrorMessage

- A detailed error message.
  - type: string

/greengrass/definition/loggers/LoggerDefinitionId/versions

POST

POST /greengrass/definition/loggers/LoggerDefinitionId/versions

Operation ID: CreateLoggerDefinitionVersion (p. 62)

Creates a version of a logger definition that has already been defined.

Produces: application/json

Body Parameters

CreateLoggerDefinitionVersionRequestBody (p. 560)

where used: body; required: true

```json
{
  "Loggers": [
    
```
"Id": "string",
"Type": "FileSystem|AWSCloudWatch",
"Component": "GreengrassSystem|Lambda",
"Level": "DEBUG|INFO|WARN|ERROR|FATAL",
"Space": 0
}
]

LoggerDefinitionVersion

Information about a logger definition version.

type: object

Loggers

A list of loggers.

type: array

items: Logger (p. 652)

Logger

Information about a logger

type: object

required: ["Id", "Type", "Component", "Level"]

Id

A descriptive or arbitrary ID for the logger. This value must be unique within the logger
definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9_:.-]+.

type: string

Type

type: string

enum: ["FileSystem", "AWSCloudWatch"]

Component

type: string

enum: ["GreengrassSystem", "Lambda"]

Level

type: string

enum: ["DEBUG", "INFO", "WARN", "ERROR", "FATAL"]

Space

The amount of file space, in KB, to use if the local file system is used for logging purposes.

type: integer

Header Parameters

**X-Amzn-Client-Token (p. 579)**

A client token used to correlate requests and responses.
Path Parameters

**LoggerDefinitionId (p. 572)**

The ID of the logger definition.

where used: path; required: true
type: string

CLI

```bash/aws greengrass create-logger-definition-version \
   --logger-definition-id <value> \
   [--loggers <value>] \
   [--amzn-client-token <value>] \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "LoggerDefinitionId": "string",
   "Loggers": [
      {
         "Id": "string",
         "Type": "FileSystem|AWSCloudWatch",
         "Component": "GreengrassSystem|Lambda",
         "Level": "DEBUG|INFO|WARN|ERROR|FATAL",
         "Space": "integer"
      }
   ],
   "AmznClientToken": "string"
}
```

Responses

**200 (CreateLoggerDefinitionVersionResponse)**

```json
{
   "Arn": "string",
   "Id": "string",
   "Version": "string",
   "CreationTimestamp": "string"
}
```

Information about a version.
type: object
Arn

The ARN of the version.

type: string

Id

The ID of the parent definition that the version is associated with.

type: string

Version

The ID of the version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the version was created.

type: string

400

Invalid request.

**GeneralError (p. 614)**

```
{
    "Message": "string",
    "ErrorDetails": [
    {
        "DetailedErrorCode": "string",
        "DetailedErrorMessage": "string"
    }
    ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: **ErrorDetail (p. 600)**

ErrorDetail

Details about the error.
type: object
DetailedErrorCode
  A detailed error code.
  type: string
DetailedErrorMessage
  A detailed error message.
  type: string

GET
GET /greengrass/definition/loggers/{LoggerDefinitionId}/versions

Operation ID: ListLoggerDefinitionVersions (p. 219)

Lists the versions of a logger definition.

Produces: application/json

Path Parameters

LoggerDefinitionId (p. 572)
  The ID of the logger definition.
  where used: path; required: true
  type: string

Query Parameters

NextToken (p. 573)
  The token for the next set of results, or null if there are no more results.
  where used: query; required: false
  type: string

MaxResults (p. 573)
  The maximum number of results to be returned per request.
  where used: query; required: false
  type: integer

CLI

aws greengrass list-logger-definition-versions
  --logger-definition-id <value> \
  [--next-token <value>] \
  [--max-results <value>] \
  [--cli-input-json <value>] \

401
cli-input-json format:

```json
{
  "LoggerDefinitionId": "string",
  "NextToken": "string",
  "MaxResults": "integer"
}
```

### Responses

**200** ([ListLoggerDefinitionVersionsResponse](#))

ListVersionsResponse (p. 650)

```json
{
  "Versions": [
    {
      "Arn": "string",
      "Id": "string",
      "Version": "string",
      "CreationTimestamp": "string"
    }
  ],
  "NextToken": "string"
}
```

ListVersionsResponse

- A list of versions.
- type: object

**Versions**

- Information about a version.
- type: array

  **items**: VersionInformation (p. 676)

VersionInformation

- Information about a version.
- type: object

**Arn**

- The ARN of the version.
- type: string

**Id**

- The ID of the parent definition that the version is associated with.
- type: string

**Version**

- The ID of the version.
type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the version was created.

type: string

NextToken

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

type: string

400

Invalid request.

GeneralError (p. 614)

```
{
  "Message": "string",
  "ErrorDetails": [
  {
    "DetailedErrorCode": "string",
    "DetailedErrorMessage": "string"
  }
  ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.
type: string

/greengrass/definition/loggers/LoggerDefinitionId/
versions/LoggerDefinitionVersionId

GET

GET /greengrass/definition/loggers/LoggerDefinitionId/
versions/LoggerDefinitionVersionId

Operation ID: GetLoggerDefinitionVersion (p. 162)
Retrieves information about a logger definition version.
Produces: application/json

Path Parameters

LoggerDefinitionVersionId (p. 572)
The ID of the logger definition version. This value maps to the Version property of the corresponding VersionInformation object, which is returned by ListLoggerDefinitionVersions requests. If the version is the last one that was associated with a logger definition, the value also maps to the LatestVersion property of the corresponding DefinitionInformation object.

where used: path; required: true
type: string

LoggerDefinitionId (p. 572)
The ID of the logger definition.
where used: path; required: true
type: string

Query Parameters

NextToken (p. 573)
The token for the next set of results, or null if there are no more results.
where used: query; required: false
type: string

CLI

```
aws greengrass get-logger-definition-version 
  --logger-definition-version-id <value> 
  --logger-definition-id <value> 
  [--next-token <value>] 
  [--cli-input-json <value>] 
  [--generate-cli-skeleton]
```

404
cli-input-json format:

```json
{
    "LoggerDefinitionVersionId": "string",
    "LoggerDefinitionId": "string",
    "NextToken": "string"
}
```

**Responses**

**200** *(GetLoggerDefinitionVersionResponse)*

Success.

GetLoggerDefinitionVersionResponse (p. 630)

```json
{
    "Arn": "string",
    "Id": "string",
    "Version": "string",
    "CreationTimestamp": "string",
    "Definition": {
        "Loggers": [
            {
                "Id": "string",
                "Type": "FileSystem|AWSCloudWatch",
                "Component": "GreengrassSystem|Lambda",
                "Level": "DEBUG|INFO|WARN|ERROR|FATAL",
                "Space": 0
            }
        ]
    }
}
```

GetLoggerDefinitionVersionResponse

Information about a logger definition version.

type: object

**Arn**

The ARN of the logger definition version.

type: string

**Id**

The ID of the logger definition version.

type: string

**Version**

The version of the logger definition version.

type: string

**CreationTimestamp**

The time, in milliseconds since the epoch, when the logger definition version was created.

type: string
Definition

Information about a logger definition version.

type: object

Loggers

A list of loggers.

type: array

items: Logger (p. 652)

Logger

Information about a logger

type: object

required: ["Id", "Type", "Component", "Level"]

Id

A descriptive or arbitrary ID for the logger. This value must be unique within the logger definition version. Maximum length is 128 characters with the pattern [ a-zA-Z0-9_:-]+.

type: string

Type

type: string

enum: ["FileSystem", "AWSCloudWatch"]

Component

type: string

enum: ["GreengrassSystem", "Lambda"]

Level

type: string

enum: ["DEBUG", "INFO", "WARN", "ERROR", "FATAL"]

Space

The amount of file space, in KB, to use if the local file system is used for logging purposes.

type: integer

400

Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```
GeneralError
  General error information.
  type: object
  required: ["Message"]
Message
  A message that contains information about the error.
  type: string
ErrorDetails
  A list of error details.
  type: array
  items: ErrorDetail (p. 600)
ErrorDetail
  Details about the error.
  type: object
DetailedErrorCode
  A detailed error code.
  type: string
DetailedErrorMessage
  A detailed error message.
  type: string

/greengrass/definition/resources

GET
GET /greengrass/definition/resources
Operation ID: ListResourceDefinitions (p. 227)
Retrieves a list of resource definitions.
Produces: application/json
Query Parameters
MaxResults (p. 573)
  The maximum number of results to be returned per request.
  where used: query; required: false
  type: integer
NextToken (p. 573)
  The token for the next set of results, or null if there are no more results.
where used: query; required: false
type: string

CLI

```bash
aws greengrass list-resource-definitions
  [--max-results <value>] 
  [--next-token <value>] 
  [--cli-input-json <value>] 
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "MaxResults": "integer",
  "NextToken": "string"
}
```

Responses

200 (ListResourceDefinitionsResponse)

The IDs of all the Greengrass resource definitions in this account.

ListDefinitionsResponse (p. 645)

```json
{
  "Definitions": [
    {
      "Name": "string",
      "Id": "string",
      "Arn": "string",
      "tags": {
        "additionalProperty0": "string",
        "additionalProperty1": "string",
        "additionalProperty2": "string"
      },
      "LastUpdatedTimestamp": "string",
      "CreationTimestamp": "string",
      "LatestVersion": "string",
      "LatestVersionArn": "string"
    }
  ],
  "NextToken": "string"
}
```

ListDefinitionsResponse

A list of definitions.
type: object
Definitions

Information about a definition.
type: array
items: DefinitionInformation (p. 594)
DefinitionInformation

Information about a definition.

type: object

Name

The name of the definition.

type: string

Id

The ID of the definition.

type: string

Arn

The ARN of the definition.

type: string

tags

The resource tags.

type: object

additionalProperties: The key-value pair for the resource tag. Type: string

LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.

type: string

LatestVersion

The ID of the latest version associated with the definition.

type: string

LatestVersionArn

The ARN of the latest version associated with the definition.

type: string

NextToken

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

type: string

POST

POST /greengrass/definition/resources
Operation ID: CreateResourceDefinition (p. 65)

Creates a resource definition that contains a list of resources to be used in a group. You can create an initial version of the definition by providing a list of resources now, or use CreateResourceDefinitionVersion later.

Produces: application/json

Body Parameters

CreateResourceDefinitionRequestBody (p. 561)

where used: body; required: true

```json
{
    "Name": "string",
    "InitialVersion": {
        "Resources": [
            {
                "Id": "string",
                "Name": "string",
                "ResourceDataContainer": {
                    "**-- NOTE:": "choose one of the following --**",
                    "LocalDeviceResourceData": {
                        "SourcePath": "string",
                        "GroupOwnerSetting": {
                            "AutoAddGroupOwner": true,
                            "GroupOwner": "string"
                        }
                    },
                    "LocalVolumeResourceData": {
                        "SourcePath": "string",
                        "DestinationPath": "string",
                        "GroupOwnerSetting": {
                            "AutoAddGroupOwner": true,
                            "GroupOwner": "string"
                        }
                    },
                    "SageMakerMachineLearningModelResourceData": {
                        "SageMakerJobArn": "string",
                        "DestinationPath": "string",
                        "OwnerSetting": {
                            "GroupOwner": "string",
                            "GroupPermission": "ro|rw"
                        }
                    },
                    "S3MachineLearningModelResourceData": {
                        "S3Uri": "string",
                        "DestinationPath": "string",
                        "OwnerSetting": {
                            "GroupOwner": "string",
                            "GroupPermission": "ro|rw"
                        }
                    },
                    "SecretsManagerSecretResourceData": {
                        "ARN": "string",
                        "AdditionalStagingLabelsToDownload": ["string"
                    ]
                }
            }
        ]
    }
}
```
"tags": {
  "keyName0": "string",
  "keyName1": "string",
  "keyName2": "string"
}

Name

The name of the resource definition.

type: string

InitialVersion

Information about a resource definition version.

type: object

Resources

A list of resources.

type: array

items: Resource (p. 656)

Resource

Information about a resource.

type: object

required: ["Id", "Name", "ResourceDataContainer"]

Id

The resource ID, used to refer to a resource in the Lambda function configuration. Maximum length is 128 characters with the pattern [a-zA-Z0-9_-.]+. This must be unique within a Greengrass group.

type: string

Name

The descriptive resource name, which is displayed on the AWS IoT Greengrass console. Maximum length is 128 characters with the pattern [a-zA-Z0-9_-.]+. This must be unique within a Greengrass group.

type: string

ResourceDataContainer

A container for resource data. The container takes only one of the following supported resource data types: LocalDeviceResourceData, LocalVolumeResourceData, SageMakerMachineLearningModelResourceData, S3MachineLearningModelResourceData, SecretsManagerSecretResourceData.


type: object

NOTE: oneOf

LocalDeviceResourceData

Attributes that define a local device resource.
type: object

SourcePath

The local absolute path of the device resource. The source path for a device resource can refer only to a character device or block device under /dev.

type: string

GroupOwnerSetting

Group owner related settings for local resources.

type: object

AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.

type: boolean

GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.

type: string

LocalVolumeResourceData

Attributes that define a local volume resource.

type: object

SourcePath

The local absolute path of the volume resource on the host. The source path for a volume resource type cannot start with /sys.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

GroupOwnerSetting

Group owner related settings for local resources.

type: object

AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.

type: boolean

GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.
SageMakerMachineLearningModelResourceData

Attributes that define an Amazon SageMaker machine learning resource.

type: object

SageMakerJobArn

The ARN of the Amazon SageMaker training job that represents the source model.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.

type: object

required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are \texttt{rw} (read-write) or \texttt{ro} (read-only).

type: string

enum: ["ro", "rw"]

S3MachineLearningModelResourceData

Attributes that define an Amazon S3 machine learning resource.

type: object

S3Uri

The URI of the source model in an S3 bucket. The model package must be in \texttt{tar.gz} or \texttt{.zip} format.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.

type: object
required: ["GroupOwner", "GroupPermission"]

**GroupOwner**

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

**GroupPermission**

The permissions that the group owner has to the resource. Valid values are `rw` (read-write) or `ro` (read-only).

type: string

elem: ["ro", "rw"]

**SecretsManagerSecretResourceData**

Attributes that define a secret resource, which references a secret from AWS Secrets Manager. AWS IoT Greengrass stores a local, encrypted copy of the secret on the Greengrass core, where it can be securely accessed by connectors and Lambda functions.

type: object

**ARN**

The ARN of the Secrets Manager secret to make available on the core. The value of the secret's latest version (represented by the `AWSCURRENT` staging label) is included by default.

type: string

**AdditionalStagingLabelsToDownload**

Optional. The staging labels in addition to `AWSCURRENT` whose values you want to make available on the core.

type: array
tags

The resource tags.

type: object
additionalProperties: The key-value pair for the resource tag. Type: string

**Header Parameters**

**X-Amzn-Client-Token (p. 579)**

A client token used to correlate requests and responses.

where used: header; required: false

type: string

**CLI**

```
aws greengrass create-resource-definition \\
  [--name <value>] \\
  [--initial-version <value>]
```


```json
{
   "Name": "string",
   "InitialVersion": {
      "Resources": [
         {
            "Id": "string",
            "Name": "string",
            "ResourceDataContainer": {
               "-- NOTE:": "choose one of the following --*",
               "LocalDeviceResourceData": {
                  "SourcePath": "string",
                  "GroupOwnerSetting": {
                     "AutoAddGroupOwner": "boolean",
                     "GroupOwner": "string"
                  }
               },
               "LocalVolumeResourceData": {
                  "SourcePath": "string",
                  "DestinationPath": "string",
                  "GroupOwnerSetting": {
                     "AutoAddGroupOwner": "boolean",
                     "GroupOwner": "string"
                  }
               },
               "SageMakerMachineLearningModelResourceData": {
                  "SageMakerJobArn": "string",
                  "DestinationPath": "string",
                  "OwnerSetting": {
                     "GroupOwner": "string",
                     "GroupPermission": "ro|rw"
                  }
               },
               "S3MachineLearningModelResourceData": {
                  "S3Uri": "string",
                  "DestinationPath": "string",
                  "OwnerSetting": {
                     "GroupOwner": "string",
                     "GroupPermission": "ro|rw"
                  }
               },
               "SecretsManagerSecretResourceData": {
                  "ARN": "string",
                  "AdditionalStagingLabelsToDownload": ["string"
               ]
               
            }
         },
         "tags": {
            "additionalProperty0": "string",
            "additionalProperty1": "string",
            "additionalProperty2": "string"
         },
         "AmznClientToken": "string"
      }
   }
}
```
Responses

200 (CreateResourceDefinitionResponse)

DefinitionInformation (p. 594)

```
{
    "Name": "string",
    "Id": "string",
    "Arn": "string",
    "LastUpdatedTimestamp": "string",
    "CreationTimestamp": "string",
    "LatestVersion": "string",
    "LatestVersionArn": "string"
}
```

DefinitionInformation

Information about a definition.

type: object

Name

The name of the definition.

type: string

Id

The ID of the definition.

type: string

Arn

The ARN of the definition.

type: string

LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.

type: string

LatestVersion

The ID of the latest version associated with the definition.

type: string

LatestVersionArn

The ARN of the latest version associated with the definition.

type: string

400

Invalid request.
### GeneralError (p. 614)

```json
{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}
```

**GeneralError**

General error information.

- **type**: object
- **required**: ["Message"]

**Message**

A message that contains information about the error.

- **type**: string

**ErrorDetails**

A list of error details.

- **type**: array
- **items**: [ErrorDetail (p. 600)]

### ErrorDetail (p. 600)

**ErrorDetail**

Details about the error.

- **type**: object

#### DetailedErrorCode

A detailed error code.

- **type**: string

#### DetailedErrorMessage

A detailed error message.

- **type**: string

### /greengrass/definition/resources/ResourceDefinitionId

**GET**

**GET /greengrass/definition/resources/ResourceDefinitionId**

**Operation ID**: GetResourceDefinition (p. 166)

Retrieves information about a resource definition, including its creation time and latest version.
Produces: application/json

**Path Parameters**

**ResourceDefinitionId (p. 574)**

The ID of the resource definition.

where used: path; required: true

type: string

**CLI**

```bash
aws greengrass get-resource-definition \
  --resource-definition-id <value>  \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "ResourceDefinitionId": "string"
}
```

**Responses**

**200 (GetResourceDefinitionResponse)**

Success.

**DefinitionInformation (p. 594)**

```json
{
  "Name": "string",
  "Id": "string",
  "Arn": "string",
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  },
  "LastUpdatedTimestamp": "string",
  "CreationTimestamp": "string",
  "LatestVersion": "string",
  "LatestVersionArn": "string"
}
```

**DefinitionInformation**

Information about a definition.

type: object

**Name**

The name of the definition.

type: string
Id
The ID of the definition.
type: string
Arn
The ARN of the definition.
type: string
tags
The resource tags.
type: object
  additionalProperties: The key-value pair for the resource tag. Type: string
LastUpdatedTimestamp
The time, in milliseconds since the epoch, when the definition was last updated.
type: string
CreationTimestamp
The time, in milliseconds since the epoch, when the definition was created.
type: string
LatestVersion
The ID of the latest version associated with the definition.
type: string
LatestVersionArn
The ARN of the latest version associated with the definition.
type: string

400
Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError
General error information.
type: object
required: ["Message"]
Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

DELETE

DELETE /greengrass/definition/resources/ResourceDefinitionId

Operation ID: DeleteResourceDefinition (p. 102)

Deletes a resource definition.

Produces: application/json

Path Parameters

ResourceDefinitionId (p. 574)

The ID of the resource definition.

where used: path; required: true

type: string

CLI

```
aws greengrass delete-resource-definition \
  --resource-definition-id <value> \ 
  [--cli-input-json <value>] \ 
  [--generate-cli-skeleton]
```

cli-input-json format:
```json
{
  "ResourceDefinitionId": "string"
}
```

**Responses**

**200**

Success.

Empty (p. 600)

```
{}
```

Empty Schema

Empty
type: object

**400**

Invalid request.

GeneralError (p. 614)

```
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError

General error information.
type: object

required: ["Message"]

Message

A message that contains information about the error.
type: string

ErrorDetails

A list of error details.
type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.
PUT /greengrass/definition/resources/ResourceDefinitionId

Operation ID: UpdateResourceDefinition (p. 265)

Updates the name of a resource definition. To update the list of available resources, use CreateResourceDefinitionVersion.

Produces: application/json

Body Parameters

UpdateDefinitionRequestBody (p. 578)

where used: body; required: true

```json
{
   "Name": "string"
}
```

Name

The name of the definition.

required: true
type: string

Path Parameters

ResourceDefinitionId (p. 574)

The ID of the resource definition.

where used: path; required: true
type: string

CLI

```
aws greengrass update-resource-definition \
   --resource-definition-id <value> \ 
   [--name <value>]  \
```
cli-input-json format:

```json
{
  "ResourceDefinitionId": "string",
  "Name": "string"
}
```

**Responses**

**200**

Success.

Empty (p. 600)

```
{
}
```

Empty Schema

Empty

type: object

**400**

Invalid request.

GeneralError (p. 614)

```
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

**Message**

A message that contains information about the error.

type: string

**ErrorDetails**

A list of error details.

type: array
items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

/greengrass/definition/resources/
ResourceDefinitionId/versions

POST

POST /greengrass/definition/resources/ResourceDefinitionId/versions

Operation ID: CreateResourceDefinitionVersion (p. 73)

Creates a version of a resource definition that has already been defined.

Produces: application/json

Body Parameters

CreateResourceDefinitionVersionRequestBody (p. 562)

Information about the resource definition version.

where used: body; required: true

```json
{
  "Resources": [
    {
      "Id": "string",
      "Name": "string",
      "ResourceDataContainer": {
        "--- NOTE:"": "choose one of the following --*",
        "LocalDeviceResourceData": {
          "SourcePath": "string",
          "GroupOwnerSetting": {
            "AutoAddGroupOwner": true,
            "GroupOwner": "string"
          }
        },
        "LocalVolumeResourceData": {
          "SourcePath": "string",
          "DestinationPath": "string",
          "GroupOwnerSetting": {
            "AutoAddGroupOwner": true,
            "GroupOwner": "string"
          }
        }
      }
    }
  ]
}
```
ResourceDefinitionVersion

Information about a resource definition version.

type: object

Resources

A list of resources.

type: array

items: Resource (p. 656)

Resource

Information about a resource.

type: object

required: ["Id", "Name", "ResourceDataContainer"]

Id

The resource ID, used to refer to a resource in the Lambda function configuration. Maximum length is 128 characters with the pattern [a-zA-Z0-9_:]+. This must be unique within a Greengrass group.

type: string

Name

The descriptive resource name, which is displayed on the AWS IoT Greengrass console. Maximum length is 128 characters with the pattern [a-zA-Z0-9_:]+. This must be unique within a Greengrass group.

type: string
ResourceDataContainer

A container for resource data. The container takes only one of the following supported resource data types: LocalDeviceResourceData, LocalVolumeResourceData, SageMakerMachineLearningModelResourceData, S3MachineLearningModelResourceData, SecretsManagerSecretResourceData.

type: object
NOTE: oneOf

LocalDeviceResourceData

Attributes that define a local device resource.

type: object

SourcePath

The local absolute path of the device resource. The source path for a device resource can refer only to a character device or block device under /dev.

type: string

GroupOwnerSetting

Group owner related settings for local resources.

type: object

AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.

type: boolean

GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.

type: string

LocalVolumeResourceData

Attributes that define a local volume resource.

type: object

SourcePath

The local absolute path of the volume resource on the host. The source path for a volume resource type cannot start with /sys.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

GroupOwnerSetting

Group owner related settings for local resources.
type: object

AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.

type: boolean

GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.

type: string

SageMakerMachineLearningModelResourceData

Attributes that define an Amazon SageMaker machine learning resource.

type: object

SageMakerJobArn

The ARN of the Amazon SageMaker training job that represents the source model.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.

type: object

required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group’s permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are rw (read-write) or ro (read-only).

type: string

enum: ["ro", "rw"]

S3MachineLearningModelResourceData

Attributes that define an Amazon S3 machine learning resource.

type: object

S3Uri

The URI of the source model in an S3 bucket. The model package must be in tar.gz or .zip format.
type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.

type: object

required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are \texttt{rw} (read-write) or \texttt{ro} (read-only).

type: string

enum: ["ro", "rw"]

SecretsManagerSecretResourceData

Attributes that define a secret resource, which references a secret from AWS Secrets Manager. AWS IoT Greengrass stores a local, encrypted copy of the secret on the Greengrass core, where it can be securely accessed by connectors and Lambda functions.

type: object

ARN

The ARN of the Secrets Manager secret to make available on the core. The value of the secret's latest version (represented by the \texttt{AWSCURRENT} staging label) is included by default.

type: string

AdditionalStagingLabelsToDownload

Optional. The staging labels in addition to \texttt{AWSCURRENT} whose values you want to make available on the core.

type: array

Header Parameters

\textbf{X-Amzn-Client-Token (p. 579)}

A client token used to correlate requests and responses.

where used: header; required: false

type: string
Path Parameters

**ResourceDefinitionId (p. 574)**

The ID of the resource definition.

where used: path; required: true
type: string

**CLI**

```
aws greengrass create-resource-definition-version
   --resource-definition-id <value>
   [--resources <value>]
   [--amzn-client-token <value>]
   [--cli-input-json <value>]
   [--generate-cli-skeleton]
```

cli-input-json format:

```
{
   "ResourceDefinitionId": "string",
   "Resources": [
      {
         "Id": "string",
         "Name": "string",
         "ResourceDataContainer": {
            "*-- NOTE:*": "choose one of the following --*",
            "LocalDeviceResourceData": {
               "SourcePath": "string",
               "GroupOwnerSetting": {
                  "AutoAddGroupOwner": "boolean",
                  "GroupOwner": "string"
               }
            },
            "LocalVolumeResourceData": {
               "SourcePath": "string",
               "DestinationPath": "string",
               "GroupOwnerSetting": {
                  "AutoAddGroupOwner": "boolean",
                  "GroupOwner": "string"
               }
            },
            "SageMakerMachineLearningModelResourceData": {
               "SageMakerJobArn": "string",
               "DestinationPath": "string",
               "OwnerSetting": {
                  "GroupOwner": "string",
                  "GroupPermission": "ro|rw"
               }
            },
            "S3MachineLearningModelResourceData": {
               "S3Uri": "string",
               "DestinationPath": "string",
               "OwnerSetting": {
                  "GroupOwner": "string",
                  "GroupPermission": "ro|rw"
               }
            },
            "SecretsManagerSecretResourceData": {
               "ARN": "string",
            }
      }
   ]
}
```
"AdditionalStagingLabelsToDownload": [ "string" ],
} },
"AmznClientToken": "string"
}

Responses

200 (CreateResourceDefinitionVersionResponse)

VersionInformation (p. 676)

{ "Arn": "string", "Id": "string", "Version": "string", "CreationTimestamp": "string" }

VersionInformation

Information about a version.

type: object

Arn

The ARN of the version.

type: string

Id

The ID of the parent definition that the version is associated with.

type: string

Version

The ID of the version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the version was created.

type: string

400

Invalid request.

GeneralError (p. 614)

{ "Message": "string", "ErrorDetails": [ { "DetailedErrorCode": "string",}]}
GeneralError

   General error information.
   type: object
   required: ["Message"]

Message

   A message that contains information about the error.
   type: string

ErrorDetails

   A list of error details.
   type: array
   items: ErrorDetail (p. 600)

ErrorDetail

   Details about the error.
   type: object

DetailedErrorCode

   A detailed error code.
   type: string

DetailedErrorMessage

   A detailed error message.
   type: string

GET

GET /greengrass/definition/resources/{ResourceDefinitionId}/versions

Operation ID: ListResourceDefinitionVersions (p. 224)

Lists the versions of a resource definition.

Produces: application/json

Path Parameters

{ResourceDefinitionId} (p. 574)

   The ID of the resource definition.
   where used: path; required: true
type: string

Query Parameters

**NextToken** (p. 573)

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

where used: query; required: false

type: string

**MaxResults** (p. 573)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

CLI

```bash
aws greengrass list-resource-definition-versions \
  --resource-definition-id <value> \
  [ --next-token <value> ] \
  [ --max-results <value> ] \
  [ --cli-input-json <value> ] \
  [ --generate-cli-skeleton ]
```

cli-input-json format:

```json
{
  "ResourceDefinitionId": "string",
  "NextToken": "string",
  "MaxResults": "integer"
}
```

Responses

**200** ([ListResourceDefinitionVersionsResponse](p. 650))

```json
{
  "Versions": [ 
    {
      "Arn": "string",
      "Id": "string",
      "Version": "string",
      "CreationTimestamp": "string"
    }
  ],
  "NextToken": "string"
}
```

A list of versions.
type: object

Versions

Information about a version.

    type: array

    items: VersionInformation (p. 676)

VersionInformation

Information about a version.

    type: object

Arn

The ARN of the version.

    type: string

Id

The ID of the parent definition that the version is associated with.

    type: string

Version

The ID of the version.

    type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the version was created.

    type: string

NextToken

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

    type: string

400

Invalid request.

GeneralError (p. 614)

```json
{
    "Message": "string",
    "ErrorDetails": [
    {
        "DetailedErrorCode": "string",
        "DetailedErrorMessage": "string"
    }
]
}
```

GeneralError

General error information.

    type: object
required: ["Message"]
Message
A message that contains information about the error.
type: string
ErrorDetails
A list of error details.
type: array
items: ErrorDetail (p. 600)
ErrorDetail
Details about the error.
type: object
DetailedErrorCode
A detailed error code.
type: string
DetailedErrorMessage
A detailed error message.
type: string

/getreengrass/definition/resources/ResourceDefinitionId/versions/ResourceDefinitionVersionId

GET
GET /getreengrass/definition/resources/ResourceDefinitionId/versions/ResourceDefinitionVersionId

Operation ID: GetResourceDefinitionVersion (p. 168)
Retrieves information about a resource definition version, including which resources are included in the version.
Produces: application/json

Path Parameters
ResourceDefinitionId (p. 574)
The ID of the resource definition.
where used: path; required: true
type: string
The ID of the resource definition version. This value maps to the `Version` property of the corresponding `VersionInformation` object, which is returned by `ListResourceDefinitionVersions` requests. If the version is the last one that was associated with a resource definition, the value also maps to the `LatestVersion` property of the corresponding `DefinitionInformation` object.

where used: path; required: true
type: string

### CLI

```
aws greengrass get-resource-definition-version \
  --resource-definition-id <value> \
  --resource-definition-version-id <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "ResourceDefinitionId": "string",
  "ResourceDefinitionVersionId": "string"
}
```

### Responses

#### 200

Success.

GetResourceDefinitionVersionResponse (p. 632)

```
{
  "Arn": "string",
  "Id": "string",
  "Version": "string",
  "CreationTimestamp": "string",
  "Definition": {
    "Resources": [
      {
        "Id": "string",
        "Name": "string",
        "ResourceDataContainer": {
          "**-- NOTE:**: "choose one of the following --**",
          "LocalDeviceResourceData": {
            "SourcePath": "string",
            "GroupOwnerSetting": {
              "AutoAddGroupOwner": true,
              "GroupOwner": "string"
            }
          },
          "LocalVolumeResourceData": {
            "SourcePath": "string",
            "DestinationPath": "string",
            "GroupOwnerSetting": {
              "AutoAddGroupOwner": true,
```
GetResourceDefinitionVersionResponse

Information about a resource definition version.

type: object

Arn

The ARN of the resource definition version.

type: string

Id

The ID of the resource definition version.

type: string

Version

The version of the resource definition version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the resource definition version was created.

type: string

Definition

Information about a resource definition version.

type: object
Resources

A list of resources.

type: array

items: Resource (p. 656)

Resource

Information about a resource.

type: object

required: ["Id", "Name", "ResourceDataContainer"]

Id

The resource ID, used to refer to a resource in the Lambda function configuration. Maximum length is 128 characters with the pattern \[a-zA-Z0-9_:\-]+\]. This must be unique within a Greengrass group.

type: string

Name

The descriptive resource name, which is displayed on the AWS IoT Greengrass console. Maximum length is 128 characters with the pattern \[a-zA-Z0-9_:\-]+\]. This must be unique within a Greengrass group.

type: string

ResourceDataContainer

A container for resource data. The container takes only one of the following supported resource data types: LocalDeviceResourceData, LocalVolumeResourceData, SageMakerMachineLearningModelResourceData, S3MachineLearningModelResourceData, SecretsManagerSecretResourceData.

type: object

NOTE: oneOf

LocalDeviceResourceData

Attributes that define a local device resource.

type: object

SourcePath

The local absolute path of the device resource. The source path for a device resource can refer only to a character device or block device under /dev.

type: string

GroupOwnerSetting

Group owner related settings for local resources.

type: object

AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.
type: boolean

GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.

type: string

LocalVolumeResourceData

Attributes that define a local volume resource.

type: object

SourcePath

The local absolute path of the volume resource on the host. The source path for a volume resource type cannot start with `/sys`.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

GroupOwnerSetting

Group owner related settings for local resources.

type: object

AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.

type: boolean

GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.

type: string

SageMakerMachineLearningModelResourceData

Attributes that define an Amazon SageMaker machine learning resource.

type: object

SageMakerJobArn

The ARN of the Amazon SageMaker training job that represents the source model.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.
GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are rw (read-write) or ro (read-only).

type: string
enum: ["ro", "rw"]

S3MachineLearningModelResourceData

Attributes that define an Amazon S3 machine learning resource.

type: object

S3Uri

The URI of the source model in an S3 bucket. The model package must be in tar.gz or .zip format.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.

type: object
required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are rw (read-write) or ro (read-only).

type: string
enum: ["ro", "rw"]

SecretsManagerSecretResourceData

Attributes that define a secret resource, which references a secret from AWS Secrets Manager. AWS IoT Greengrass stores a local, encrypted copy of the secret on the Greengrass core, where it can be securely accessed by connectors and Lambda functions.
type: object

ARN

The ARN of the Secrets Manager secret to make available on the core. The value of the secret's latest version (represented by the `AWSCURRENT` staging label) is included by default.

type: string

AdditionalStagingLabelsToDownload

Optional. The staging labels in addition to `AWSCURRENT` whose values you want to make available on the core.

type: array

400

Invalid request.

**GeneralError (p. 614)**

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: **ErrorDetail (p. 600)**

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string
DetailedErrorMessage

A detailed error message.

type: string

/greengrass/definition/subscriptions

GET

GET /greengrass/definition/subscriptions

Operation ID: ListSubscriptionDefinitions (p. 232)

Retrieves a list of subscription definitions.

Produces: application/json

Query Parameters

MaxResults (p. 573)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

NextToken (p. 573)

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

where used: query; required: false

type: string

CLI

aws greengrass list-subscription-definitions
   [--max-results <value>] \
   [--next-token <value>] \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]

cli-input-json format:

```json
{
   "MaxResults": "integer",
   "NextToken": "string"
}
```

Responses

200 (ListSubscriptionDefinitionsResponse)

ListDefinitionsResponse (p. 645)
ListDefinitionsResponse

A list of definitions.

type: object

Definitions

Information about a definition.

type: array

items: DefinitionInformation (p. 594)

DefinitionInformation

Information about a definition.

type: object

Name

The name of the definition.

type: string

Id

The ID of the definition.

type: string

Arn

The ARN of the definition.

type: string

tags

The resource tags.

type: object

additionalProperties: The key-value pair for the resource tag. Type: string
LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.

type: string

LatestVersion

The ID of the latest version associated with the definition.

type: string

LatestVersionArn

The ARN of the latest version associated with the definition.

type: string

NextToken

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

type: string

POST

POST /greengrass/definition/subscriptions

Operation ID: CreateSubscriptionDefinition (p. 84)

Creates a subscription definition. You can provide the initial version of the subscription definition now or use CreateSubscriptionDefinitionVersion later.

Produces: application/json

Body Parameters

CreateSubscriptionDefinitionRequestBody (p. 568)

where used: body; required: true

{
   "Name": "string",
   "InitialVersion": {
      "Subscriptions": [
         {
            "Id": "string",
            "Source": "string",
            "Subject": "string",
            "Target": "string"
         }
      ],
   "tags": {
      "keyName0": "string",
      "keyName1": "string",
      "keyName2": "string"
   }
   },
}
Name

The name of the subscription definition.

type: string

InitialVersion

Information about a subscription definition version.

type: object

Subscriptions

A list of subscriptions.

type: array

items: Subscription (p. 672)

Subscription

Information about a subscription.

type: object

required: ["Id", "Source", "Subject", "Target"]

Id

A descriptive or arbitrary ID for the subscription. This value must be unique within the
subscription definition version. Maximum length is 128 characters with the pattern
[a-zA-Z0-9:_-]+.

type: string

Source

The source of the subscription. Can be a thing ARN, the ARN of a Lambda function alias
(recommended) or version, a connector ARN, 'cloud' (which represents AWS IoT), or
'GGShadowService'. If you specify a Lambda function, this ARN should match the ARN used to
add the function to the Greengrass group.

type: string

Subject

The MQTT topic used to route the message.

type: string

Target

Where the message is sent. Can be a thing ARN, the ARN of a Lambda function alias
(recommended) or version, a connector ARN, 'cloud' (which represents AWS IoT), or
'GGShadowService'. If you specify a Lambda function, this ARN should match the ARN used to
add the function to the Greengrass group.

type: string

tags

The resource tags.
type: object
additionalProperties: The key-value pair for the resource tag. Type: string

Header Parameters

X-Amzn-Client-Token (p. 579)
A client token used to correlate requests and responses.
where used: header; required: false
type: string

CLI

```bash
aws greengrass create-subscription-definition
  [--name <value>] 
  [--initial-version <value>] 
  [--tags <value>] 
  [--amzn-client-token <value>] 
  [--cli-input-json <value>] 
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "Name": "string",
  "InitialVersion": {
    "Subscriptions": [
      {
        "Id": "string",
        "Source": "string",
        "Subject": "string",
        "Target": "string"
      }
    ],
    "tags": {
      "additionalProperty0": "string",
      "additionalProperty1": "string",
      "additionalProperty2": "string"
    },
    "AmznClientToken": "string"
  }
}
```

Responses

200 (CreateSubscriptionDefinitionResponse)

DefinitionInformation (p. 594)

```json
{
  "Name": "string",
  "Id": "string",
  "Arn": "string",
  "LastUpdatedTimestamp": "string",
  "CreationTimestamp": "string",
  "LatestVersion": "string",
}
```
"LatestVersionArn": "string"
}

DefinitionInformation

Information about a definition.

type: object

Name

The name of the definition.

type: string

Id

The ID of the definition.

type: string

Arn

The ARN of the definition.

type: string

LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.

type: string

LatestVersion

The ID of the latest version associated with the definition.

type: string

LatestVersionArn

The ARN of the latest version associated with the definition.

type: string

400

Invalid request.

GeneralError (p. 614)

{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
GeneralError
    General error information.
    type: object
    required: ["Message"]
    Message
    A message that contains information about the error.
    type: string
    ErrorDetails
    A list of error details.
    type: array
    items: ErrorDetail (p. 600)
    ErrorDetail
    Details about the error.
    type: object
    DetailedErrorCode
    A detailed error code.
    type: string
    DetailedErrorMessage
    A detailed error message.
    type: string

/getrenggrass/definition/subscriptions/SubscriptonDefinitionId

GET

GET /greengrass/definition/subscriptions/SubscriptonDefinitionId

Operation ID: GetSubscriptionDefinition (p. 176)

Retrieves information about a subscription definition.

Produces: application/json

Path Parameters

SubscriptionDefinitionId (p. 576)

The ID of the subscription definition.

where used: path; required: true

type: string
CLI

```
aws greengrass get-subscription-definition \
  --subscription-definition-id <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "SubscriptionDefinitionId": "string"
}
```

Responses

200 (GetSubscriptionDefinitionResponse)

`DefinitionInformation (p. 594)`

```
{
  "Name": "string",
  "Id": "string",
  "Arn": "string",
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  },
  "LastUpdatedTimestamp": "string",
  "CreationTimestamp": "string",
  "LatestVersion": "string",
  "LatestVersionArn": "string"
}
```

DefinitionInformation

Information about a definition.

type: object

Name

The name of the definition.

type: string

Id

The ID of the definition.

type: string

Arn

The ARN of the definition.

type: string

tags

The resource tags.

type: object
additionalProperties: The key-value pair for the resource tag. Type: string

LastUpdatedTimestamp
The time, in milliseconds since the epoch, when the definition was last updated.
type: string

CreationTimestamp
The time, in milliseconds since the epoch, when the definition was created.
type: string

LatestVersion
The ID of the latest version associated with the definition.
type: string

LatestVersionArn
The ARN of the latest version associated with the definition.
type: string

400
Invalid request.

GeneralError (p. 614)

```json
{
   "Message": "string",
   "ErrorDetails": [
   {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
   }
   ]
}
```

GeneralError
General error information.
type: object
required: ["Message"]

Message
A message that contains information about the error.
type: string

ErrorDetails
A list of error details.
type: array
items: ErrorDetail (p. 600)

ErrorDetail
Details about the error.
DELETE

DELETE /greengrass/definition/subscriptions/SubscriptionDefinitionId

Operation ID: DeleteSubscriptionDefinition (p. 104)

Deletes a subscription definition.

Produces: application/json

Path Parameters

SubscriptionDefinitionId (p. 576)

The ID of the subscription definition.

where used: path; required: true

type: string

CLI

aws greengrass delete-subscription-definition
   --subscription-definition-id <value> \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]

cli-input-json format:

{
   "SubscriptionDefinitionId": "string"
}

Responses

200

Success.

Empty (p. 600)
Empty Schema
Empty
  type: object

400
Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError
General error information.
  type: object
  required: ["Message"]
Message
A message that contains information about the error.
  type: string
ErrorDetails
A list of error details.
  type: array
  items: ErrorDetail (p. 600)
ErrorDetail
Details about the error.
  type: object
DetailedErrorCode
A detailed error code.
  type: string
DetailedErrorMessage
A detailed error message.
  type: string

PUT
PUT /greengrass/definition/subscriptions/SubscriptionDefinitionId
Operation ID: UpdateSubscriptionDefinition (p. 267)

Updates the name of a subscription definition. To update the list of available subscriptions, use CreateSubscriptionDefinitionVersion.

Produces: application/json

**Body Parameters**

**UpdateDefinitionRequestBody** (p. 578)

where used: body; required: true

```
{
   "Name": "string"
}
```

**Name**

The name of the definition.

required: true
type: string

**Path Parameters**

**SubscriptionDefinitionId** (p. 576)

The ID of the subscription definition.

where used: path; required: true
type: string

**CLI**

```
aws greengrass update-subscription-definition \
   --subscription-definition-id <value> \
   [--name <value>] \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```
{
   "SubscriptionDefinitionId": "string",
   "Name": "string"
}
```

**Responses**

**200**

Success.

Empty (p. 600)
Empty Schema

Empty

type: object

400

Invalid request.

GeneralError (p. 614)

{
  "Message": "string",
  "ErrorDetails": [
  {
    "DetailedErrorCode": "string",
    "DetailedErrorMessage": "string"
  }
  ]
}

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string
POST /greengrass/definition/subscriptions/SubscriptionDefinitionId/versions

Operation ID: CreateSubscriptionDefinitionVersion (p. 88)

Creates a version of a subscription definition that has already been defined.

Produce: application/json

Body Parameters

CreateSubscriptionDefinitionVersionRequestBody (p. 568)

where used: body; required: true

```json
{
  "Subscriptions": [
    {
      "Id": "string",
      "Source": "string",
      "Subject": "string",
      "Target": "string"
    }
  ]
}
```

SubscriptionDefinitionVersion

Information about a subscription definition version.

type: object

Subscriptions

A list of subscriptions.

type: array

items: Subscription (p. 672)

Subscription

Information about a subscription.

type: object

required: ["Id", "Source", "Subject", "Target"]

Id

A descriptive or arbitrary ID for the subscription. This value must be unique within the subscription definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9-_]+.

type: string
Source

The source of the subscription. Can be a thing ARN, the ARN of a Lambda function alias (recommended) or version, a connector ARN, 'cloud' (which represents AWS IoT), or 'GGShadowService'. If you specify a Lambda function, this ARN should match the ARN used to add the function to the Greengrass group.

type: string

Subject

The MQTT topic used to route the message.

type: string

Target

Where the message is sent. Can be a thing ARN, the ARN of a Lambda function alias (recommended) or version, a connector ARN, 'cloud' (which represents AWS IoT), or 'GGShadowService'. If you specify a Lambda function, this ARN should match the ARN used to add the function to the Greengrass group.

type: string

Header Parameters

X-Amzn-Client-Token (p. 579)

A client token used to correlate requests and responses.

where used: header; required: false

    type: string

Path Parameters

SubscriptionDefinitionId (p. 576)

The ID of the subscription definition.

where used: path; required: true

    type: string

CLI

```
aws greengrass create-subscription-definition-version \
   --subscription-definition-id <value> \
   [--subscriptions <value>] \
   [--amzn-client-token <value>] \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "SubscriptionDefinitionId": "string",
   "Subscriptions": [
```

455
AWS Greengrass API Reference
/greengrass/definition/subscriptions/
SubscriptionDefinitionId/versions

{  
  "Id": "string",
  "Source": "string",
  "Subject": "string",
  "Target": "string"
}

"AmznClientToken": "string"

Responses

200 (CreateSubscriptionDefinitionVersionResponse)

VersionInformation (p. 676)

{  
  "Arn": "string",
  "Id": "string",
  "Version": "string",
  "Version": "string",
  "CreationTimestamp": "string"
}

VersionInformation

Information about a version.

type: object

Arn

The ARN of the version.

type: string

Id

The ID of the parent definition that the version is associated with.

type: string

Version

The ID of the version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the version was created.

type: string

400

Invalid request.

GeneralError (p. 614)

{  
  "Message": "string",
  "ErrorDetails": [  
    {  
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
GET

GET /greengrass/definition/subscriptions/SubscriptionDefinitionId/versions

Operation ID: ListSubscriptionDefinitionVersions (p. 229)

Lists the versions of a subscription definition.

Produces: application/json

Path Parameters

SubscriptionDefinitionId (p. 576)

The ID of the subscription definition.

where used: path; required: true

type: string
## Query Parameters

**NextToken (p. 573)**

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

where used: query; required: false

type: string

**MaxResults (p. 573)**

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

### CLI

```bash
aws greengrass list-subscription-definition-versions \
   --subscription-definition-id <value> \
   [--next-token <value>] \
   [--max-results <value>] \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "SubscriptionDefinitionId": "string",
   "NextToken": "string",
   "MaxResults": "integer"
}
```

### Responses

**200 (ListSubscriptionDefinitionVersionsResponse)**

#### ListVersionsResponse (p. 650)

```json
{
   "Versions": [
   {
      "Arn": "string",
      "Id": "string",
      "Version": "string",
      "CreationTimestamp": "string"
   }
   ],
   "NextToken": "string"
}
```

ListVersionsResponse

A list of versions.

where used: response; required: false

type: object
Versions

Information about a version.

- type: array
- items: [VersionInformation (p. 676)]

VersionInformation

Information about a version.

- type: object

  Arn
  
  The ARN of the version.
  
  - type: string

  Id
  
  The ID of the parent definition that the version is associated with.
  
  - type: string

  Version
  
  The ID of the version.
  
  - type: string

  CreationTimestamp
  
  The time, in milliseconds since the epoch, when the version was created.
  
  - type: string

  NextToken
  
  The token for the next set of results, or null if there are no more results.
  
  - type: string

400

Invalid request.

**GeneralError (p. 614)**

```
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

**GeneralError**

General error information.

- type: object

  required: ["Message"]
Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

/greengrass/definition/subscriptions/SubscriptionDefinitionId/versions/SubscriptionDefinitionVersionId

GET

GET /greengrass/definition/subscriptions/SubscriptionDefinitionId/versions/SubscriptionDefinitionVersionId

Operation ID: GetSubscriptionDefinitionVersion (p. 179)

Retrieves information about a subscription definition version.

Produces: application/json

Path Parameters

SubscriptionDefinitionId (p. 576)

The ID of the subscription definition.

where used: path; required: true

type: string

SubscriptionDefinitionVersionId (p. 576)

The ID of the subscription definition version. This value maps to the Version property of the corresponding VersionInformation object, which is returned by
ListSubscriptionDefinitionVersions requests. If the version is the last one that was associated with a subscription definition, the value also maps to the LatestVersion property of the corresponding DefinitionInformation object.

where used: path; required: true
type: string

Query Parameters

NextToken (p. 573)
The token for the next set of results, or null if there are no more results.

where used: query; required: false
type: string

CLI

```bash
aws greengrass get-subscription-definition-version \
  --subscription-definition-id <value> \ 
  --subscription-definition-version-id <value> \ 
  [--next-token <value>] \ 
  [--cli-input-json <value>] \ 
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "SubscriptionDefinitionId": "string",
  "SubscriptionDefinitionVersionId": "string",
  "NextToken": "string"
}
```

Responses

200 (GetSubscriptionDefinitionVersionResponse)

GetSubscriptionDefinitionVersionResponse (p. 637)

```json
{
  "Arn": "string",
  "Id": "string",
  "Version": "string",
  "CreationTimestamp": "string",
  "Definition": {
    "Subscriptions": [
      {
        "Id": "string",
        "Source": "string",
        "Subject": "string",
        "Target": "string"
      }
    ],
  },
  "NextToken": "string"
}
```
GetSubscriptionDefinitionVersionResponse

Information about a subscription definition version.

type: object

Arn

The ARN of the subscription definition version.

type: string

Id

The ID of the subscription definition version.

type: string

Version

The version of the subscription definition version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the subscription definition version was created.

type: string

Definition

Information about a subscription definition version.

type: object

Subscriptions

A list of subscriptions.

type: array

items: Subscription (p. 672)

Subscription

Information about a subscription.

type: object

required: ["Id", "Source", "Subject", "Target"]

Id

A descriptive or arbitrary ID for the subscription. This value must be unique within the subscription definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-\.]+.

type: string

Source

The source of the subscription. Can be a thing ARN, the ARN of a Lambda function alias (recommended) or version, a connector ARN, 'cloud' (which represents AWS IoT), or 'GGShadowService'. If you specify a Lambda function, this ARN should match the ARN used to add the function to the Greengrass group.

type: string
Subject

The MQTT topic used to route the message.

type: string

Target

Where the message is sent. Can be a thing ARN, the ARN of a Lambda function alias (recommended) or version, a connector ARN, 'cloud' (which represents AWS IoT), or 'GGShadowService'. If you specify a Lambda function, this ARN should match the ARN used to add the function to the Greengrass group.

type: string

NextToken

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

type: string

400

Invalid request.

**GeneralError (p. 614)**

```
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

**GeneralError**

General error information.

type: object

required: ["Message"]

**Message**

A message that contains information about the error.

type: string

**ErrorDetails**

A list of error details.

type: array

items: **ErrorDetail (p. 600)**

**ErrorDetail**

Details about the error.

type: object

**DetailedErrorCode**

A detailed error code.
type: string

DetailedErrorMessage

A detailed error message.

type: string

/greengrass/groups

GET

GET /greengrass/groups

Operation ID: ListGroups (p. 217)

Retrieves a list of groups.

Produces: application/json

Query Parameters

**MaxResults** (p. 573)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

**NextToken** (p. 573)

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

where used: query; required: false

type: string

CLI

```
aws greengrass list-groups \
  [--max-results <value>] \
  [--next-token <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "MaxResults": "integer",
  "NextToken": "string"
}
```

Responses

**200** (ListGroupsResponse)

ListGroupsResponse (p. 649)
{  "Groups": [  {  "Name": "string",  "Id": "string",  "Arn": "string",  "LastUpdatedTimestamp": "string",  "CreationTimestamp": "string",  "LatestVersion": "string",  "LatestVersionArn": "string"  }  ],  "NextToken": "string"  }

ListGroupsResponse
  type: object
Groups
    Information about a group.
    type: array
    items: GroupInformation (p. 640)

GroupInformation
  Information about a group.
  type: object
Name
    The name of the group.
    type: string
Id
    The ID of the group.
    type: string
Arn
    The ARN of the group.
    type: string
LastUpdatedTimestamp
    The time, in milliseconds since the epoch, when the group was last updated.
    type: string
CreationTimestamp
    The time, in milliseconds since the epoch, when the group was created.
    type: string
LatestVersion
    The ID of the latest version associated with the group.
    type: string
LatestVersionArn
The ARN of the latest version associated with the group.
type: string

NextToken
The token for the next set of results, or null if there are no more results.
type: string

POST
POST /greengrass/groups
Operation ID: CreateGroup (p. 47)
Creates a group. You can provide the initial version of the group or use CreateGroupVersion later. Tip: You can use the gg_group_setup package (https://github.com/awslabs/aws-greengrass-group-setup) as a library or command-line application to create and deploy Greengrass groups.

Produces: application/json

Body Parameters

CreateGroupRequestBody (p. 558)
where used: body; required: true

```json
{
  "Name": "string",
  "InitialVersion": {
    "CoreDefinitionVersionArn": "string",
    "DeviceDefinitionVersionArn": "string",
    "FunctionDefinitionVersionArn": "string",
    "SubscriptionDefinitionVersionArn": "string",
    "LoggerDefinitionVersionArn": "string",
    "ResourceDefinitionVersionArn": "string",
    "ConnectorDefinitionVersionArn": "string"
  },
  "tags": {
    "keyName0": "string",
    "keyName1": "string",
    "keyName2": "string"
  }
}
```

Name
The name of the group.
type: string

InitialVersion
Information about a group version.
type: object

CoreDefinitionVersionArn
The ARN of the core definition version for this group.
DeviceDefinitionVersionArn
The ARN of the device definition version for this group.
type: string

FunctionDefinitionVersionArn
The ARN of the function definition version for this group.
type: string

SubscriptionDefinitionVersionArn
The ARN of the subscription definition version for this group.
type: string

LoggerDefinitionVersionArn
The ARN of the logger definition version for this group.
type: string

ResourceDefinitionVersionArn
The ARN of the resource definition version for this group.
type: string

ConnectorDefinitionVersionArn
The ARN of the connector definition version for this group.
type: string

tags
The resource tags.
type: object

additionalProperties: The key-value pair for the resource tag. Type: string

Header Parameters

X-Amzn-Client-Token (p. 579)
A client token used to correlate requests and responses.
where used: header; required: false
type: string

CLI

aws greengrass create-group \  
[ --name <value> ] \  
[ --initial-version <value> ] \  
[ --tags <value> ] \  
[ --amzn-client-token <value> ] \

cli-input-json format:

```json
{
  "Name": "string",
  "InitialVersion": {
    "CoreDefinitionVersionArn": "string",
    "DeviceDefinitionVersionArn": "string",
    "FunctionDefinitionVersionArn": "string",
    "SubscriptionDefinitionVersionArn": "string",
    "LoggerDefinitionVersionArn": "string",
    "ResourceDefinitionVersionArn": "string",
    "ConnectorDefinitionVersionArn": "string"
  },
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  },
  "AmznClientToken": "string"
}
```

Responses

200 (CreateGroupResponse)

Success. The group was created.

**DefinitionInformation (p. 594)**

```json
{
  "Name": "string",
  "Id": "string",
  "Arn": "string",
  "LastUpdatedTimestamp": "string",
  "CreationTimestamp": "string",
  "LatestVersion": "string",
  "LatestVersionArn": "string"
}
```

**DefinitionInformation**

Information about a definition.

- **Name**
  
  The name of the definition.

  type: string

- **Id**
  
  The ID of the definition.

  type: string

- **Arn**
  
  The ARN of the definition.
type: string
LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.

type: string
CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.

type: string
LatestVersion

The ID of the latest version associated with the definition.

type: string
LatestVersionArn

The ARN of the latest version associated with the definition.

type: string

400

Invalid request.

GeneralError (p. 614)

{
   "Message": "string",
   "ErrorDetails": [
   {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
   }
   ]
}

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.
/greengrass/groups/GroupId

GET

GET /greengrass/groups/GroupId

Operation ID: GetGroup (p. 148)

Retrieves information about a group.

Produces: application/json

Path Parameters

GroupId (p. 571)

The ID of the Greengrass group.

where used: path; required: true

type: string

CLI

aws greengrass get-group \
  --group-id <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]

cli-input-json format:

{
  "GroupId": "string"
}

Responses

200 (GetGroupResponse)

Success.

DefinitionInformation (p. 594)

{
"Name": "string",
"Id": "string",
"Arn": "string",
"tags": {
   "additionalProperty0": "string",
   "additionalProperty1": "string",
   "additionalProperty2": "string"
},
"LastUpdatedTimestamp": "string",
"CreationTimestamp": "string",
"LatestVersion": "string",
"LatestVersionArn": "string"

DefinitionInformation
   Information about a definition.
   type: object
Name
   The name of the definition.
   type: string
Id
   The ID of the definition.
   type: string
Arn
   The ARN of the definition.
   type: string
tags
   The resource tags.
   type: object
   additionalProperties: The key-value pair for the resource tag. Type: string
LastUpdatedTimestamp
   The time, in milliseconds since the epoch, when the definition was last updated.
   type: string
CreationTimestamp
   The time, in milliseconds since the epoch, when the definition was created.
   type: string
LatestVersion
   The ID of the latest version associated with the definition.
   type: string
LatestVersionArn
   The ARN of the latest version associated with the definition.
400

Invalid request.

**GeneralError (p. 614)**

```
{
   "Message": "string",
   "ErrorDetails": [
   {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
   }
   ]
}
```

**GeneralError**

General error information.

*type: object*

*required: ["Message"]*

**Message**

A message that contains information about the error.

*type: string*

**ErrorDetails**

A list of error details.

*type: array*

*items: ErrorDetail (p. 600)*

**ErrorDetail**

Details about the error.

*type: object*

**DetailedErrorCode**

A detailed error code.

*type: string*

**DetailedErrorMessage**

A detailed error message.

*type: string*

**DELETE**

DELETE `/greengrass/groups/GroupId`

Operation ID: DeleteGroup (p. 99)

Deletes a group.
Produces: application/json

Path Parameters

**GroupId (p. 571)**

The ID of the Greengrass group.

where used: path; required: true
type: string

CLI

```bash
aws greengrass delete-group
  --group-id <value>
  [--cli-input-json <value>]
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "GroupId": "string"
}
```

Responses

**200**

Success.

Empty (p. 600)

```json
{}
```

Empty Schema

Empty
type: object

**400**

Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```
GeneralError
   General error information.
   type: object
   required: ["Message"]
Message
   A message that contains information about the error.
   type: string
ErrorDetails
   A list of error details.
   type: array
   items: ErrorDetail (p. 600)
ErrorDetail
   Details about the error.
   type: object
DetailedErrorCode
   A detailed error code.
   type: string
DetailedErrorMessage
   A detailed error message.
   type: string

PUT

PUT /greengrass/groups/GroupId

Operation ID: UpdateGroup (p. 258)

Updates the name of a group. To update group components, use CreateGroupVersion.

Produces: application/json

Body Parameters

UpdateDefinitionRequestBody (p. 578)

where used: body; required: true

{  
   "Name": "string"
}

Name

The name of the definition.
required: true
type: string

Path Parameters

**GroupId (p. 571)**

The ID of the Greengrass group.
where used: path; required: true
type: string

CLI

```bash
aws greengrass update-group \
  --group-id <value> \
  [--name <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "GroupId": "string",
  "Name": "string"
}
```

Responses

**200**

Success.

**Empty (p. 600)**

```json
{
}
```

Empty Schema

Empty
type: object

**400**

Invalid request.

**GeneralError (p. 614)**

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string",
      "DetailedException": "string",
      "DetailedFault": "string",
      "DetailedType": "string",
      "Code": "string",
      "Message": "string",
      "RequestId": "string",
      "ErrorType": "string"
    }
  ]
}
```
"DetailedErrorMessage": "string"
}
]

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

/greengrass/groups/GroupId/certificateauthorities

GET

GET /greengrass/groups/GroupId/certificateauthorities

Operation ID: ListGroupCertificateAuthorities (p. 211)

Retrieves the current CAs for a group.

Producers: application/json

Path Parameters

GroupId (p. 571)

The ID of the Greengrass group.

476
where used: path; required: true

type: string

CLI

aws greengrass list-group-certificate-authorities \
   --group-id <value> \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]

cli-input-json format:

{  
   "GroupId": "string"
}

Responses

200

Success. The response body contains the PKI configuration.

ListGroupCertificateAuthoritiesResponse (p. 648)

{  
   "GroupCertificateAuthorities": [  
      {  
         "GroupCertificateAuthorityId": "string",  
         "GroupCertificateAuthorityArn": "string"  
      }  
   ]
}

ListGroupCertificateAuthoritiesResponse

type: object

GroupCertificateAuthorities

A list of certificate authorities associated with the group.

type: array

items: GroupCertificateAuthorityProperties (p. 639)

Group Certificate Authority Properties

Information about a certificate authority for a group.

type: object

GroupCertificateAuthorityId

The ID of the certificate authority for the group.

type: string

GroupCertificateAuthorityArn

The ARN of the certificate authority for the group.
type: string

400

Invalid request.

GeneralError (p. 614)

{
 "Message": "string",
 "ErrorDetails": [
   {
     "DetailedErrorCode": "string",
     "DetailedErrorMessage": "string"
   }
  ]
}

GeneralError

General error information.

  type: object
  required: ["Message"]

Message

A message that contains information about the error.

  type: string

ErrorDetails

A list of error details.

  type: array
  items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

  type: object

DetailedErrorCode

A detailed error code.

  type: string

DetailedErrorMessage

A detailed error message.

  type: string

500

Server error.

GeneralError (p. 614)

{
 "Message": "string",
}
"ErrorDetails": [
    {
        "DetailedErrorCode": "string",
        "DetailedErrorMessage": "string"
    }
]

**GeneralError**

General error information.

*type: object*

.required: ["Message"]

**Message**

A message that contains information about the error.

*type: string*

**ErrorDetails**

A list of error details.

*type: array*

.items: **ErrorDetail** (p. 600)

**ErrorDetail**

Details about the error.

*type: object*

**DetailedErrorCode**

A detailed error code.

*type: string*

**DetailedErrorMessage**

A detailed error message.

*type: string*

**POST**

POST /greengrass/groups/GroupId/certificateauthorities

**Operation ID:** CreateGroupCertificateAuthority (p. 51)

Creates a CA for the group. If a CA already exists, it rotates the existing CA.

**Produces:** application/json

**Header Parameters**

**X-Amzn-Client-Token (p. 579)**

A client token used to correlate requests and responses.
Path Parameters

**GroupId (p. 571)**

The ID of the Greengrass group.

where used: path; required: true

type: string

CLI

```bash
aws greengrass create-group-certificate-authority \
--group-id <value> \
[--amzn-client-token <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

cli-input-json format:

```json
{
"GroupId": "string",
"AmznClientToken": "string"
}
```

Responses

**200**

Success. The response body contains the new, active CA ARN.

**CreateGroupCertificateAuthorityResponse (p. 593)**

```json
{
  "GroupCertificateAuthorityArn": "string"
}
```

CreateGroupCertificateAuthorityResponse

type: object

**GroupCertificateAuthorityArn**

The ARN of the group certificate authority.

type: string

**400**

Invalid request.

**GeneralError (p. 614)**

```json
{
}
```
"Message": "string",
"ErrorDetails": [
    {
        "DetailedErrorCode": "string",
        "DetailedErrorMessage": "string"
    }
]
}

**GeneralError**

General error information.

- **type**: object
- **required**: ["Message"]

**Message**

A message that contains information about the error.

- **type**: string

**ErrorDetails**

A list of error details.

- **type**: array
  - **items**: ErrorDetail (p. 600)

**ErrorDetail**

Details about the error.

- **type**: object

**DetailedErrorCode**

A detailed error code.

- **type**: string

**DetailedErrorMessage**

A detailed error message.

- **type**: string

**500**

Server error.

**GeneralError (p. 614)**

```json
{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}
```
AWS Greengrass API Reference
/greengrass/groups/GroupId/certificateauthorities/configuration/expiry

GeneralError
  General error information.
  type: object
  required: ["Message"]
Message
  A message that contains information about the error.
  type: string
ErrorDetails
  A list of error details.
  type: array
  items: ErrorDetail (p. 600)
ErrorDetail
  Details about the error.
  type: object
DetailedErrorCode
  A detailed error code.
  type: string
DetailedErrorMessage
  A detailed error message.
  type: string

/greengrass/groups/GroupId/certificateauthorities/configuration/expiry

GET

GET /greengrass/groups/GroupId/certificateauthorities/configuration/expiry

Operation ID: GetGroupCertificateConfiguration (p. 154)

Retrieves the current configuration for the CA used by the group.

Produces: application/json

Path Parameters

GroupId (p. 571)
  The ID of the Greengrass group.
  where used: path; required: true
  type: string
CLI

```bash
aws greengrass get-group-certificate-configuration \
  --group-id <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "GroupId": "string"
}
```

Responses

200

Success. The response body contains the PKI configuration.

GroupCertificateConfiguration (p. 639)

```json
{
  "GroupId": "string",
  "CertificateAuthorityExpiryInMilliseconds": "string",
  "CertificateExpiryInMilliseconds": "string"
}
```

GroupCertificateConfiguration

Information about a group certificate configuration.

- **GroupId**
  - type: string
  - The ID of the group certificate configuration.

- **CertificateAuthorityExpiryInMilliseconds**
  - type: string
  - The amount of time, in milliseconds, before the certificate authority expires.

- **CertificateExpiryInMilliseconds**
  - type: string
  - The amount of time, in milliseconds, before the certificate expires.

400

Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
    }
  ]
}
```
"DetailedErrorMessage": "string"
"
]

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

500

Server error.

GeneralError (p. 614)

{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}

GeneralError

General error information.


type: object
PUT

PUT /greengrass/groups/GroupId/certificateauthorities/configuration/expiry

Operation ID: UpdateGroupCertificateConfiguration (p. 260)

Updates the certificate expiry time for a group.

Produces: application/json

Body Parameters

UpdateGroupCertificateConfigurationRequestBody (p. 579)

where used: body; required: true

```json
{
  "CertificateExpiryInMilliseconds": "string"
}
```

updateGroupCertificateConfigurationRequest

type: object

required: ["CertificateExpiryInMilliseconds"]

CertificateExpiryInMilliseconds

The amount of time, in milliseconds, before the certificate expires.
type: string

**Path Parameters**

**GroupId (p. 571)**

The ID of the Greengrass group.

where used: path; required: true

type: string

**CLI**

```
aws greengrass update-group-certificate-configuration
--group-id <value> \
[--certificate-expiry-in-milliseconds <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "GroupId": "string",
  "CertificateExpiryInMilliseconds": "string"
}
```

**Responses**

200

Success. The response body contains the PKI configuration.

**GroupCertificateConfiguration (p. 639)**

```
{
  "GroupId": "string",
  "CertificateAuthorityExpiryInMilliseconds": "string",
  "CertificateExpiryInMilliseconds": "string"
}
```

**GroupCertificateConfiguration**

Information about a group certificate configuration.

- type: object
  
  **GroupId**

  The ID of the group certificate configuration.

  type: string

  **CertificateAuthorityExpiryInMilliseconds**

  The amount of time, in milliseconds, before the certificate authority expires.

  type: string
CertificateExpiryInMilliseconds

The amount of time, in milliseconds, before the certificate expires.

type: string

400

Invalid request.

GeneralError (p. 614)

```json
{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

500

Server error.
GeneralError (p. 614)

```json
{
   "Message": "string",
   "ErrorDetails": [
      {
         "DetailedErrorCode": "string",
         "DetailedErrorMessage": "string"
      }
   ]
}
```

**GeneralError**

General error information.

- type: object
  - required: ["Message"]

**Message**

A message that contains information about the error.

- type: string

**ErrorDetails**

A list of error details.

- type: array
  - items: ErrorDetail (p. 600)

**ErrorDetail**

Details about the error.

- type: object

**DetailedErrorCode**

A detailed error code.

- type: string

**DetailedErrorMessage**

A detailed error message.

- type: string

---

**/greengrass/groups/GroupId/certificateauthorities/CertificateAuthorityId**

**GET**

GET `/greengrass/groups/GroupId/certificateauthorities/CertificateAuthorityId`

**Operation ID:** GetGroupCertificateAuthority (p. 151)

Retreives the CA associated with a group. Returns the public key of the CA.
Path Parameters

GroupId (p. 571)

The ID of the Greengrass group.

where used: path; required: true

type: string

CertificateAuthorityId (p. 546)

The ID of the certificate authority.

where used: path; required: true

type: string

CLI

```
aws greengrass get-group-certificate-authority \
   --group-id <value> \
   --certificate-authority-id <value> \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```
{
   "GroupId": "string",
   "CertificateAuthorityId": "string"
}
```

Responses

200

Success. The response body contains the PKI configuration.

GetGroupCertificateAuthorityResponse (p. 628)

```
{
   "PemEncodedCertificate": "string",
   "GroupCertificateAuthorityArn": "string",
   "GroupCertificateAuthorityId": "string"
}
```

GetGroupCertificateAuthorityResponse

Information about a certificate authority for a group.

   type: object

PemEncodedCertificate

The PEM encoded certificate for the group.

   type: string
GroupCertificateAuthorityArn

The ARN of the certificate authority for the group.

type: string

GroupCertificateAuthorityId

The ID of the certificate authority for the group.

type: string

400

Invalid request.

GeneralError (p. 614)

```json
{
"Message": "string",
"ErrorDetails": [
{
"DetailedErrorCode": "string",
"DetailedErrorMessage": "string"
}
]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string
500
Server error.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError

General error information.

- type: object
- required: ["Message"]

Message

A message that contains information about the error.

- type: string

ErrorDetails

A list of error details.

- type: array

  - items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

- type: object

  - DetailedErrorCode
  - type: string

  - DetailedErrorMessage
  - type: string

/greengrass/groups/GroupId/deployments

POST

POST /greengrass/groups/GroupId/deployments

Operation ID: CreateDeployment (p. 24)
Creates a deployment. CreateDeployment requests are idempotent with respect to the X-Amzn-Client-Token token and the request parameters.

Produces: application/json

Body Parameters

CreateDeploymentRequestBody (p. 551)

where used: body; required: true

```json
{
    "DeploymentType": "NewDeployment|Redeployment|ResetDeployment|ForceResetDeployment",
    "DeploymentId": "string",
    "GroupVersionId": "string"
}
```

CreateDeploymentRequest

Information about a deployment.

type: object

required: ["DeploymentType"]

DeploymentType

The type of deployment. When used for CreateDeployment, only NewDeployment and Redeployment are valid.

type: string

enum: ["NewDeployment", "Redeployment", "ResetDeployment", "ForceResetDeployment"]

DeploymentId

The ID of the previous deployment you want to redeploy.

type: string

GroupVersionId

The ID of the group version to be deployed.

type: string

Header Parameters

X-Amzn-Client-Token (p. 579)

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Path Parameters

GroupId (p. 571)

The ID of the Greengrass group.
where used: path; required: true

type: string

CLI

```bash
aws greengrass create-deployment \
  --group-id <value> \
  [--deployment-type <value>] \
  [--deployment-id <value>] \
  [--group-version-id <value>] \
  [--amzn-client-token <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "GroupId": "string",
  "DeploymentType": "NewDeployment|Redeployment|ResetDeployment|ForceResetDeployment",
  "DeploymentId": "string",
  "GroupVersionId": "string",
  "AmznClientToken": "string"
}
```

Responses

200

Success. The group was deployed.

CreateDeploymentResponse (p. 592)

```json
{
  "DeploymentArn": "string",
  "DeploymentId": "string"
}
```

CreateDeploymentResponse

  type: object

DeploymentArn

  The ARN of the deployment.

  type: string

DeploymentId

  The ID of the deployment.

  type: string

400

Invalid request.

GeneralError (p. 614)

```json
{
}
```
"Message": "string",
"ErrorDetails": [
  {
    "DetailedErrorCode": "string",
    "DetailedErrorMessage": "string"
  }
]

GeneralError

  General error information.

  type: object

  required: ["Message"]

Message

  A message that contains information about the error.

  type: string

ErrorDetails

  A list of error details.

  type: array

  items: ErrorDetail (p. 600)

ErrorDetail

  Details about the error.

  type: object

DetailedErrorCode

  A detailed error code.

  type: string

DetailedErrorMessage

  A detailed error message.

  type: string

GET

GET /greengrass/groups/GroupId/deployments

Operation ID: ListDeployments (p. 198)

Returns a history of deployments for the group.

Produces: application/json

Path Parameters

GroupId (p. 571)

  The ID of the Greengrass group.
where used: path; required: true

type: string

Query Parameters

MaxResults (p. 573)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

NextToken (p. 573)

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

where used: query; required: false

type: string

CLI

```bash
aws greengrass list-deployments \
   --group-id <value> \
   [--max-results <value>] \
   [--next-token <value>] \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "GroupId": "string",
   "MaxResults": "integer",
   "NextToken": "string"
}
```

Responses

200

Success. The response body contains the list of deployments for the given group.

ListDeploymentsResponse (p. 647)

```json
{
   "Deployments": [
      {
         "GroupArn": "string",
         "DeploymentId": "string",
         "DeploymentArn": "string",
         "DeploymentType": "NewDeployment|Redeployment|ResetDeployment|ForceResetDeployment",
         "CreatedAt": "string"
      }
   ],
```
"NextToken": "string"
}

ListDeploymentsResponse
  type: object
Deploys
  type: array
    items: Deployment (p. 596)
Deployment
  Information about a deployment.
    type: object
GroupArn
  The ARN of the group for this deployment.
    type: string
DeploymentId
  The ID of the deployment.
    type: string
DeploymentArn
  The ARN of the deployment.
    type: string
DeploymentType
  The type of deployment. When used for CreateDeployment, only NewDeployment and Redeployment are valid.
    type: string
    enum: ["NewDeployment", "Redeployment", "ResetDeployment", "ForceResetDeployment"]
CreatedAt
  The time, in milliseconds since the epoch, when the deployment was created.
    type: string
NextToken
  The token for the next set of results, or null if there are no more results.
    in: query
    type: string

400
Invalid request.

GeneralError (p. 614)
POST /greengrass/groups/GroupId/deployments/$reset

Operation ID: ResetDeployments (p. 236)

Resets a group's deployments.

Produces: application/json
Body Parameters

**ResetDeploymentsRequestBody (p. 573)**

Information required to reset deployments.
where used: body; required: true

```json
{
    "Force": true
}
```

**ResetDeploymentsRequest**

Information about a group reset request.
type: object

**Force**

If true, performs a best-effort only core reset.
type: boolean

Header Parameters

**X-Amzn-Client-Token (p. 579)**

A client token used to correlate requests and responses.
where used: header; required: false
type: string

Path Parameters

**GroupId (p. 571)**

The ID of the Greengrass group.
where used: path; required: true
type: string

CLI

```
aws greengrass reset-deployments \
  --group-id <value> \
  [--force | --no-force] \
  [--amzn-client-token <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
    "GroupId": "string",
```
"Force": "boolean",
"AmznClientToken": "string"
}

Responses

200

Success. The group's deployments were reset.

ResetDeploymentsResponse (p. 655)

{
  "DeploymentArn": "string",
  "DeploymentId": "string"
}

ResetDeploymentsResponse

type: object

DeploymentArn

The ARN of the deployment.

type: string

DeploymentId

The ID of the deployment.

type: string

400

Invalid request.

GeneralError (p. 614)

{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string
ErrorDetails
  A list of error details.
  type: array
  items: ErrorDetail (p. 600)
ErrorDetail
  Details about the error.
  type: object
DetailedErrorCode
  A detailed error code.
  type: string
DetailedErrorMessage
  A detailed error message.
  type: string

/greengrass/groups/GroupId/deployments/DeploymentId/status

GET

GET /greengrass/groups/GroupId/deployments/DeploymentId/status

Operation ID: GetDeploymentStatus (p. 131)

Returns the status of a deployment.

Produces: application/json

Path Parameters

GroupId (p. 571)
  The ID of the Greengrass group.
  where used: path; required: true
  type: string

DeploymentId (p. 570)
  The ID of the deployment.
  where used: path; required: true
  type: string

CLI

aws greengrass get-deployment-status \
AWS Greengrass API Reference
/greengrass/groups/GroupId/
deployments/DeploymentId/status

```
--group-id <value> \
--deployment-id <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "GroupId": "string",
  "DeploymentId": "string"
}
```

**Responses**

200

Success. The response body contains the status of the deployment for the group.

GetDeploymentStatusResponse (p. 621)

```
{
  "DeploymentStatus": "string",
  "DeploymentType": "NewDeployment|Redeployment|ResetDeployment|ForceResetDeployment",
  "UpdatedAt": "string",
  "ErrorMessage": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GetDeploymentStatusResponse

Information about the status of a deployment for a group.

- **type**: object
- **DeploymentStatus**
  - The status of the deployment: Building, InProgress, Success, or Failure.
  - **type**: string
- **DeploymentType**
  - The type of deployment. When used for CreateDeployment, only NewDeployment and Redeployment are valid.
  - **type**: string
  - **enum**: ["NewDeployment", "Redeployment", "ResetDeployment", "ForceResetDeployment"]
- **UpdatedAt**
  - The time, in milliseconds since the epoch, when the deployment status was updated.
  - **type**: string
- **ErrorMessage**
  - Error message
type: string

ErrorMessage

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

Example JSON Response:

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)
ErrorDetail
   
   - Details about the error.
   
   - type: object

DetailedErrorCode
   
   - A detailed error code.
   
   - type: string

DetailedErrorMessage
   
   - A detailed error message.
   
   - type: string

/greengrass/groups/GroupId/role

PUT

PUT /greengrass/groups/GroupId/role

Operation ID: AssociateRoleToGroup (p. 3)

Associates a role with a group. Your Greengrass core uses the role to access AWS services. The role's permissions should allow Greengrass Lambda functions and connectors to perform actions against the cloud.

Body Parameters

AssociateRoleToGroupRequestBody (p. 544)

   - where used: body; required: true

   
   
   { 
   "RoleArn": "string"
   }

AssociateRoleToGroupRequest

   - type: object

   - required: ["RoleArn"]

RoleArn

   - The ARN of the role to associate with this group.

   - type: string

Path Parameters

GroupId (p. 571)

   - The ID of the Greengrass group.

   - where used: path; required: true
type: string

**CLI**

```
aws greengrass associate-role-to-group \
  --group-id <value> \
  [--role-arn <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "GroupId": "string",
  "RoleArn": "string"
}
```

**Responses**

**200**

Success.

AssociateRoleToGroupResponse (p. 638)

```
{
  "AssociatedAt": "string"
}
```

**Group**

  type: object

AssociatedAt

  The time, in milliseconds since the epoch, when the role ARN was associated with the group.

  type: string

**400**

Invalid request.

GeneralError (p. 614)

```
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

**GeneralError**

  General error information.

  type: object
required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

500

Server error.

GeneralError (p. 614)

{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.
DELETE

DELETE /greengrass/groups/GroupId/role

Operation ID: DisassociateRoleFromGroup (p. 105)

Disassociates the role from a group.

Path Parameters

GroupId (p. 571)
   The ID of the Greengrass group.
   where used: path; required: true
   type: string

CLI

aws greengrass disassociate-role-from-group \
   --group-id <value> \ 
   [--cli-input-json <value>] \ 
   [--generate-cli-skeleton]

cli-input-json format:

{
   "GroupId": "string"
}

Responses

200
   Success.
### DisassociateRoleFromGroupResponse (p. 599)

```
{
  "DisassociatedAt": "string"
}
```

**DisassociateRoleFromGroupResponse**

- **type**: object

**DisassociatedAt**

- The time, in milliseconds since the epoch, when the role was disassociated from the group.
  - **type**: string

**400**

Invalid request.

### GeneralError (p. 614)

```
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

**GeneralError**

- General error information.
  - **type**: object
    - required: ["Message"]

**Message**

- A message that contains information about the error.
  - **type**: string

**ErrorDetails**

- A list of error details.
  - **type**: array
    - items: **ErrorDetail (p. 600)**

**ErrorDetail**

- Details about the error.
  - **type**: object

**DetailedErrorCode**

- A detailed error code.
  - **type**: string
DetailedErrorMessage

A detailed error message.

type: string

500

Server error.

**GeneralError (p. 614)**

```json
{
  "Message": "string",
  "ErrorDetails": [
  {
    "DetailedErrorCode": "string",
    "DetailedErrorMessage": "string"
  }
  ]
}
```

**GeneralError**

General error information.

type: object

required: ["Message"]

**Message**

A message that contains information about the error.

type: string

**ErrorDetails**

A list of error details.

type: array

items: **ErrorDetail (p. 600)**

**ErrorDetail**

Details about the error.

type: object

**DetailedErrorCode**

A detailed error code.

type: string

**DetailedErrorMessage**

A detailed error message.

type: string

**GET**

GET /greengrass/groups/GroupId/role
Operation ID: GetAssociatedRole (p. 110)

Retrieves the role associated with a group.

**Path Parameters**

**GroupId (p. 571)**

The ID of the Greengrass group.

where used: path; required: true

type: string

**CLI**

```bash
aws greengrass get-associated-role
   --group-id <value>
   [--cli-input-json <value>]
   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "GroupId": "string"
}
```

**Responses**

200

Success.

GetAssociatedRoleResponse (p. 615)

```json
{
   "AssociatedAt": "string",
   "RoleArn": "string"
}
```

GetAssociatedRoleResponse

<table>
<thead>
<tr>
<th>type: object</th>
</tr>
</thead>
</table>

AssociatedAt

The time when the role was associated with the group.

<table>
<thead>
<tr>
<th>type: string</th>
</tr>
</thead>
</table>

RoleArn

The ARN of the role that is associated with the group.

<table>
<thead>
<tr>
<th>type: string</th>
</tr>
</thead>
</table>

400

Invalid request.
GeneralError (p. 614)

```json
{
   "Message": "string",
   "ErrorDetails": [ 
      { 
         "DetailedErrorCode": "string",
         "DetailedErrorMessage": "string"
      }
   ]
}
```

**GeneralError**

General error information.

- **type:** object
- **required:** ["Message"]

**Message**

A message that contains information about the error.

- **type:** string

**ErrorDetails**

A list of error details.

- **type:** array
- **items:** ErrorDetail (p. 600)

**ErrorDetail**

Details about the error.

- **type:** object

**DetailedErrorCode**

A detailed error code.

- **type:** string

**DetailedErrorMessage**

A detailed error message.

- **type:** string

**500**

Server error.

GeneralError (p. 614)

```json
{
   "Message": "string",
   "ErrorDetails": [ 
      { 
         "DetailedErrorCode": "string",
         "DetailedErrorMessage": "string"
      }
   ]
}
```
GeneralError
   General error information.
   type: object
   required: ["Message"]
Message
   A message that contains information about the error.
   type: string
ErrorDetails
   A list of error details.
   type: array
   items: ErrorDetail (p. 600)
ErrorDetail
   Details about the error.
   type: object
DetailedErrorCode
   A detailed error code.
   type: string
DetailedErrorMessage
   A detailed error message.
   type: string

/post/greengrass/groups/GroupId/versions

POST
POST /post/greengrass/groups/GroupId/versions
Operation ID: CreateGroupVersion (p. 54)
Creates a version of a group that has already been defined.
Produces: application/json

Body Parameters
CreateGroupVersionRequestBody (p. 558)
   where used: body; required: true
   
{"CoreDefinitionVersionArn": "string",}
"DeviceDefinitionVersionArn": "string",
"FunctionDefinitionVersionArn": "string",
"SubscriptionDefinitionVersionArn": "string",
"LoggerDefinitionVersionArn": "string",
"ResourceDefinitionVersionArn": "string",
"ConnectorDefinitionVersionArn": "string"
}

GroupVersion

Information about a group version.

type: object

CoreDefinitionVersionArn

The ARN of the core definition version for this group.

type: string

DeviceDefinitionVersionArn

The ARN of the device definition version for this group.

type: string

FunctionDefinitionVersionArn

The ARN of the function definition version for this group.

type: string

SubscriptionDefinitionVersionArn

The ARN of the subscription definition version for this group.

type: string

LoggerDefinitionVersionArn

The ARN of the logger definition version for this group.

type: string

ResourceDefinitionVersionArn

The ARN of the resource definition version for this group.

type: string

ConnectorDefinitionVersionArn

The ARN of the connector definition version for this group.

type: string

Header Parameters

X-Amzn-Client-Token (p. 579)

A client token used to correlate requests and responses.

where used: header; required: false

type: string
Path Parameters

**GroupId (p. 571)**

The ID of the Greengrass group.

where used: path; required: true

type: string

CLI

```bash
aws greengrass create-group-version \
  --group-id <value> \
  [--core-definition-version-arn <value>] \
  [--device-definition-version-arn <value>] \
  [--function-definition-version-arn <value>] \
  [--subscription-definition-version-arn <value>] \
  [--logger-definition-version-arn <value>] \
  [--resource-definition-version-arn <value>] \
  [--connector-definition-version-arn <value>] \
  [--amzn-client-token <value>]  \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "GroupId": "string",
  "CoreDefinitionVersionArn": "string",
  "DeviceDefinitionVersionArn": "string",
  "FunctionDefinitionVersionArn": "string",
  "SubscriptionDefinitionVersionArn": "string",
  "LoggerDefinitionVersionArn": "string",
  "ResourceDefinitionVersionArn": "string",
  "ConnectorDefinitionVersionArn": "string",
  "AmznClientToken": "string"
}
```

Responses

**200 (CreateGroupVersionResponse)**

Success. The response contains information about the group version.

**VersionInformation (p. 676)**

```json
{
  "Arn": "string",
  "Id": "string",
  "Version": "string",
  "CreationTimestamp": "string"
}
```

VersionInformation

Information about a version.

type: object
The ARN of the version.

type: string

The ID of the parent definition that the version is associated with.

type: string

The ID of the version.

type: string

The time, in milliseconds since the epoch, when the version was created.

type: string

Invalid request.

**GeneralError (p. 614)**

```
{
    "Message": "string",
    "ErrorDetails": [
    {
        "DetailedErrorCode": "string",
        "DetailedErrorMessage": "string"
    }
    ]
}
```

**GeneralError**

General error information.

type: object

required: ["Message"]

**Message**

A message that contains information about the error.

type: string

**ErrorDetails**

A list of error details.

type: array

items: **ErrorDetail (p. 600)**

**ErrorDetail**

Details about the error.

type: object
DetailedErrorCode
   A detailed error code.
   type: string
DetailedErrorMessage
   A detailed error message.
   type: string

GET
GET /greengrass/groups/GroupId/versions

Operation ID: ListGroupVersions (p. 214)
Lists the versions of a group.
Produces: application/json

Path Parameters
GroupId (p. 571)
   The ID of the Greengrass group.
   where used: path; required: true
   type: string

Query Parameters
NextToken (p. 573)
   The token for the next set of results, or null if there are no more results.
   where used: query; required: false
   type: string
MaxResults (p. 573)
   The maximum number of results to be returned per request.
   where used: query; required: false
   type: integer

CLI
aws greengrass list-group-versions \
   --group-id <value> \
   [--next-token <value>] \
   [--max-results <value>] \
   [--cli-input-json <value>] \
   [--generate-cli-skeleton]
cli-input-json format:

```json
{
   "GroupId": "string",
   "NextToken": "string",
   "MaxResults": "integer"
}
```

Responses

200 (ListGroupVersionsResponse)

Success. The response contains the list of versions and metadata for the given group.

ListVersionsResponse (p. 650)

```json
{
   "Versions": [
      {
         "Arn": "string",
         "Id": "string",
         "Version": "string",
         "CreationTimestamp": "string"
      }
   ],
   "NextToken": "string"
}
```

ListVersionsResponse

A list of versions.

type: object

Versions

Information about a version.

type: array

items: VersionInformation (p. 676)

VersionInformation

Information about a version.

type: object

Arn

The ARN of the version.

type: string

Id

The ID of the parent definition that the version is associated with.

type: string

Version

The ID of the version.

type: string
CreationTimestamp

The time, in milliseconds since the epoch, when the version was created.

type: string

NextToken

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

type: string

400

Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string
GET /greengrass/groups/GroupId/versions/GroupVersionId

Operation ID: GetGroupVersion (p. 156)

Retrieves information about a group version.

Produces: application/json

Path Parameters

GroupId (p. 571)

The ID of the Greengrass group.

where used: path; required: true

type: string

GroupVersionId (p. 572)

The ID of the group version. This value maps to the Version property of the corresponding VersionInformation object, which is returned by ListGroupVersions requests. If the version is the last one that was associated with a group, the value also maps to the LatestVersion property of the corresponding GroupInformation object.

where used: path; required: true

type: string

CLI

aws greengrass get-group-version \
    --group-id <value> \
    --group-version-id <value> \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]

cli-input-json format:

```json
{
    "GroupId": "string",
    "GroupVersionId": "string"
}
```

Responses

200

Success.

GetGroupVersionResponse (p. 629)
GetGroupVersionResponse

Information about a group version.

type: object

Arn

The ARN of the group version.

type: string

Id

The ID of the group that the version is associated with.

type: string

Version

The ID of the group version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the group version was created.

type: string

Definition

Information about a group version.

type: object

CoreDefinitionVersionArn

The ARN of the core definition version for this group.

type: string

DeviceDefinitionVersionArn

The ARN of the device definition version for this group.

type: string

FunctionDefinitionVersionArn

The ARN of the function definition version for this group.

type: string

LoggerDefinitionVersionArn

The ARN of the logger definition version for this group.

type: string

SubscriptionDefinitionVersionArn

The ARN of the subscription definition version for this group.

type: string

ResourceDefinitionVersionArn

The ARN of the resource definition version for this group.

type: string

ConnectorDefinitionVersionArn

The ARN of the connector definition version for this group.

type: string
type: string
SubscriptionDefinitionVersionArn

The ARN of the subscription definition version for this group.

type: string
LoggerDefinitionVersionArn

The ARN of the logger definition version for this group.

type: string
ResourceDefinitionVersionArn

The ARN of the resource definition version for this group.

type: string
ConnectorDefinitionVersionArn

The ARN of the connector definition version for this group.

400

Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.
type: object

DetailedErrorCode

A detailed error code.
type: string

DetailedErrorMessage

A detailed error message.
type: string

/greengrass/servicerole

PUT

PUT /greengrass/servicerole

Operation ID: AssociateServiceRoleToAccount (p. 6)

Associates a role with your account. AWS IoT Greengrass uses the role to access your Lambda functions and AWS IoT resources. A role is required for deployments to succeed. The role must have at least minimum permissions in the policy AWSGreengrassResourceAccessRolePolicy.

Body Parameters

AssociateServiceRoleToAccountRequestBody (p. 545)

where used: body; required: true

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

AssociateServiceRoleToAccountRequest
type: object

required: ["RoleArn"]

RoleArn

The ARN of the service role to associate with your account.
type: string

CLI

```bash
aws greengrass associate-service-role-to-account \  
  [--role-arn <value>] \  
  [--cli-input-json <value>] \  
  [--generate-cli-skeleton]
```

cli-input-json format:

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

## Responses

### 200

Success.

AssociateServiceRoleToAccountResponse (p. 582)

```json
{
   "AssociatedAt": "string"
}
```

AssociateServiceRoleToAccountResponse
type: object

AssociatedAt

The time when the service role was associated with the account.
type: string

### 400

Invalid request.

GeneralError (p. 614)

```json
{
   "Message": "string",
   "ErrorDetails": [
   {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
   }
   ]
}
```

GeneralError
type: object

required: ["Message"]

Message

A message that contains information about the error.
type: string

ErrorDetails

A list of error details.
type: array

items: ErrorDetail (p. 600)
ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

500

Server error.

GeneralError (p. 614)

```json
{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.
DELETE

DELETE /greengrass/servicerole

Operation ID: DisassociateServiceRoleFromAccount (p. 108)

Disassociates the service role from your account. Without a service role, deployments do not work.

CLI

```bash
aws greengrass disassociate-service-role-from-account  
   [--cli-input-json <value>] 
   [--generate-cli-skeleton]
```

Responses

200

Success.

DisassociateServiceRoleFromAccountResponse (p. 599)

```json
{
   "DisassociatedAt": "string"
}
```

DisassociateServiceRoleFromAccountResponse

type: object

DisassociatedAt

The time when the service role was disassociated from the account.

type: string

500

Server error.

GeneralError (p. 614)

```json
{
   "Message": "string",
   "ErrorDetails": [
      {
         "DetailedErrorCode": "string",
         "DetailedErrorMessage": "string"
      }
   ]
}
```
General Error

- General error information.
  - type: object
    - required: ['Message']
- Message
  - A message that contains information about the error.
    - type: string

Error Details

- A list of error details.
  - type: array
    - items: ErrorDetail (p. 600)

Error Detail

- Details about the error.
  - type: object

Detailed Error Code

- A detailed error code.
  - type: string

Detailed Error Message

- A detailed error message.
  - type: string

**GET**

GET /greengrass/servicerole

Operation ID: GetServiceRoleForAccount (p. 175)

Retrieves the service role that is attached to your account.

CLI

```
aws greengrass get-service-role-for-account  
  [--cli-input-json <value>] 
  [--generate-cli-skeleton]
```

**Responses**

**200**

Success.

GetServiceRoleForAccountResponse (p. 636)

```
{
```
GetServiceRoleForAccountResponse

type: object

AssociatedAt

The time when the service role was associated with the account.

type: string

RoleArn

The ARN of the role associated with the account.

type: string

500

Server error.

GeneralError (p. 614)

{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.
/greengrass/things/ThingName/connectivityInfo

GET

GET /greengrass/things/\{ThingName\}/connectivityInfo

Operation ID: GetConnectivityInfo (p. 115)

Retrieves the connectivity information for a core.

Produces: application/json

Path Parameters

**ThingName (p. 577)**

The thing name.

where used: path; required: true

type: string

CLI

```bash
aws greengrass get-connectivity-info
   --thing-name <value> \n   [--cli-input-json <value>] \n   [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
   "ThingName": "string"
}
```

Responses

**200**

Success.

GetConnectivityInfoResponse (p. 617)

```json
{
   "message": "string",
   "ConnectivityInfo": [
   {
      "Id": "string",
      "HostAddress": "string",
      "PortNumber": 0x01,
```

---

527
Information about a Greengrass core's connectivity.

message

A message about the connectivity info request.

type: string

ConnectivityInfo

Connectivity info list.

type: array

items: ConnectivityInfo (p. 588)

Information about a Greengrass core's connectivity.

required: ["Id", "HostAddress"]

Id

The ID of the connectivity information.

type: string

HostAddress

The endpoint for the Greengrass core. Can be an IP address or DNS address.

type: string

PortNumber

The port of the Greengrass core, usually 8883.

   type: integer

   format: int32

Metadata

Metadata for this endpoint.

   type: string

400

Invalid request.

GeneralError (p. 614)
GeneralError
  General error information.
  type: object
  required: ["Message"]
Message
  A message that contains information about the error.
  type: string
ErrorDetails
  A list of error details.
  type: array
  items: ErrorDetail (p. 600)
ErrorDetail
  Details about the error.
  type: object
DetailedErrorCode
  A detailed error code.
  type: string
DetailedErrorMessage
  A detailed error message.
  type: string

500
  Server error.

GeneralError (p. 614)

{}"Message": "string",
"ErrorDetails": [
  {
    "DetailedErrorCode": "string",
    "DetailedErrorMessage": "string"
  }
]"
type: string

ErrorDetails

A list of error details.
type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.
type: object

DetailedErrorCode

A detailed error code.
type: string

DetailedErrorMessage

A detailed error message.
type: string

**PUT**

PUT /greengrass/things/"ThingName"/connectivityInfo

Operation ID: UpdateConnectivityInfo (p. 246)

Updates the connectivity information for the core. Devices that belong to the group that has this core receive this information to find the location of the core and connect to it.

Produces: application/json

**Body Parameters**

*UpdateConnectivityInfoRequestBody* (p. 577)

The connectivity information.

where used: body; required:

```json
{
   "ConnectivityInfo": [
   {
      "Id": "string",
      "HostAddress": "string",
      "PortNumber": 0x01,
      "Metadata": "string"
   }
   ]
}
```

Information required to update a Greengrass core's connectivity.

*ConnectivityInfo*

A list of connectivity info.
type: array

items: ConnectivityInfo (p. 588)

Information about a Greengrass core's connectivity.

required: ["Id", "HostAddress"]

Id

The ID of the connectivity information.

type: string

HostAddress

The endpoint for the Greengrass core. Can be an IP address or DNS address.

type: string

PortNumber

The port of the Greengrass core, usually 8883.

type: integer

format: int32

Metadata

Metadata for this endpoint.

type: string

Path Parameters

ThingName (p. 577)

The thing name.

where used: path; required: true

type: string

CLI

```
aws greengrass update-connectivity-info \
    --thing-name <value> \
    [--connectivity-info <value>] \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

cli-input-json format:

```
{
    "ThingName": "string",
    "ConnectivityInfo": [
        {
            "Id": "string",
            "HostAddress": "string",
            "PortNumber": "int32-integer",
            "Metadata": "string"
        }
    ]
}
```
Responses

200
Success.

UpdateConnectivityInfoResponse (p. 675)

```json
{
  "message": "string",
  "Version": "string"
}
```

message
A message about the connectivity info update request.
type: string

Version
The new version of the connectivity info.
type: string

400
Invalid request.

GeneralError (p. 614)

```json
{
  "Message": "string",
  "ErrorDetails": [
    {
      "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
    }
  ]
}
```

GeneralError
General error information.
type: object
  required: ["Message"]

Message
A message that contains information about the error.
type: string

ErrorDetails
A list of error details.
type: array
ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

500

Server error.

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail

>ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.
/greengrass/uploads

POST

POST /greengrass/uploads

Operation ID: CreateSoftwareUpdateJob (p. 80)

Creates a software update for a core or group of cores (specified as an IoT thing group.) Use this to update the OTA agent and the Greengrass core software. It uses the AWS IoT Jobs feature, which provides more commands for managing a Greengrass Core software update job.

Produces: application/json

Body Parameters

CreateSoftwareUpdateJobRequestBody (p. 567)

where used: body; required: true

```
{
  "UpdateTargetsArchitecture": "armv6l|armv7l|x86_64|aarch64",
  "UpdateTargets": [
    "string"
  ],
  "SoftwareToUpdate": "core|ota_agent",
  "S3UrlSignerRole": "string",
  "UpdateAgentLogLevel": "NONE|TRACE|DEBUG|VERBOSE|INFO|WARN|ERROR|FATAL",
  "UpdateTargetsOperatingSystem": "ubuntu|raspbian|amazon_linux|openwrt"
}
```

CreateSoftwareUpdateJobRequest

Request for the CreateSoftwareUpdateJob API.

type: object


UpdateTargetsArchitecture

The architecture of the cores that are the targets of an update.

type: string

enum: ["armv6l", "armv7l", "x86_64", "aarch64"]

UpdateTargets

The ARNs of the targets (IoT things or IoT thing groups) that this update is applied to.

type: array
SoftwareToUpdate

The piece of software on the Greengrass core that will be updated.

type: string
enum: ["core", "ota_agent"]

S3UrlSignerRole

The IAM role that Greengrass uses to create presigned URLs that point to the update artifact.

type: string

UpdateAgentLogLevel

The minimum level of log statements that should be logged by the OTA agent during an update.

type: string
enum: ["NONE", "TRACE", "DEBUG", "VERBOSE", "INFO", "WARN", "ERROR", "FATAL"]

UpdateTargetsOperatingSystem

The operating system of the cores that are the targets of an update.

type: string
enum: ["ubuntu", "raspbian", "amazon_linux", "openwrt"]

Header Parameters

X-Amzn-Client-Token (p. 579)

A client token used to correlate requests and responses.

where used: header; required: false

type: string

CLI

```
aws greengrass create-software-update-job \
  [--update-targets-architecture <value>] \
  [--update-targets <value>] \
  [--software-to-update <value>] \
  [--s3-url-signer-role <value>] \
  [--update-agent-log-level <value>] \
  [--update-targets-operating-system <value>] \
  [--amzn-client-token <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "UpdateTargetsArchitecture": "armv6l|armv7l|x86_64|aarch64",
  "UpdateTargets": [ 
    "string"
  ],
  "SoftwareToUpdate": "core|ota_agent",
}
```
"S3UrlSignerRole": "string",
"UpdateAgentLogLevel": "NONE|TRACE|DEBUG|VERBOSE|INFO|WARN|ERROR|FATAL",
"UpdateTargetsOperatingSystem": "ubuntu|raspbian|amazon_linux|openwrt",
"AmznClientToken": "string"
}

**Responses**

**200**

Success.

CreateSoftwareUpdateJobResponse (p. 594)

```json
{
    "IotJobId": "string",
    "IotJobArn": "string",
    "PlatformSoftwareVersion": "string"
}
```

CreateSoftwareUpdateJobResponse

  type: object

  IotJobId

    The IoT job ID that corresponds to this update.

    type: string

  IotJobArn

    The IoT job ARN that corresponds to this update.

    type: string

  PlatformSoftwareVersion

    The software version installed on the device or devices after the update.

    type: string

**400**

Invalid request.

GeneralError (p. 614)

```json
{
    "Message": "string",
    "ErrorDetails": [
    {
        "DetailedErrorCode": "string",
        "DetailedErrorMessage": "string"
    }
    ]
}
```

GeneralError

  General error information.

  type: object
required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

500

Server error.

GeneralError (p. 614)

```
{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}
```

GeneralError

General error information.

type: object

required: ["Message"]

Message

A message that contains information about the error.

type: string

ErrorDetails

A list of error details.
type: array
items: ErrorDetail (p. 600)

ErrorDetail
Details about the error.
type: object

DetailedErrorCode
A detailed error code.
type: string

DetailedErrorMessage
A detailed error message.
type: string

/tags/resource-arn
GET

GET /tags/resource-arn

Operation ID: ListTagsForResource (p. 234)

Lists tags for a Greengrass resource. Valid resources are Group, ConnectorDefinition, CoreDefinition, DeviceDefinition, FunctionDefinition, LoggerDefinition, ResourceDefinition, SubscriptionDefinition, and BulkDeployment.

Produces: application/json

CLI

```bash
aws greengrass list-tags-for-resource \
  --resource-arn <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```json
{
  "ResourceArn": "string"
}
```

Parameters:

ResourceArn (p. 574)

The Amazon Resource Name (ARN) of the resource whose tags you want to retrieve.

where used: path; required: true

type: string
Responses

200

HTTP Status Code 200: OK.

tags (p. 678)

```
{
    "tags": {
        "keyName0": "value0",
        "keyName1": "value1",
        "keyName2": "value2"
    }
}
```

The resource tags.

**type:** object

**additionalProperties:** The key-value pair for the resource tag. Type: string

400

Invalid request.

**GeneralError (p. 614)**

```
{
    "Message": "string",
    "ErrorDetails": [
        {
            "DetailedErrorCode": "string",
            "DetailedErrorMessage": "string"
        }
    ]
}
```

**GeneralError**

General error information.

**type:** object

**required:** ["Message"]

**Message**

A message that contains information about the error.

**type:** string

**ErrorDetails**

A list of error details.

**type:** array

**items:** **ErrorDetail (p. 600)**

**ErrorDetail**

Details about the error.
AWS Greengrass API Reference
(tags/resource-arn)

POST

POST /tags/resource-arn

Operation ID: TagResource (p. 243)

Adds tags to a Greengrass resource. Valid resources are Group, ConnectorDefinition, CoreDefinition, DeviceDefinition, FunctionDefinition, LoggerDefinition, ResourceDefinition, SubscriptionDefinition, and BulkDeployment.

Produces: application/json

CLI

aws greengrass tag-resource
   --resource-arn <value>
   [--tags <value>]
   [--cli-input-json <value>]
   [--generate-cli-skeleton]

cli-input-json format:

{
   "ResourceArn": "string",
   "tags": {
      "additionalProperty0": "string",
      "additionalProperty1": "string",
      "additionalProperty2": "string"
   }
}

Parameters:

ResourceArn (p. 574)

The Amazon Resource Name (ARN) of the resource to attach the tags to.

where used: path; required: true

type: string

tags (p. 576)

A map of the key-value pairs for the resource tag.

where used: body; required: true

type: tags
**Responses**

**204**

HTTP Status Code 204: Successful response.

**400**

Invalid request.

**GeneralError (p. 614)**

```json
{
   "Message": "string",
   "ErrorDetails": [
      {
         "DetailedErrorCode": "string",
         "DetailedErrorMessage": "string"
      }
   ]
}
```

**GeneralError**

General error information.

type: object

required: ["Message"]

**Message**

A message that contains information about the error.

type: string

**ErrorDetails**

A list of error details.

type: array

items: **ErrorDetail (p. 600)**

**ErrorDetail**

Details about the error.

type: object

**DetailedErrorCode**

A detailed error code.

type: string

**DetailedErrorMessage**

A detailed error message.

type: string

**DELETE**

DELETE /tags/resource-arn
Operation ID: UntagResource (p. 245)

Removes tags from a Greengrass resource. Valid resources are Group, ConnectorDefinition, CoreDefinition, DeviceDefinition, FunctionDefinition, LoggerDefinition, ResourceDefinition, SubscriptionDefinition, and BulkDeployment.

Produces: application/json

CLI

```
aws greengrass untag-resource  \
   --resource-arn <value>  \
   --tag-keys <value>  \
   [{--cli-input-json <value>}]  \
   [--generate-cli-skeleton]
```

cli-input-json format:

```
{
   "ResourceArn": "string",
   "TagKeys": [
      "string"
   ]
}
```

Parameters:

TagKeys (p. 577)

An array of tag keys to delete.

where used: query; required: true

type: array of strings

ResourceArn (p. 574)

The Amazon Resource Name (ARN) of the resource to remove the tags from.

where used: path; required: true

type: string

Responses

204

HTTP Status Code 204: Successful response.

400

Invalid request.

GenericError (p. 614)

```
{
   "Message": "string",
   "ErrorDetails": [
      {
         "DetailedErrorCode": "string",
      }
   ]
}
```
Parameters

The following parameters are used:

- `AssociateRoleToGroupRequestBody` (p. 544)
- `AssociateServiceRoleToAccountRequestBody` (p. 545)
- `BulkDeploymentId` (p. 545)
- `CertificateAuthorityId` (p. 546)
- `ConnectorDefinitionId` (p. 546)
- `ConnectorDefinitionVersionId` (p. 546)
- `CoreDefinitionId` (p. 547)
- `CoreDefinitionVersionId` (p. 547)
- `CreateConnectorDefinitionRequestBody` (p. 547)
- `CreateConnectorDefinitionVersionRequestBody` (p. 548)
- `CreateCoreDefinitionRequestBody` (p. 549)
AssociateRoleToGroupRequestBody

```json
{
  "RoleArn": "string"
}
```
AssociateRoleToGroupRequestBody

```json
{}
```

*AssociateRoleToGroupRequestBody*

in: body
required: true

schema: *AssociateRoleToGroupRequest (p. 582)*

*AssociateRoleToGroupRequest*

type: object
required: ["RoleArn"]

*RoleArn*

The ARN of the role to associate with this group.

type: string

---

**AssociateServiceRoleToAccountRequestBody**

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

*AssociateServiceRoleToAccountRequestBody*

in: body
required: true

schema: *AssociateServiceRoleToAccountRequest (p. 582)*

*AssociateServiceRoleToAccountRequest*

type: object
required: ["RoleArn"]

*RoleArn*

The ARN of the service role to associate with your account.

type: string

---

**BulkDeploymentId**

```json
{
   "BulkDeploymentId": "string"
}
```

*BulkDeploymentId*

The ID of the bulk deployment.

in: path
CertificateAuthorityId

```
{
  "CertificateAuthorityId": "string"
}
```

CertificateAuthorityId

The ID of the certificate authority.

in: path

required: true

type: string

ConnectorDefinitionId

```
{
  "ConnectorDefinitionId": "string"
}
```

ConnectorDefinitionId

The ID of the connector definition.

in: path

required: true

type: string

ConnectorDefinitionVersionId

```
{
  "ConnectorDefinitionVersionId": "string"
}
```

ConnectorDefinitionVersionId

The ID of the connector definition version. This value maps to the Version property of the corresponding VersionInformation object, which is returned by ListConnectorDefinitionVersions requests. If the version is the last one that was associated with a connector definition, the value also maps to the LatestVersion property of the corresponding DefinitionInformation object.

in: path

required: true
**CoreDefinitionId**

```json
{"CoreDefinitionId": "string"}
```

CoreDefinitionId

The ID of the core definition.

- in: path
- required: true
- type: string

**CoreDefinitionVersionId**

```json
{"CoreDefinitionVersionId": "string"}
```

CoreDefinitionVersionId

The ID of the core definition version. This value maps to the `Version` property of the corresponding `VersionInformation` object, which is returned by `ListCoreDefinitionVersions` requests. If the version is the last one that was associated with a core definition, the value also maps to the `LatestVersion` property of the corresponding `DefinitionInformation` object.

- in: path
- required: true
- type: string

**CreateConnectorDefinitionRequestBody**

```json
{"Name": "string",
 "InitialVersion": {
   "Connectors": [
     {
       "Id": "string",
       "ConnectorArn": "string",
       "Parameters": {
         "additionalProperty0": "string",
         "additionalProperty1": "string",
         "additionalProperty2": "string"
       }
     }
   ],
   "tags": {  
   }
 },
 "tags": {
```
CreateConnectorDefinitionRequestBody

in: body
required: true

properties

Name: The name of the connector definition. Type: string
InitialVersion: Information about the initial version of the connector definition. Type: ConnectorDefinitionVersion
tags: The tags to attach to the new resource. Type: tags

CreateConnectorDefinitionVersionRequestBody

in: body
required: true
schema: ConnectorDefinitionVersion (p. 589)

ConnectorDefinitionVersion

Information about the connector definition version, which is a container for connectors.
type: object

Connectors

A list of references to connectors in this version, with their corresponding configuration settings.
type: array
items: Connector (p. 588)

Connector

Information about a connector. Connectors run on the Greengrass core and contain built-in integration with local infrastructure, device protocols, AWS, and other cloud services.
CreateCoreDefinitionRequestBody

Information required to create a core definition.

in: body
required: true

properties

Name: The name of the core definition. Type: string

InitialVersion: Information about the initial version of the core definition. Type: CoreDefinitionVersion

tags: The tags to attach to the new resource. Type: tags
CreateCoreDefinitionVersionRequestBody

```json
{
   "Cores": [
      {
         "Id": "string",
         "ThingArn": "string",
         "CertificateArn": "string",
         "SyncShadow": true
      }
   ]
}
```

CreateCoreDefinitionVersionRequestBody

- **in**: body
- **required**: true
- **schema**: CoreDefinitionVersion (p. 591)

CoreDefinitionVersion

Information about a core definition version.

- **type**: object

Cores

A list of cores in the core definition version.

- **type**: array

  - **items**: Core (p. 590)

Core

Information about a core.

- **type**: object

  - **required**: ["Id", "ThingArn", "CertificateArn"]

Id

A descriptive or arbitrary ID for the core. This value must be unique within the core definition version. Maximum length is 128 characters with the pattern `[a-zA-Z0-9_:]+`.

- **type**: string

ThingArn

The ARN of the thing that is the core.

- **type**: string

CertificateArn

The ARN of the certificate associated with the core.

- **type**: string

SyncShadow

If true, the core's local shadow is synced with the cloud automatically.
CreateDeploymentRequestBody

```json
{
  "DeploymentType": "NewDeployment|Redeployment|ResetDeployment|ForceResetDeployment",
  "DeploymentId": "string",
  "GroupVersionId": "string"
}
```

CreateDeploymentRequestBody

- in: body
- required: true
- schema: CreateDeploymentRequest (p. 592)

CreateDeploymentRequest

Information about a deployment.

- type: object
- required: ["DeploymentType"]

DeploymentType

The type of deployment. When used for CreateDeployment, only NewDeployment and Redeployment are valid.

- type: string
- enum: ["NewDeployment", "Redeployment", "ResetDeployment", "ForceResetDeployment"]

DeploymentId

The ID of the previous deployment you want to redeploy.

- type: string

GroupVersionId

The ID of the group version to be deployed.

- type: string

CreateDeviceDefinitionRequestBody

```json
{
  "Name": "string",
  "InitialVersion": {
    "Devices": [
      {
        "Id": "string",
        "ThingArn": "string",
        "CertificateArn": "string",
        "SyncShadow": true
      }
    ]
  }
}
```
"tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
}

CreateDeviceDefinitionRequestBody

Information required to create a device definition.

in: body

required: true

properties

Name: The name of the device definition. Type: string

InitialVersion: Information about the initial version of the device definition. Type: DeviceDefinitionVersion

tags: The tags to attach to the new resource. Type: tags

CreateDeviceDefinitionVersionRequestBody

{
    "Devices": [
        {
            "Id": "string",
            "ThingArn": "string",
            "CertificateArn": "string",
            "SyncShadow": true
        }
    ]
}

CreateDeviceDefinitionVersionRequestBody

in: body

required: true

schema: DeviceDefinitionVersion (p. 598)

DeviceDefinitionVersion

Information about a device definition version.

type: object

Devices

A list of devices in the definition version.

type: array

items: Device (p. 597)

Device

Information about a device.
CreateFunctionDefinitionRequestBody

```json
{
  "Name": "string",
  "InitialVersion": {
    "DefaultConfig": {
      "Execution": {
        "IsolationMode": "GreengrassContainer|NoContainer",
        "RunAs": {
          "Uid": 1001,
          "Gid": 1002
        }
      }
    },
    "Functions": [
      {
        "Id": "string",
        "FunctionArn": "string",
        "FunctionConfiguration": {
          "Pinned": true,
          "Executable": "string",
          "ExecArgs": "string",
          "MemorySize": 0,
          "Timeout": 0,
          "EncodingType": "binary|json",
          "Environment": {
            "Variables": {
              "additionalProperty0": "string",
              "additionalProperty1": "string",
              "additionalProperty2": "string"
            }
          }
        }
      }
    }
  }
}
```
CreateFunctionDefinitionVersionRequestBody

```json
{
    "DefaultConfig": {
        "Execution": {
            "IsolationMode": "GreengrassContainer|NoContainer",
            "RunAs": {
                "Uid": 1001,
                "Gid": 1002
            }
        }
    },
    "Functions": [
        {
            "Id": "string",
            "FunctionArn": "string",
            "FunctionConfiguration": {
                "Pinned": true,
                "Executable": "string",
                "ExecArgs": "string",
                "MemorySize": 0,
                "Timeout": 0,
                "EncodingType": "binary|json",
                "Environment": {
                    "Variables": {
```
CreateFunctionDefinitionVersionRequestBody

Information about the function definition version.

in: body
required: true
schema: FunctionDefinitionVersion (p. 609)

FunctionDefinitionVersion

Information about a function definition version.

type: object

DefaultConfig

The default configuration that applies to all Lambda functions in the group. Individual Lambda functions can override these settings.

type: object

Execution

Configuration information that specifies how a Lambda function runs.

Functions

A list of Lambda functions in this function definition version.

type: array

items: Function (p. 601)

Information about a Lambda function.

type: object

required: ["Id"]
Id

A descriptive or arbitrary ID for the function. This value must be unique within the function
definition version. Maximum length is 128 characters with the pattern `[a-zA-Z0-9_:\-]+`.

type: string

FunctionArn

The ARN of the alias (recommended) or version of the target Lambda function.

type: string

FunctionConfiguration

The configuration of the Lambda function.

type: object

Pinned

True if the function is pinned. Pinned means the function is long-lived and starts when the core starts.

type: boolean

Executable

The name of the function executable.

type: string

ExecArgs

The execution arguments.

type: string

MemorySize

The memory size, in KB, required by the function. This setting does not apply and should be cleared when you run the Lambda function without containerization.

type: integer

Timeout

The allowed function execution time, after which Lambda should terminate the function. This timeout still applies to pinned Lambda functions for each request.

type: integer

EncodingType

The expected encoding type of the input payload for the function. The default is `json`.

type: string

enum: ["binary", "json"]

Environment

The environment configuration of the function.

type: object

Variables

Environment variables for the Lambda function’s configuration.
type: object

additionalProperties: An object with properties of type string that represent the environment variables.

ResourceAccessPolicies

A list of the resources, with their permissions, to which the Lambda function is granted access. A Lambda function can have at most 10 resources. ResourceAccessPolicies applies only when you run the Lambda function in a Greengrass container.

type: array

items: ResourceAccessPolicy (p. 660)

ResourceAccessPolicy

A policy used by the function to access a resource.

type: object

required: ["ResourceId"]

ResourceId

The ID of the resource. (This ID is assigned to the resource when you create the resource definition.)

type: string

Permission

The type of permission a function has to access a resource.

type: string

enum: ["ro", "rw"]

AccessSysfs

If true, the Lambda function is allowed to access the host's /sys folder. Use this when the Lambda function needs to read device information from /sys. This setting applies only when you run the Lambda function in a Greengrass container.

type: boolean

Execution

Configuration information that specifies how a Lambda function runs.

type: object

IsolationMode

Specifies whether the Lambda function runs in a Greengrass container (default) or without containerization. Unless your scenario requires that you run without containerization, we recommend that you run in a Greengrass container. Omit this value to run the Lambda function with the default containerization for the group.

type: string

enum: ["GreengrassContainer", "NoContainer"]

RunAs

Specifies the user and group whose permissions are used when running the Lambda function. You can specify one or both values to override the default values. To minimize the risk of unintended changes or malicious attacks, we recommend that you avoid running as root unless absolutely
necessary. To run as root, you must update config.json in greengrass-root/config to set allowFunctionsToRunAsRoot to yes.

type: object

Uid
The user ID whose permissions are used to run a Lambda function.

type: integer

Gid
The group ID whose permissions are used to run a Lambda function.

type: integer

CreateGroupRequestBody

```json
{
  "Name": "string",
  "InitialVersion": {
    "CoreDefinitionVersionArn": "string",
    "DeviceDefinitionVersionArn": "string",
    "FunctionDefinitionVersionArn": "string",
    "SubscriptionDefinitionVersionArn": "string",
    "LoggerDefinitionVersionArn": "string",
    "ResourceDefinitionVersionArn": "string",
    "ConnectorDefinitionVersionArn": "string"
  },
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  }
}
```

CreateGroupRequestBody

in: body

required: true

properties

Name: The name of the group. Type: string

InitialVersion: Information about the initial version of the group. Type: GroupVersion

tags: The tags to attach to the new resource. Type: tags

CreateGroupVersionRequestBody

```json
{
  "CoreDefinitionVersionArn": "string",
  "DeviceDefinitionVersionArn": "string",
  "FunctionDefinitionVersionArn": "string",
  "SubscriptionDefinitionVersionArn": "string",
  "LoggerDefinitionVersionArn": "string",
  "ResourceDefinitionVersionArn": "string",
  "ConnectorDefinitionVersionArn": "string"
}
```
CreateLoggerDefinitionRequestBody

{
    "Name": "string",
    "InitialVersion": {
        "Loggers": [
            {
                "Id": "string",
                "..."  
            }
        ]
    }
}

CreateGroupVersionRequestBody

in: body
required: true
schema: GroupVersion (p. 641)

GroupVersion

Information about a group version.
type: object

CoreDefinitionVersionArn

The ARN of the core definition version for this group.
type: string

DeviceDefinitionVersionArn

The ARN of the device definition version for this group.
type: string

FunctionDefinitionVersionArn

The ARN of the function definition version for this group.
type: string

SubscriptionDefinitionVersionArn

The ARN of the subscription definition version for this group.
type: string

LoggerDefinitionVersionArn

The ARN of the logger definition version for this group.
type: string

ResourceDefinitionVersionArn

The ARN of the resource definition version for this group.
type: string

ConnectorDefinitionVersionArn

The ARN of the connector definition version for this group.
type: string
CreateLoggerDefinitionVersionRequestBody

```json
{
    "Loggers": [
        {
            "Id": "string",
            "Type": "FileSystem|AWSCloudWatch",
            "Component": "GreengrassSystem|Lambda",
            "Level": "DEBUG|INFO|WARN|ERROR|FATAL",
            "Space": 0
        }
    ]
}
```

CreateLoggerDefinitionRequestBody

```json
in: body
required: true
properties
Name: The name of the logger definition. Type: string
InitialVersion: Information about the initial version of the logger definition. Type: LoggerDefinitionVersion
tags: The tags to attach to the new resource. Type: tags
```

CreateLoggerDefinitionVersionRequestBody

```json
CreateLoggerDefinitionVersionRequestBody
in: body
required: true
schema: LoggerDefinitionVersion (p. 653)
LoggerDefinitionVersion
Information about a logger definition version.
type: object
Loggers
A list of loggers.
type: array
```
CreateResourceDefinitionRequestBody

```json
{
    "Name": "string",
    "InitialVersion": {
        "Resources": [
            {
                "Id": "string",
                "Name": "string",
                "ResourceDataContainer": {
                    "*-- NOTE:": "choose one of the following --*",
                    "LocalDeviceResourceData": {
                        "SourcePath": "string",
                        "GroupOwnerSetting": {
                            "AutoAddGroupOwner": true,
                            "GroupOwner": "string"
                        }
                    },
                    "LocalVolumeResourceData": {
                        "SourcePath": "string",
                        "DestinationPath": "string",
                        "GroupOwnerSetting": {
                            "AutoAddGroupOwner": true,
                            "GroupOwner": "string"
                        }
                    }
                }
            }
        ]
    }
}
```
CreateResourceDefinitionRequestBody

in: body
required: true
properties

Name: The name of the resource definition. Type: string

InitialVersion: Information about the initial version of the resource definition. Type: ResourceDefinitionVersion

tags: The tags to attach to the new resource. Type: tags

CreateResourceDefinitionVersionRequestBody

{
  "Resources": [
    {
      "Id": "string",
      "Name": "string",
      "ResourceDataContainer": {
        "-- NOTE:": "choose one of the following --",
        "LocalDeviceResourceData": {
          "SourcePath": "string",
          "GroupOwnerSetting": {
            "GroupOwner": "string",
            "GroupPermission": "ro|rw"
          }
        }
      }
    }
  ],
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  }
}
CreateResourceDefinitionVersionRequestBody

Information about the resource definition version.

in: body
required: true

schema: ResourceDefinitionVersion (p. 664)

ResourceDefinitionVersion

Information about a resource definition version.

type: object

Resources

A list of resources.

type: array

items: Resource (p. 656)

Resource

Information about a resource.
type: object

required: ["Id", "Name", "ResourceDataContainer"]

Id

The resource ID, used to refer to a resource in the Lambda function configuration. Maximum length is 128 characters with the pattern `[a-zA-Z0-9_:.-]`. This must be unique within a Greengrass group.

type: string

Name

The descriptive resource name, which is displayed on the AWS IoT Greengrass console. Maximum length is 128 characters with the pattern `[a-zA-Z0-9_:.-]`. This must be unique within a Greengrass group.

type: string

ResourceDataContainer

A container for resource data. The container takes only one of the following supported resource data types: LocalDeviceResourceData, LocalVolumeResourceData, SageMakerMachineLearningModelResourceData, S3MachineLearningModelResourceData, SecretsManagerSecretResourceData.

type: object

NOTE: oneOf

LocalDeviceResourceData

Attributes that define a local device resource.

type: object

SourcePath

The local absolute path of the device resource. The source path for a device resource can refer only to a character device or block device under `/dev`.

type: string

GroupOwnerSetting

Group owner related settings for local resources.

type: object

AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.

type: boolean

GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.

type: string

LocalVolumeResourceData

Attributes that define a local volume resource.
type: object

SourcePath

The local absolute path of the volume resource on the host. The source path for a volume resource type cannot start with `/sys`.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

GroupOwnerSetting

Group owner related settings for local resources.

AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.

type: boolean

GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.

type: string

SageMakerMachineLearningModelResourceData

Attributes that define an Amazon SageMaker machine learning resource.

type: object

SageMakerJobArn

The ARN of the Amazon SageMaker training job that represents the source model.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.

type: object

required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string
GroupPermission

The permissions that the group owner has to the resource. Valid values are rw (read-write) or ro (read-only).

type: string
enum: ["ro", "rw"]

S3MachineLearningModelResourceData

Attributes that define an Amazon S3 machine learning resource.

type: object

S3Uri

The URI of the source model in an S3 bucket. The model package must be in tar.gz or .zip format.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.

type: object

required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are rw (read-write) or ro (read-only).

type: string
enum: ["ro", "rw"]

SecretsManagerSecretResourceData

Attributes that define a secret resource, which references a secret from AWS Secrets Manager. AWS IoT Greengrass stores a local, encrypted copy of the secret on the Greengrass core, where it can be securely accessed by connectors and Lambda functions.

type: object

ARN

The ARN of the Secrets Manager secret to make available on the core. The value of the secret's latest version (represented by the AWSCURRENT staging label) is included by default.

type: string
AdditionalStagingLabelsToDownload

Optional. The staging labels in addition to AWSCURRENT whose values you want to make available on the core.

type: array

CreateSoftwareUpdateJobRequestBody

```json
{
  "UpdateTargetsArchitecture": "armv6l|armv7l|x86_64|aarch64",
  "UpdateTargets": [
    "string"
  ],
  "SoftwareToUpdate": "core|ota_agent",
  "S3UrlSignerRole": "string",
  "UpdateAgentLogLevel": "NONE|TRACE|DEBUG|VERBOSE|INFO|WARN|ERROR|FATAL",
  "UpdateTargetsOperatingSystem": "ubuntu|raspbian|amazon_linux|openwrt"
}
```

CreateSoftwareUpdateJobRequestBody

in: body
required: true

schema: CreateSoftwareUpdateJobRequest (p. 593)

CreateSoftwareUpdateJobRequest

Request for the CreateSoftwareUpdateJob API.

type: object

"UpdateTargetsOperatingSystem"]

UpdateTargetsArchitecture

The architecture of the cores that are the targets of an update.

type: string

enum: ["armv6l", "armv7l", "x86_64", "aarch64"]

UpdateTargets

The ARNs of the targets (IoT things or IoT thing groups) that this update is applied to.

type: array

SoftwareToUpdate

The piece of software on the Greengrass core that will be updated.

type: string

enum: ["core", "ota_agent"]

S3UrlSignerRole

The IAM role that Greengrass uses to create presigned URLs that point to the update artifact.
CreateSubscriptionDefinitionRequestBody

```json
{
  "Name": "string",
  "InitialVersion": {
    "Subscriptions": [
      {
        "Id": "string",
        "Source": "string",
        "Subject": "string",
        "Target": "string"
      }
    ],
    "tags": {
      "additionalProperty0": "string",
      "additionalProperty1": "string",
      "additionalProperty2": "string"
    }
  }
}
```

CreateSubscriptionDefinitionRequestBody

- `in`: body
- `required`: true
- `properties`
  - `Name`: The name of the subscription definition. Type: string
  - `InitialVersion`: Information about the initial version of the subscription definition. Type: SubscriptionDefinitionVersion
  - `tags`: The tags to attach to the new resource. Type: tags

CreateSubscriptionDefinitionVersionRequestBody

```json
{
  "Subscriptions": [
    {
    ...
    
```
CreateSubscriptionDefinitionVersionRequestBody

in: body
required: true

schema: SubscriptionDefinitionVersion (p. 673)

SubscriptionDefinitionVersion

Information about a subscription definition version.

type: object

Subscriptions

A list of subscriptions.

type: array

items: Subscription (p. 672)

Subscription

Information about a subscription.

type: object

required: ["Id", "Source", "Subject", "Target"]

Id

A descriptive or arbitrary ID for the subscription. This value must be unique within the subscription definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9_:\-\_]+.

type: string

Source

The source of the subscription. Can be a thing ARN, the ARN of a Lambda function alias (recommended) or version, a connector ARN, 'cloud' (which represents AWS IoT), or 'GGShadowService'. If you specify a Lambda function, this ARN should match the ARN used to add the function to the Greengrass group.

type: string

Subject

The MQTT topic used to route the message.

type: string

Target

Where the message is sent. Can be a thing ARN, the ARN of a Lambda function alias (recommended) or version, a connector ARN, 'cloud' (which represents AWS IoT), or 'GGShadowService'. If you specify
a Lambda function, this ARN should match the ARN used to add the function to the Greengrass group.

type: string

DeploymentId

```json
{
  "DeploymentId": "string"
}
```

**DeploymentId**

The ID of the deployment.

- in: path
- required: true
- type: string

DeviceDefinitionId

```json
{
  "DeviceDefinitionId": "string"
}
```

**DeviceDefinitionId**

The ID of the device definition.

- in: path
- required: true
- type: string

DeviceDefinitionVersionId

```json
{
  "DeviceDefinitionVersionId": "string"
}
```

**DeviceDefinitionVersionId**

The ID of the device definition version. This value maps to the Version property of the corresponding VersionInformation object, which is returned by ListDeviceDefinitionVersions requests. If the version is the last one that was associated with a device definition, the value also maps to the LatestVersion property of the corresponding DefinitionInformation object.

- in: path
- required: true
FunctionDefinitionId

```json
{"FunctionDefinitionId": "string"}
```

FunctionDefinitionId

The ID of the function definition.

in: path

required: true

type: string

FunctionDefinitionVersionId

```json
{"FunctionDefinitionVersionId": "string"}
```

FunctionDefinitionVersionId

The ID of the function definition version. This value maps to the `Version` property of the corresponding `VersionInformation` object, which is returned by `ListFunctionDefinitionVersions` requests. If the version is the last one that was associated with a function definition, the value also maps to the `LatestVersion` property of the corresponding `DefinitionInformation` object.

in: path

required: true

type: string

GroupId

```json
{"GroupId": "string"}
```

GroupId

The ID of the Greengrass group.

in: path

required: true

type: string
**GroupVersionId**

```json
{
  "GroupVersionId": "string"
}
```

**GroupVersionId**

The ID of the group version. This value maps to the `Version` property of the corresponding `VersionInformation` object, which is returned by `ListGroupVersions` requests. If the version is the last one that was associated with a group, the value also maps to the `LatestVersion` property of the corresponding `GroupInformation` object.

- in: path
- required: true
- type: string

**LoggerDefinitionId**

```json
{
  "LoggerDefinitionId": "string"
}
```

**LoggerDefinitionId**

The ID of the logger definition.

- in: path
- required: true
- type: string

**LoggerDefinitionVersionId**

```json
{
  "LoggerDefinitionVersionId": "string"
}
```

**LoggerDefinitionVersionId**

The ID of the logger definition version. This value maps to the `Version` property of the corresponding `VersionInformation` object, which is returned by `ListLoggerDefinitionVersions` requests. If the version is the last one that was associated with a logger definition, the value also maps to the `LatestVersion` property of the corresponding `DefinitionInformation` object.

- in: path
- required: true
- type: string
MaxResults

```json
{
    "MaxResults": 0
}
```

MaxResults

The maximum number of results to be returned per request.

- in: query
- required: false
- min: 1
- type: integer

NextToken

```json
{
    "NextToken": "string"
}
```

NextToken

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

- in: query
- required: false
- type: string

ResetDeploymentsRequestBody

```json
{
    "Force": true
}
```

ResetDeploymentsRequestBody

Information required to reset deployments.

- in: body
- required: true

- schema: ResetDeploymentsRequest (p. 655)

ResetDeploymentsRequest

Information about a group reset request.

- type: object
Force

If true, performs a best-effort only core reset.

type: boolean

**ResourceArn**

```json
{  "ResourceArn": "string"
}
```

**ResourceArn**

The Amazon Resource Name (ARN) of the target resource for tagging operations.

in: path
required: true
type: string

**ResourceDefinitionId**

```json
{  "ResourceDefinitionId": "string"
}
```

**ResourceDefinitionId**

The ID of the resource definition.

in: path
required: true
type: string

**ResourceDefinitionVersionId**

```json
{  "ResourceDefinitionVersionId": "string"
}
```

**ResourceDefinitionVersionId**

The ID of the resource definition version. This value maps to the `Version` property of the corresponding `VersionInformation` object, which is returned by `ListResourceDefinitionVersions` requests. If the version is the last one that was associated with a resource definition, the value also maps to the `LatestVersion` property of the corresponding `DefinitionInformation` object.

in: path
StartBulkDeploymentRequestBody

```json
{
  "InputFileUri": "string",
  "ExecutionRoleArn": "string",
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  }
}
```

Information about a bulk deployment. You cannot start a new bulk deployment while another one is still running or in a non-terminal state.

type: object

required: ['InputFileUri', 'ExecutionRoleArn']

**InputFileUri**

The URI of the input file in the S3 bucket. The execution role must have `getObject` permissions on this bucket to access the input file. The input file is a JSON-serialized, line-delimited file with UTF-8 encoding that provides a list of group and version IDs and the deployment type. This file must be less than 100 MB. Currently, AWS IoT Greengrass supports only `NewDeployment` deployment types.

```
{"GroupId": "groupId1", "GroupVersionId": "groupVersionId1", "DeploymentType": "NewDeployment"}
{"GroupId": "groupId2", "GroupVersionId": "groupVersionId2", "DeploymentType": "NewDeployment"}
{"GroupId": "groupId3", "GroupVersionId": "groupVersionId3", "DeploymentType": "NewDeployment"}
...
```

For more information, including the input file format, see Create Bulk Deployments For Groups.

type: string

**ExecutionRoleArn**

The ARN of the execution role to associate with the bulk deployment operation. This IAM role must allow the `greengrass:CreateDeployment` action for all group versions that are listed in the input file. This IAM role must have access to the S3 bucket that contains the input file.

type: string
tags

The resource tags.

- type: object
- additionalProperties: The key-value pair for the resource tag. Type: string

**SubscriptionDefinitionId**

```json
{
    "SubscriptionDefinitionId": "string"
}
```

**SubscriptionDefinitionId**

The ID of the subscription definition.

- in: path
- required: true
- type: string

**SubscriptionDefinitionVersionId**

```json
{
    "SubscriptionDefinitionVersionId": "string"
}
```

**SubscriptionDefinitionVersionId**

The ID of the subscription definition version. This value maps to the Version property of the corresponding VersionInformation object, which is returned by ListSubscriptionDefinitionVersions requests. If the version is the last one that was associated with a subscription definition, the value also maps to the LatestVersion property of the corresponding DefinitionInformation object.

- in: path
- required: true
- type: string

**tags**

```json
{
    "tags": {
        "additionalProperty0": "string",
        "additionalProperty1": "string",
        "additionalProperty2": "string"
    }
}
```
tags
A map of the key-value pairs for the resource tag.
in: body
required: true
type: tags

ThingName

{
   "ThingName": "string"
}

ThingName
The thing name.
in: path
required: true
type: string

TagKeys

{  "TagKeys": [  "string"
  ]
}

TagKeys
An array of tag keys to delete.
in: query
required: true
type: array of strings

UpdateConnectivityInfoRequestBody

{
   "ConnectivityInfo": [
   {  "Id": "string",
      "HostAddress": "string",
      "PortNumber": 0x01,
      "Metadata": "string"
   }
   ]
}
UpdateDefinitionRequestBody

in: body

required: true

{ "Name": "string" }
properties

Name: The name of the definition. Required: true; Type: string

UpdateGroupCertificateConfigurationRequestBody

```json
{
  "CertificateExpiryInMilliseconds": "string"
}
```

UpdateGroupCertificateConfigurationRequestBody

in: body
required: true

schema: UpdateGroupCertificateConfigurationRequest (p. 678)

updateGroupCertificateConfigurationRequest

type: object
required: ["CertificateExpiryInMilliseconds"]

CertificateExpiryInMilliseconds

The amount of time, in milliseconds, before the certificate expires.
type: string

X-Amzn-Client-Token

```json
{
  "X-Amzn-Client-Token": "string"
}
```

X-Amzn-Client-Token

A client token used to correlate requests and responses.
in: header
type: string
required: false

Definitions

The following definitions are used:
- AssociateRoleToGroupRequest (p. 582)
- AssociateServiceRoleToAccountRequest (p. 582)
- AssociateServiceRoleToAccountResponse (p. 582)
- BulkDeployment (p. 583)
• BulkDeploymentMetrics (p. 583)
• BulkDeploymentResult (p. 584)
• BulkDeploymentResults (p. 585)
• BulkDeploymentStatus (p. 587)
• BulkDeployments (p. 587)
• ConnectivityInfo (p. 588)
• Connector (p. 588)
• ConnectorDefinitionVersion (p. 589)
• Core (p. 590)
• CoreDefinitionVersion (p. 591)
• CreateDeploymentRequest (p. 592)
• CreateDeploymentResponse (p. 592)
• CreateGroupCertificateAuthorityResponse (p. 593)
• CreateSoftwareUpdateJobRequest (p. 593)
• CreateSoftwareUpdateJobResponse (p. 594)
• DefinitionInformation (p. 594)
• Deployment (p. 596)
• DeploymentType (p. 596)
• Deployments (p. 597)
• Device (p. 597)
• DeviceDefinitionVersion (p. 598)
• DisassociateRoleFromGroupResponse (p. 599)
• DisassociateServiceRoleFromAccountResponse (p. 599)
• Empty Schema (p. 600)
• ErrorDetail (p. 600)
• ErrorDetails (p. 600)
• Function (p. 601)
• FunctionConfiguration (p. 604)
• FunctionConfigurationEnvironment (p. 606)
• FunctionDefaultConfig (p. 608)
• FunctionDefaultExecutionConfig (p. 609)
• FunctionDefinitionVersion (p. 609)
• FunctionExecutionConfig (p. 613)
• FunctionIsolationMode (p. 614)
• FunctionRunAsConfig (p. 614)
• GeneralError (p. 614)
• GetAssociatedRoleResponse (p. 615)
• GetBulkDeploymentStatusResponse (p. 616)
• GetConnectivityInfoResponse (p. 617)
• GetConnectorDefinitionVersionResponse (p. 618)
• GetCoreDefinitionVersionResponse (p. 620)
• GetDeploymentStatusResponse (p. 621)
• GetDeviceDefinitionVersionResponse (p. 623)
• GetFunctionDefinitionVersionResponse (p. 624)
• GetGroupCertificateAuthorityResponse (p. 628)
• GetGroupVersionResponse (p. 629)
• GetLoggerDefinitionVersionResponse (p. 630)
• GetResourceDefinitionVersionResponse (p. 632)
• GetServiceRoleForAccountResponse (p. 636)
• GetSubscriptionDefinitionVersionResponse (p. 637)
• Group (p. 638)
• Group Certificate Authority Properties (p. 639)
• GroupCertificateConfiguration (p. 639)
• GroupInformation (p. 640)
• GroupOwnerSetting (p. 641)
• GroupVersion (p. 641)
• IDList (p. 642)
• ListBulkDeploymentDetailedReportsResponse (p. 642)
• ListBulkDeploymentsResponse (p. 644)
• ListDefinitionsResponse (p. 645)
• ListDeploymentsRequest (p. 646)
• ListDeploymentsResponse (p. 647)
• ListGroupCertificateAuthoritiesResponse (p. 648)
• ListGroupsResponse (p. 649)
• ListVersionsResponse (p. 650)
• LocalDeviceResourceData (p. 651)
• LocalVolumeResourceData (p. 652)
• Logger (p. 652)
• LoggerComponent (p. 653)
• LoggerDefinitionVersion (p. 653)
• LoggerLevel (p. 654)
• LoggerType (p. 655)
• Permission (p. 655)
• ResetDeploymentsRequest (p. 655)
• ResetDeploymentsResponse (p. 655)
• Resource (p. 656)
• ResourceAccessPolicy (p. 660)
• ResourceDataContainer (p. 660)
• ResourceDefinitionVersion (p. 664)
• ResourceDownloadOwnerSetting (p. 668)
• S3MachineLearningModelResourceData (p. 668)
• S3UrlSignerRole (p. 669)
• SageMakerMachineLearningModelResourceData (p. 669)
• SecretsManagerSecretResourceData (p. 670)
• SoftwareToUpdate (p. 671)
• StartBulkDeploymentRequest (p. 671)
• StartBulkDeploymentResponse (p. 672)
• Subscription (p. 672)
- SubscriptionDefinitionVersion (p. 673)
- UpdateAgentLogLevel (p. 674)
- UpdateConnectivityInfoRequest (p. 674)
- UpdateConnectivityInfoResponse (p. 675)
- UpdateTargets (p. 676)
- UpdateTargetsArchitecture (p. 676)
- UpdateTargetsOperatingSystem (p. 676)
- VersionInformation (p. 676)
- Versions (p. 677)
- tags (p. 678)
- updateGroupCertificateConfigurationRequest (p. 678)

### AssociateRoleToGroupRequest

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

**AssociateRoleToGroupRequest**

- `RoleArn` 
  - type: string
  - The ARN of the role to associate with this group.

### AssociateServiceRoleToAccountRequest

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

**AssociateServiceRoleToAccountRequest**

- `RoleArn` 
  - type: string
  - The ARN of the service role to associate with your account.

### AssociateServiceRoleToAccountResponse

```json
{
}
```
"AssociatedAt": "string"
}

AssociateServiceRoleToAccountResponse
type: object

AssociatedAt
The time when the service role was associated with the account.
type: string

BulkDeployment
type: object

BulkDeploymentId
The ID of the bulk deployment.
type: string

BulkDeploymentArn
The ARN of the bulk deployment.
type: string

CreatedAt
The time, in ISO format, when the deployment was created.
type: string

BulkDeploymentMetrics
type: object

"RecordsProcessed": 0,
"InvalidInputRecords": 0,
"RetryAttempts": 0
}

BulkDeploymentMetrics
Relevant metrics on input records processed during bulk deployment.
type: object
RecordsProcessed

The total number of group records from the input file that have been processed or attempted so far.

  type: integer

InvalidInputRecords

The total number of records that returned a non-retryable error. For example, this can occur if a group record from the input file uses an invalid format or specifies a nonexistent group version, or if the execution role doesn't grant permission to deploy a group or group version.

  type: integer

RetryAttempts

The total number of deployment attempts that returned a retryable error. For example, a retry is triggered if the attempt to deploy a group returns a throttling error. StartBulkDeployment retries a group deployment up to five times.

  type: integer

BulkDeploymentResult

Information about an individual group deployment in a bulk deployment operation.

  type: object

DeploymentId

The ID of the group deployment.

  type: string

DeploymentArn

The ARN of the group deployment.

  type: string

GroupArn

The ARN of the Greengrass group.

  type: string
DeploymentType

The type of deployment. When used for CreateDeployment, only NewDeployment and Redeployment are valid.

type: string
enum: ["NewDeployment", "Redeployment", "ResetDeployment", "ForceResetDeployment"]

CreatedAt

The time, in ISO format, when the deployment was created.

type: string

DeploymentStatus

The current status of the group deployment: Building, InProgress, Success, or Failure.

type: string

ErrorMessage

The error message for a failed deployment.

type: string

ErrorDetails

A list of error details.

type: array
items: ErrorDetail (p. 600)

BulkDeploymentResults

```json
[
  {
    "DeploymentId": "string",
    "DeploymentArn": "string",
    "GroupArn": "string",
    "DeploymentType": "NewDeployment|Redeployment|ResetDeployment|ForceResetDeployment",
    "CreatedAt": "string",
    "DeploymentStatus": "string",
    "ErrorMessage": "string",
    "ErrorDetails": [
```

585
BulkDeploymentResult

Information about an individual group deployment in a bulk deployment operation.

- **DeploymentId**
  - The ID of the group deployment.
  - `type: string`

- **DeploymentArn**
  - The ARN of the group deployment.
  - `type: string`

- **GroupArn**
  - The ARN of the Greengrass group.
  - `type: string`

- **DeploymentType**
  - The type of deployment. When used for CreateDeployment, only NewDeployment and Redeployment are valid.
  - `type: string`
  - `enum: ["NewDeployment", "Redeployment", "ResetDeployment", "ForceResetDeployment"]`

- **CreatedAt**
  - The time, in ISO format, when the deployment was created.
  - `type: string`

- **DeploymentStatus**
  - The current status of the group deployment: Building, InProgress, Success, or Failure.
  - `type: string`

- **ErrorMessage**
  - The error message for a failed deployment.
  - `type: string`

- **ErrorDetails**
  - A list of error details.
BulkDeploymentStatus

```json
{
}
```

The current status of the bulk deployment.

- **type**: string
- **enum**: 
  - Initializing
  - Running
  - Completed
  - Stopping
  - Stopped
  - Failed

BulkDeployments

```json
[
{
  "BulkDeploymentId": "string",
  "BulkDeploymentArn": "string",
  "CreatedAt": "string"
}
]
```

Information about a bulk deployment. You cannot start a new bulk deployment while another one is still running or in a non-terminal state.

- **type**: object
- **BulkDeploymentId**: The ID of the bulk deployment.
  - **type**: string
BulkDeploymentArn

The ARN of the bulk deployment.

type: string

CreatedAt

The time, in ISO format, when the deployment was created.

type: string

ConnectivityInfo

Information about a Greengrass core's connectivity.

required: ["Id", "HostAddress"]

Id

The ID of the connectivity information.

type: string

HostAddress

The endpoint for the Greengrass core. Can be an IP address or DNS address.

type: string

PortNumber

The port of the Greengrass core, usually 8883.

type: integer

format: int32

Metadata

Metadata for this endpoint.

type: string

Connector

{ "Id": "string", "ConnectorArn": "string", "Parameters": { "additionalProperty0": "string", "additionalProperty1": "string", "additionalProperty2": "string" } }
Connecting to local infrastructure, device protocols, AWS, and other cloud services.

type: object

required: ["Id", "ConnectorArn"]

Id
A descriptive or arbitrary ID for the connector. This value must be unique within the connector definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9-_:]+.

type: string

ConnectorArn
The ARN of the connector.

type: string

Parameters
The parameters or configuration used by the connector.

type: object

additionalProperties: An object with properties of type string that represent the connector configuration.

ConnectorDefinitionVersion

```
{  
  "Connectors": [  
    {  
      "Id": "string",  
      "ConnectorArn": "string",  
      "Parameters": {  
        "additionalProperty0": "string",  
        "additionalProperty1": "string",  
        "additionalProperty2": "string"  
      }  
    }  
  ]
}
```

ConnectorDefinitionVersion
Information about the connector definition version, which is a container for connectors.

type: object

Connectors
A list of references to connectors in this version, with their corresponding configuration settings.

type: array

items: Connector (p. 588)
Connector

Information about a connector. Connectors run on the Greengrass core and contain built-in integration with local infrastructure, device protocols, AWS, and other cloud services.

type: object
required: ["Id", "ConnectorArn"]

Id

A descriptive or arbitrary ID for the connector. This value must be unique within the connector definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-]+.

type: string

ConnectorArn

The ARN of the connector.

type: string

Parameters

The parameters or configuration used by the connector.

type: object
additionalProperties: An object with properties of type string that represent the connector configuration.

Core

Information about a core.

type: object
required: ["Id", "ThingArn", "CertificateArn"]

Id

A descriptive or arbitrary ID for the core. This value must be unique within the core definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-]+.

type: string

ThingArn

The ARN of the thing that is the core.

type: string

CertificateArn

The ARN of the certificate associated with the core.
CoreDefinitionVersion

Information about a core definition version.

type: object

Cores

A list of cores in the core definition version.

type: array

items: Core (p. 590)

Core

Information about a core.

type: object

required: ["Id", "ThingArn", "CertificateArn"]

Id

A descriptive or arbitrary ID for the core. This value must be unique within the core definition version. Maximum length is 128 characters with the pattern \[a-zA-Z0-9:._#]+. 

type: string

ThingArn

The ARN of the thing that is the core.

type: string

CertificateArn

The ARN of the certificate associated with the core.

type: string

SyncShadow

If true, the core's local shadow is synced with the cloud automatically.
CreateDeploymentRequest

```json
{
  "DeploymentType": "NewDeployment|Redeployment|ResetDeployment|ForceResetDeployment",
  "DeploymentId": "string",
  "GroupVersionId": "string"
}
```

Information about a deployment.

**DeploymentType**

The type of deployment. When used for `CreateDeployment`, only `NewDeployment` and `Redeployment` are valid.

**DeploymentId**

The ID of the previous deployment you want to redeploy.

**GroupVersionId**

The ID of the group version to be deployed.

CreateDeploymentResponse

```json
{
  "DeploymentArn": "string",
  "DeploymentId": "string"
}
```

**DeploymentArn**

The ARN of the deployment.

**DeploymentId**

The ID of the deployment.
CreateGroupCertificateAuthorityResponse

```json
{
    "GroupCertificateAuthorityArn": "string"
}
```

CreateGroupCertificateAuthorityResponse
type: object

GroupCertificateAuthorityArn
The ARN of the group certificate authority.
type: string

CreateSoftwareUpdateJobRequest

```json
{
    "UpdateTargetsArchitecture": "armv6l|armv7l|x86_64|aarch64",
    "UpdateTargets": [
        "string"
    ],
    "SoftwareToUpdate": "core|ota_agent",
    "S3UrlSignerRole": "string",
    "UpdateAgentLogLevel": "NONE|TRACE|DEBUG|VERBOSE|INFO|WARN|ERROR|FATAL",
    "UpdateTargetsOperatingSystem": "ubuntu|raspbian|amazon_linux|openwrt"
}
```

CreateSoftwareUpdateJobRequest
Request for the CreateSoftwareUpdateJob API.
type: object


UpdateTargetsArchitecture
The architecture of the cores that are the targets of an update.
type: string
eum: ["armv6l", "armv7l", "x86_64", "aarch64"]

UpdateTargets
The ARNs of the targets (IoT things or IoT thing groups) that this update is applied to.
type: array

SoftwareToUpdate
The piece of software on the Greengrass core that will be updated.
type: string
enum: ["core", "ota_agent"]

S3UrlSignerRole

The IAM role that Greengrass uses to create presigned URLs that point to the update artifact.

type: string

UpdateAgentLogLevel

The minimum level of log statements that should be logged by the OTA agent during an update.

type: string

enum: ["NONE", "TRACE", "DEBUG", "VERBOSE", "INFO", "WARN", "ERROR", "FATAL"]

UpdateTargetsOperatingSystem

The operating system of the cores that are the targets of an update.

type: string

enum: ["ubuntu", "raspbian", "amazon_linux", "openwrt"]

**CreateSoftwareUpdateJobResponse**

```
{
    "IotJobId": "string",
    "IotJobArn": "string",
    "PlatformSoftwareVersion": "string"
}
```

CreateSoftwareUpdateJobResponse

type: object

IotJobId

The IoT job ID that corresponds to this update.

type: string

IotJobArn

The IoT job ARN that corresponds to this update.

type: string

PlatformSoftwareVersion

The software version installed on the device or devices after the update.

type: string

**DefinitionInformation**

```
{
    "Name": "string",
    "Id": "string",
    "Arn": "string",
}
```

594
"tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
},
"LastUpdatedTimestamp": "string",
"CreationTimestamp": "string",
"LatestVersion": "string",
"LatestVersionArn": "string"

DefinitionInformation

Information about a definition.

type: object

Name

The name of the definition.

type: string

Id

The ID of the definition.

type: string

Arn

The ARN of the definition.

type: string

tags

The resource tags.

type: object

  additionalProperties: The key-value pair for the resource tag. Type: string

LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the definition was last updated.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the definition was created.

type: string

LatestVersion

The ID of the latest version associated with the definition.

type: string

LatestVersionArn

The ARN of the latest version associated with the definition.

type: string
## Deployment

```
{  
  "GroupArn": "string",
  "DeploymentId": "string",
  "DeploymentArn": "string",
  "DeploymentType": "NewDeployment|Redeployment|ResetDeployment|ForceResetDeployment",
  "CreatedAt": "string"
}
```

**Deployment**

Information about a deployment.

type: object

**GroupArn**

The ARN of the group for this deployment.

type: string

**DeploymentId**

The ID of the deployment.

type: string

**DeploymentArn**

The ARN of the deployment.

type: string

**DeploymentType**

The type of deployment. When used for `createDeployment`, only `NewDeployment` and `Redeployment` are valid.

type: string

enum: ["NewDeployment", "Redeployment", "ResetDeployment", "ForceResetDeployment"]

**CreatedAt**

The time, in milliseconds since the epoch, when the deployment was created.

type: string

## DeploymentType

```
{ }
```

The type of deployment. When used for `createDeployment`, only `NewDeployment` and `Redeployment` are valid.

type: string

enum: ["NewDeployment", "Redeployment", "ResetDeployment", "ForceResetDeployment"]
Deployments

[  
  {  
    "GroupArn": "string",  
    "DeploymentId": "string",  
    "DeploymentArn": "string",  
    "DeploymentType": "NewDeployment|Redeployment|ResetDeployment|ForceResetDeployment",  
    "CreatedAt": "string"  
  }  
]

type: array
items: Deployment (p. 596)

Deployment

Information about a deployment.

type: object

GroupArn

The ARN of the group for this deployment.

type: string

DeploymentId

The ID of the deployment.

type: string

DeploymentArn

The ARN of the deployment.

type: string

DeploymentType

The type of deployment. When used for CreateDeployment, only NewDeployment and Redeployment are valid.

type: string

enum: ["NewDeployment", "Redeployment", "ResetDeployment", "ForceResetDeployment"]

CreatedAt

The time, in milliseconds since the epoch, when the deployment was created.

type: string

Device

{
  "Id": "string",
  "ThingArn": "string",
}
"CertificateArn": "string",
"SyncShadow": true
}

Device

Information about a device.

type: object

required: ["Id", "ThingArn", "CertificateArn"]

Id

A descriptive or arbitrary ID for the device. This value must be unique within the device definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-]+.

type: string

ThingArn

The thing ARN of the device.

type: string

CertificateArn

The ARN of the certificate associated with the device.

type: string

SyncShadow

If true, the device's local shadow is synced with the cloud automatically.

type: boolean

DeviceDefinitionVersion

{

"Devices": [

{

"Id": "string",
"ThingArn": "string",
"CertificateArn": "string",
"SyncShadow": true

}

]

}

DeviceDefinitionVersion

Information about a device definition version.

type: object

Devices

A list of devices in the definition version.

type: array
items: Device (p. 597)

Device

Information about a device.

type: object

required: ["Id", "ThingArn", "CertificateArn"]

Id

A descriptive or arbitrary ID for the device. This value must be unique within the device definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-]+.

type: string

ThingArn

The thing ARN of the device.

type: string

CertificateArn

The ARN of the certificate associated with the device.

type: string

SyncShadow

If true, the device's local shadow is synced with the cloud automatically.

type: boolean

DisassociateRoleFromGroupResponse

```json
{
  "DisassociatedAt": "string"
}
```

DisassociateRoleFromGroupResponse

type: object

DisassociatedAt

The time, in milliseconds since the epoch, when the role was disassociated from the group.

type: string

DisassociateServiceRoleFromAccountResponse

```json
{
  "DisassociatedAt": "string"
}
```

DisassociateServiceRoleFromAccountResponse

type: object
DisassociatedAt

The time when the service role was disassociated from the account.

type: string

Empty Schema

{}

Empty Schema

Empty
type: object

ErrorDetail

{  "DetailedErrorCode": "string",
   "DetailedErrorMessage": "string"
}

ErrorDetail

Details about the error.
type: object

DetailedErrorCode

A detailed error code.
type: string

DetailedErrorMessage

A detailed error message.
type: string

ErrorDetails

[
  {  "DetailedErrorCode": "string",
      "DetailedErrorMessage": "string"
  }
]

ErrorDetails

A list of error details.
Function

Information about a Lambda function.

type: object

required: ["Id"]
Id

A descriptive or arbitrary ID for the function. This value must be unique within the function definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-]+.

  type: string
FunctionArn

  The ARN of the alias (recommended) or version of the target Lambda function.

  type: string
FunctionConfiguration

  The configuration of the Lambda function.

  type: object
Pinned

  True if the function is pinned. Pinned means the function is long-lived and starts when the core starts.

  type: boolean
Executable

  The name of the function executable.

  type: string
ExecArgs

  The execution arguments.

  type: string
MemorySize

  The memory size, in KB, required by the function. This setting does not apply and should be cleared when you run the Lambda function without containerization.

  type: integer
Timeout

  The allowed function execution time, after which Lambda should terminate the function. This timeout still applies to pinned Lambda functions for each request.

  type: integer
EncodingType

  The expected encoding type of the input payload for the function. The default is json.

  type: string
  enum: ["binary", "json"]
Environment

  The environment configuration of the function.

  type: object
Variables

  Environment variables for the Lambda function's configuration.
type: object

additionalProperties: An object with properties of type string that represent the environment variables.

ResourceAccessPolicies

A list of the resources, with their permissions, to which the Lambda function is granted access. A Lambda function can have at most 10 resources. ResourceAccessPolicies applies only when you run the Lambda function in a Greengrass container.

type: array

items: ResourceAccessPolicy (p. 660)

ResourceAccessPolicy

A policy used by the function to access a resource.

type: object

required: ["ResourceId"]

ResourceId

The ID of the resource. (This ID is assigned to the resource when you create the resource definition.)

type: string

Permission

The type of permission a function has to access a resource.

type: string

enum: ["ro", "rw"]

AccessSysfs

If true, the Lambda function is allowed to access the host's /sys folder. Use this when the Lambda function needs to read device information from /sys. This setting applies only when you run the Lambda function in a Greengrass container.

type: boolean

Execution

Configuration information that specifies how a Lambda function runs.

type: object

IsolationMode

Specifies whether the Lambda function runs in a Greengrass container (default) or without containerization. Unless your scenario requires that you run without containerization, we recommend that you run in a Greengrass container. Omit this value to run the Lambda function with the default containerization for the group.

type: string

enum: ["GreengrassContainer", "NoContainer"]

RunAs

Specifies the user and group whose permissions are used when running the Lambda function. You can specify one or both values to override the default values. To minimize the risk of unintended changes or malicious attacks, we recommend that you avoid running as root unless absolutely
necessary. To run as root, you must update config.json in greengrass-root/config to set allowFunctionsToRunAsRoot to yes.

type: object

Uid

The user ID whose permissions are used to run a Lambda function.

type: integer

Gid

The group ID whose permissions are used to run a Lambda function.

type: integer

FunctionConfiguration

{
  "Pinned": true,
  "Executable": "string",
  "ExecArgs": "string",
  "MemorySize": 0,
  "Timeout": 0,
  "EncodingType": "binary|json",
  "Environment": {
    "Variables": {
      "additionalProperty0": "string",
      "additionalProperty1": "string",
      "additionalProperty2": "string"
    },
    "ResourceAccessPolicies": [
      {
        "ResourceId": "string",
        "Permission": "ro|rw"
      }
    ],
    "AccessSysfs": true,
    "Execution": {
      "IsolationMode": "GreengrassContainer|NoContainer",
      "RunAs": {
        "Uid": 1001,
        "Gid": 1002
      }
    }
  }
}

The configuration of the Lambda function.

type: object

Pinned

True if the function is pinned. Pinned means the function is long-lived and starts when the core starts.

type: boolean

Executable

The name of the function executable.
FunctionConfiguration

- type: string
- ExecArgs
  - The execution arguments.
  - type: string
- MemorySize
  - The memory size, in KB, required by the function. This setting does not apply and should be cleared when you run the Lambda function without containerization.
  - type: integer
- Timeout
  - The allowed function execution time, after which Lambda should terminate the function. This timeout still applies to pinned Lambda functions for each request.
  - type: integer
- EncodingType
  - The expected encoding type of the input payload for the function. The default is json.
  - type: string
  - enum: ["binary", "json"]
- Environment
  - The environment configuration of the function.
  - type: object
- Variables
  - Environment variables for the Lambda function’s configuration.
  - type: object
  - additionalProperties: An object with properties of type string that represent the environment variables.
- ResourceAccessPolicies
  - A list of the resources, with their permissions, to which the Lambda function is granted access. A Lambda function can have at most 10 resources. ResourceAccessPolicies applies only when you run the Lambda function in a Greengrass container.
  - type: array
  - items: ResourceAccessPolicy (p. 660)
- ResourceAccessPolicy
  - A policy used by the function to access a resource.
  - type: object
  - required: ["ResourceId"]
- ResourceId
  - The ID of the resource. (This ID is assigned to the resource when you create the resource definition.)
  - type: string
Permission
The type of permission a function has to access a resource.

type: string
enum: ["ro", "rw"]

AccessSysfs
If true, the Lambda function is allowed to access the host's /sys folder. Use this when the Lambda function needs to read device information from /sys. This setting applies only when you run the Lambda function in a Greengrass container.

type: boolean

Execution
Configuration information that specifies how a Lambda function runs.

type: object

IsolationMode
Specifies whether the Lambda function runs in a Greengrass container (default) or without containerization. Unless your scenario requires that you run without containerization, we recommend that you run in a Greengrass container. Omit this value to run the Lambda function with the default containerization for the group.

type: string
enum: ["GreengrassContainer", "NoContainer"]

RunAs
Specifies the user and group whose permissions are used when running the Lambda function. You can specify one or both values to override the default values. To minimize the risk of unintended changes or malicious attacks, we recommend that you avoid running as root unless absolutely necessary. To run as root, you must update config.json in greengrass-root/config to set allowFunctionsToRunAsRoot to yes.

type: object

Uid
The user ID whose permissions are used to run a Lambda function.

type: integer

Gid
The group ID whose permissions are used to run a Lambda function.

type: integer

FunctionConfigurationEnvironment

```json
{
  "Variables": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  },
  "ResourceAccessPolicies": [
```
FunctionConfigurationEnvironment

```json
{
    "ResourceId": "string",
    "Permission": "ro|rw"
}
```

```
],
"AccessSysfs": true,
"Execution": {
    "IsolationMode": "GreengrassContainer|NoContainer",
    "RunAs": {
        "Uid": 1001,
        "Gid": 1002
    }
}
```

The environment configuration of the function.

- **type:** object

**Variables**

Environment variables for the Lambda function's configuration.

- **type:** object
  - **additionalProperties:** An object with properties of type string that represent the environment variables.

**ResourceAccessPolicies**

A list of the resources, with their permissions, to which the Lambda function is granted access. A Lambda function can have at most 10 resources. ResourceAccessPolicies applies only when you run the Lambda function in a Greengrass container.

- **type:** array
  - **items:** ResourceAccessPolicy (p. 660)

**ResourceAccessPolicy**

A policy used by the function to access a resource.

- **type:** object
  - **required:** ["ResourceId"

**ResourceId**

The ID of the resource. (This ID is assigned to the resource when you create the resource definiton.)

- **type:** string

**Permission**

The type of permission a function has to access a resource.

- **type:** string
  - **enum:** ["ro", "rw"]

**AccessSysfs**

If true, the Lambda function is allowed to access the host's /sys folder. Use this when the Lambda function needs to read device information from /sys. This setting applies only when you run the Lambda function in a Greengrass container.
FunctionDefaultConfig

```json
{
  "Execution": {
    "IsolationMode": "GreengrassContainer|NoContainer",
    "RunAs": {
      "Uid": 1001,
      "Gid": 1002
    }
  }
}
```

FunctionDefaultConfig

The default configuration that applies to all Lambda functions in the group. Individual Lambda functions can override these settings.

- **Execution**

Configuration information that specifies how a Lambda function runs.
**FunctionDefaultExecutionConfig**

```json
{
    "IsolationMode": "GreengrassContainer|NoContainer",
    "RunAs": {
        "Uid": 1001,
        "Gid": 1002
    }
}
```

**FunctionExecutionConfig**

Configuration information that specifies how a Lambda function runs.

- **type**: object

**IsolationMode**

Specifies whether the Lambda function runs in a Greengrass container (default) or without containerization. Unless your scenario requires that you run without containerization, we recommend that you run in a Greengrass container. Omit this value to run the Lambda function with the default containerization for the group.

- **type**: string
- **enum**: ["GreengrassContainer", "NoContainer"]

**RunAs**

Specifies the user and group whose permissions are used when running the Lambda function. You can specify one or both values to override the default values. To minimize the risk of unintended changes or malicious attacks, we recommend that you avoid running as root unless absolutely necessary. To run as root, you must update config.json in `greengrass-root/config` to set `allowFunctionsToRunAsRoot` to `yes`.

- **type**: object

**Uid**

The user ID whose permissions are used to run a Lambda function.

- **type**: integer

**Gid**

The group ID whose permissions are used to run a Lambda function.

- **type**: integer

**FunctionDefinitionVersion**

```json
{
    "DefaultConfig": {
        "Execution": {
            "IsolationMode": "GreengrassContainer|NoContainer",
            "RunAs": {
                "Uid": 1001,
                "Gid": 1002
            }
        }
    }
}
```
"Functions": [
    {
        "Id": "string",
        "FunctionArn": "string",
        "FunctionConfiguration": {
            "Pinned": true,
            "Executable": "string",
            "ExecArgs": "string",
            "MemorySize": 0,
            "Timeout": 0,
            "EncodingType": "binary|json",
            "Environment": {
                "Variables": {
                    "additionalProperty0": "string",
                    "additionalProperty1": "string",
                    "additionalProperty2": "string"
                }
            }
        }
    }
]

FunctionDefinitionVersion

Information about a function definition version.

type: object

DefaultConfig

The default configuration that applies to all Lambda functions in the group. Individual Lambda functions can override these settings.

type: object

Execution

Configuration information that specifies how a Lambda function runs.

Functions

A list of Lambda functions in this function definition version.

type: array

items: Function (p. 601)

Information about a Lambda function.

type: object
required: ["Id"]

Id

A descriptive or arbitrary ID for the function. This value must be unique within the function
definition version. Maximum length is 128 characters with the pattern \[ a-zA-Z0-9:_-\]+. 

type: string

FunctionArn

The ARN of the alias (recommended) or version of the target Lambda function.

type: string

FunctionConfiguration

The configuration of the Lambda function.

type: object

Pinned

True if the function is pinned. Pinned means the function is long-lived and starts when the core
starts.

type: boolean

Executable

The name of the function executable.

type: string

ExecArgs

The execution arguments.

type: string

MemorySize

The memory size, in KB, required by the function. This setting does not apply and should be cleared
when you run the Lambda function without containerization.

type: integer

Timeout

The allowed function execution time, after which Lambda should terminate the function. This
timeout still applies to pinned Lambda functions for each request.

type: integer

EncodingType

The expected encoding type of the input payload for the function. The default is \texttt{json}.

type: string

enum: ["binary", "json"]

Environment

The environment configuration of the function.

type: object
Variables

Environment variables for the Lambda function's configuration.

type: object

additionalProperties: An object with properties of type string that represent the environment variables.

ResourceAccessPolicies

A list of the resources, with their permissions, to which the Lambda function is granted access. A Lambda function can have at most 10 resources. ResourceAccessPolicies applies only when you run the Lambda function in a Greengrass container.

type: array

items: ResourceAccessPolicy (p. 660)

ResourceAccessPolicy

A policy used by the function to access a resource.

type: object

required: ["ResourceId"]

ResourceId

The ID of the resource. (This ID is assigned to the resource when you create the resource definition.)

type: string

Permission

The type of permission a function has to access a resource.

type: string

enum: ["ro", "rw"]

AccessSysfs

If true, the Lambda function is allowed to access the host's /sys folder. Use this when the Lambda function needs to read device information from /sys. This setting applies only when you run the Lambda function in a Greengrass container.

type: boolean

Execution

Configuration information that specifies how a Lambda function runs.

 type: object

IsolationMode

Specifies whether the Lambda function runs in a Greengrass container (default) or without containerization. Unless your scenario requires that you run without containerization, we recommend that you run in a Greengrass container. Omit this value to run the Lambda function with the default containerization for the group.

type: string

enum: ["GreengrassContainer", "NoContainer"]
RunAs

Specifies the user and group whose permissions are used when running the Lambda function. You can specify one or both values to override the default values. To minimize the risk of unintended changes or malicious attacks, we recommend that you avoid running as root unless absolutely necessary. To run as root, you must update config.json in greengrass-root/config to set allowFunctionsToRunAsRoot to yes.

type: object

Uid

The user ID whose permissions are used to run a Lambda function.

type: integer

Gid

The group ID whose permissions are used to run a Lambda function.

type: integer

FunctionExecutionConfig

Configuration information that specifies how a Lambda function runs.

type: object

IsolationMode

Specifies whether the Lambda function runs in a Greengrass container (default) or without containerization. Unless your scenario requires that you run without containerization, we recommend that you run in a Greengrass container. Omit this value to run the Lambda function with the default containerization for the group.

type: string

enum: ["GreengrassContainer", "NoContainer"]

RunAs

Specifies the user and group whose permissions are used when running the Lambda function. You can specify one or both values to override the default values. To minimize the risk of unintended changes or malicious attacks, we recommend that you avoid running as root unless absolutely necessary. To run as root, you must update config.json in greengrass-root/config to set allowFunctionsToRunAsRoot to yes.

type: object

Uid

The user ID whose permissions are used to run a Lambda function.
FunctionIsolationMode
type: string
eenum: ["GreengrassContainer", "NoContainer"]

FunctionRunAsConfig

type: object

Uid
The user ID whose permissions are used to run a Lambda function.
type: integer

Gid
The group ID whose permissions are used to run a Lambda function.
type: integer

GeneralError
GetAssociatedRoleResponse

{  
  "AssociatedAt": "string",
  "RoleArn": "string"
}

GetAssociatedRoleResponse

  type: object

AssociatedAt

  The time when the role was associated with the group.
type: string

RoleArn

The ARN of the role that is associated with the group.

type: string

GetBulkDeploymentStatusResponse

Information about the status of a bulk deployment at the time of the request.

type: object

BulkDeploymentStatus

The current status of the bulk deployment.

type: string

enum: ['Initializing', 'Running', 'Completed', 'Stopping', 'Stopped', 'Failed']

BulkDeploymentMetrics

Relevant metrics on input records processed during bulk deployment.

type: object

RecordsProcessed

The total number of group records from the input file that have been processed or attempted so far.

type: integer

InvalidInputRecords

The total number of records that returned a non-retryable error. For example, this can occur if a group record from the input file uses an invalid format or specifies a nonexistent group version, or if the execution role doesn't grant permission to deploy a group or group version.
type: integer

RetryAttempts

The total number of deployment attempts that returned a retryable error. For example, a retry is triggered if the attempt to deploy a group returns a throttling error. StartBulkDeployment retries a group deployment up to five times.

type: integer
tag$tags

The resource tags.
type: object

additionalProperties: The key-value pair for the resource tag. Type: string

CreatedAt

The time, in ISO format, when the deployment was created.
type: string

ErrorMessage

Error message
type: string

ErrorDetails

A list of error details.
type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.
type: object

DetailedErrorCode

A detailed error code.
type: string

DetailedErrorMessage

A detailed error message.
type: string

---

GetConnectivityInfoResponse

```
{
    "message": "string",
    "ConnectivityInfo": [
        {
            "Id": "string",
            "HostAddress": "string",
            "PortNumber": 0x01,
            "Metadata": "string"
        }
    ]
}
```
Information about a Greengrass core's connectivity.

message

A message about the connectivity info request.

type: string

ConnectivityInfo

Connectivity info list.

type: array

items: ConnectivityInfo (p. 588)

Information about a Greengrass core's connectivity.

required: ["Id", "HostAddress"]

Id

The ID of the connectivity information.

type: string

HostAddress

The endpoint for the Greengrass core. Can be an IP address or DNS address.

type: string

PortNumber

The port of the Greengrass core, usually 8883.

type: integer

form: int32

Metadata

Metadata for this endpoint.

 type: string

GetConnectorDefinitionVersionResponse

```json
{
  "Arn": "string",
  "Id": "string",
  "Version": "string",
  "CreationTimestamp": "string",
  "Definition": {
    "Connectors": [
      {
        "Id": "string",
        "ConnectorArn": "string",
        "Parameters": {
```
GetConnectorDefinitionVersionResponse

Information about a connector definition version.

type: object

Arn

The ARN of the connector definition version.

type: string

Id

The ID of the connector definition version.

type: string

Version

The version of the connector definition version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the connector definition version was created.

type: string

Definition

Information about the connector definition version, which is a container for connectors.

type: object

Connectors

A list of references to connectors in this version, with their corresponding configuration settings.

type: array

items: Connector (p. 588)

Connector

Information about a connector. Connectors run on the Greengrass core and contain built-in integration with local infrastructure, device protocols, AWS, and other cloud services.

type: object

required: ["Id", "ConnectorArn"]

Id

A descriptive or arbitrary ID for the connector. This value must be unique within the connector definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9-_]+.
GetCoreDefinitionVersionResponse

- **type**: string

**ConnectorArn**

The ARN of the connector.

- **type**: string

**Parameters**

The parameters or configuration used by the connector.

- **type**: object

  - additionalProperties: An object with properties of type `string` that represent the connector configuration.

**NextToken**

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

- **type**: string

```json
GetCoreDefinitionVersionResponse
{
  "Arn": "string",
  "Id": "string",
  "Version": "string",
  "CreationTimestamp": "string",
  "Definition": {
    "Cores": [
      {
        "Id": "string",
        "ThingArn": "string",
        "CertificateArn": "string",
        "SyncShadow": true
      }
    ],
  },
  "NextToken": "string"
}
```

**GetCoreDefinitionVersionResponse**

- **type**: object

**Arn**

The ARN of the core definition version.

- **type**: string

**Id**

The ID of the core definition version.

- **type**: string

**Version**

The version of the core definition version.

- **type**: string
CreationTimestamp
  The time, in milliseconds since the epoch, when the core definition version was created.
  type: string
Definition
  Information about a core definition version.
  type: object
Cores
  A list of cores in the core definition version.
  type: array
  items: Core (p. 590)
Core
  Information about a core.
  type: object
  required: ["Id", "ThingArn", "CertificateArn"]
Id
  A descriptive or arbitrary ID for the core. This value must be unique within the core definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-]+.
  type: string
ThingArn
  The ARN of the thing that is the core.
  type: string
CertificateArn
  The ARN of the certificate associated with the core.
  type: string
SyncShadow
  If true, the core's local shadow is synced with the cloud automatically.
  type: boolean
NextToken
  The token for the next set of results, or null if there are no more results.
  type: string

GetDeploymentStatusResponse

{  "DeploymentStatus": "string",  "DeploymentType": "NewDeployment|Redeployment|ResetDeployment|ForceResetDeployment",  "Cores": [    {      "Id": "string",      "ThingArn": "string",      "CertificateArn": "string",      "SyncShadow": true    }  ],  "CreationTimestamp": 1234567890}
GetDeploymentStatusResponse

Information about the status of a deployment for a group.

type: object

DeploymentStatus

The status of the deployment: Building, InProgress, Success, or Failure.

type: string

DeploymentType

The type of deployment. When used for CreateDeployment, only NewDeployment and Redeployment are valid.

type: string

enum: ["NewDeployment", "Redeployment", "ResetDeployment", "ForceResetDeployment"]

UpdatedAt

The time, in milliseconds since the epoch, when the deployment status was updated.

type: string

ErrorMessage

Error message

type: string

ErrorDetails

A list of error details.

type: array

items: ErrorDetail (p. 600)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.
GetDeviceDefinitionVersionResponse

{  
  "Arn": "string",
  "Id": "string",
  "Version": "string",
  "CreationTimestamp": "string",
  "Definition": {  
    "Devices": [  
      {  
        "Id": "string",
        "ThingArn": "string",
        "CertificateArn": "string",
        "SyncShadow": true
      }
    ],
    "NextToken": "string"
  }
}

GetDeviceDefinitionVersionResponse

type: object

Arn

The ARN of the device definition version.

type: string

Id

The ID of the device definition version.

type: string

Version

The version of the device definition version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the device definition version was created.

type: string

Definition

Information about a device definition version.

type: object

Devices

A list of devices in the definition version.

type: array

items: Device (p. 597)
Device

Information about a device.

type: object

required: ["Id", "ThingArn", "CertificateArn"]

Id

A descriptive or arbitrary ID for the device. This value must be unique within the device definition version. Maximum length is 128 characters with the pattern `[a-zA-Z0-9-_]+`.

type: string

ThingArn

The thing ARN of the device.

type: string

CertificateArn

The ARN of the certificate associated with the device.

type: string

SyncShadow

If true, the device's local shadow is synced with the cloud automatically.


type: boolean

NextToken

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


type: string

---

**GetFunctionDefinitionVersionResponse**

```json
{
  "Arn": "string",
  "Id": "string",
  "Version": "string",
  "CreationTimestamp": "string",
  "Definition": {
    "DefaultConfig": {
      "Execution": {
        "IsolationMode": "GreengrassContainer|NoContainer",
        "RunAs": {
          "Uid": 1001,
          "Gid": 1002
        }
      }
    },
    "Functions": [
      {
        "Id": "string",
        "FunctionArn": "string",
        "FunctionConfiguration": {
          "Pinned": true,
          "Executable": "string",
          "ExecArgs": "string",
```
GetFunctionDefinitionVersionResponse

Information about a function definition version.

type: object

Arn

The ARN of the function definition version.

type: string

Id

The ID of the function definition version.

type: string

Version

The version of the function definition version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the function definition version was created.

type: string

Definition

Information about a function definition version.

type: object
DefaultConfig

The default configuration that applies to all Lambda functions in the group. Individual Lambda functions can override these settings.

type: object

Execution

Configuration information that specifies how a Lambda function runs.

Functions

A list of Lambda functions in this function definition version.

type: array

items: Function (p. 601)

Information about a Lambda function.

type: object

required: ["Id"]

Id

A descriptive or arbitrary ID for the function. This value must be unique within the function definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-]+.

type: string

FunctionArn

The ARN of the alias (recommended) or version of the target Lambda function.

type: string

FunctionConfiguration

The configuration of the Lambda function.

type: object

Pinned

True if the function is pinned. Pinned means the function is long-lived and starts when the core starts.

type: boolean

Executable

The name of the function executable.

type: string

ExecArgs

The execution arguments.

type: string

MemorySize

The memory size, in KB, required by the function. This setting does not apply and should be cleared when you run the Lambda function without containerization.
type: integer

**Timeout**

The allowed function execution time, after which Lambda should terminate the function. This timeout still applies to pinned Lambda functions for each request.

type: integer

**EncodingType**

The expected encoding type of the input payload for the function. The default is `json`.

type: string

enum: `["binary", "json"]`

**Environment**

The environment configuration of the function.

type: object

**Variables**

Environment variables for the Lambda function's configuration.

type: object

additionalProperties: An object with properties of type `string` that represent the environment variables.

**ResourceAccessPolicies**

A list of the resources, with their permissions, to which the Lambda function is granted access. A Lambda function can have at most 10 resources. ResourceAccessPolicies applies only when you run the Lambda function in a Greengrass container.

type: array

items: [ResourceAccessPolicy (p. 660)]

**ResourceAccessPolicy**

A policy used by the function to access a resource.

type: object

required: ["ResourceId"]

**ResourceId**

The ID of the resource. (This ID is assigned to the resource when you create the resource definition.)

type: string

**Permission**

The type of permission a function has to access a resource.

type: string

enum: `["ro", "rw"]`

**AccessSysfs**

If true, the Lambda function is allowed to access the host's /sys folder. Use this when the Lambda function needs to read device information from /sys. This setting applies only when you run the Lambda function in a Greengrass container.
GetGroupCertificateAuthorityResponse

Information about a certificate authority for a group.

type: object

PemEncodedCertificate

The PEM encoded certificate for the group.
GetGroupVersionResponse

Information about a group version.

type: object

Arn

The ARN of the group version.

type: string

Id

The ID of the group that the version is associated with.

type: string

Version

The ID of the group version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the group version was created.

type: string

Definition

Information about a group version.
AWS Greengrass API Reference
GetLoggerDefinitionVersionResponse

type: object
CoreDefinitionVersionArn

The ARN of the core definition version for this group.
type: string
DeviceDefinitionVersionArn

The ARN of the device definition version for this group.
type: string
FunctionDefinitionVersionArn

The ARN of the function definition version for this group.
type: string
SubscriptionDefinitionVersionArn

The ARN of the subscription definition version for this group.
type: string
LoggerDefinitionVersionArn

The ARN of the logger definition version for this group.
type: string
ResourceDefinitionVersionArn

The ARN of the resource definition version for this group.
type: string
ConnectorDefinitionVersionArn

The ARN of the connector definition version for this group.
type: string

GetLoggerDefinitionVersionResponse
{
    "Arn": "string",
    "Id": "string",
    "Version": "string",
    "CreationTimestamp": "string",
    "Definition": {
        "Loggers": [
            {
                "Id": "string",
                "Type": "FileSystem|AWSCloudWatch",
                "Component": "GreengrassSystem|Lambda",
                "Level": "DEBUG|INFO|WARN|ERROR|FATAL",
                "Space": 0
            }
        ]
    }
}
GetLoggerDefinitionVersionResponse

Information about a logger definition version.

type: object

Arn

The ARN of the logger definition version.

type: string

Id

The ID of the logger definition version.

type: string

Version

The version of the logger definition version.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the logger definition version was created.

type: string

Definition

Information about a logger definition version.

type: object

Loggers

A list of loggers.

type: array

items: Logger (p. 652)

Logger

Information about a logger

type: object

required: ["Id", "Type", "Component", "Level"]

Id

A descriptive or arbitrary ID for the logger. This value must be unique within the logger definition version. Maximum length is 128 characters with the pattern \[a-zA-Z0-9:_-]+\.

type: string

Type

type: string

enum: ["FileSystem", "AWSCloudWatch"]

Component

type: string
enum: ["GreengrassSystem", "Lambda"]

**Level**

- type: string
  - enum: ["DEBUG", "INFO", "WARN", "ERROR", "FATAL"]

**Space**

- The amount of file space, in KB, to use if the local file system is used for logging purposes.
- type: integer

---

**GetResourceDefinitionVersionResponse**

```json
{
  "Arn": "string",
  "Id": "string",
  "Version": "string",
  "CreationTimestamp": "string",
  "Definition": {
    "Resources": [
      {
        "Id": "string",
        "Name": "string",
        "ResourceDataContainer": {
          "/*-- NOTE:": "choose one of the following --*/",
          "LocalDeviceResourceData": {
            "SourcePath": "string",
            "GroupOwnerSetting": {
              "AutoAddGroupOwner": true,
              "GroupOwner": "string"
            }
          },
          "LocalVolumeResourceData": {
            "SourcePath": "string",
            "DestinationPath": "string",
            "GroupOwnerSetting": {
              "AutoAddGroupOwner": true,
              "GroupOwner": "string"
            }
          },
          "SageMakerMachineLearningModelResourceData": {
            "SageMakerJobArn": "string",
            "DestinationPath": "string",
            "OwnerSetting": {
              "GroupOwner": "string",
              "GroupPermission": "ro|rw"
            }
          },
          "S3MachineLearningModelResourceData": {
            "S3Uri": "string",
            "DestinationPath": "string",
            "OwnerSetting": {
              "GroupOwner": "string",
              "GroupPermission": "ro|rw"
            }
          },
          "SecretsManagerSecretResourceData": {
            "ARN": "string",
            "AdditionalStagingLabelsToDownload": ["string"]
          }
        }
      }
    ]
  }
}
```
GetResourceDefinitionVersionResponse

Information about a resource definition version.
type: object

Arn

The ARN of the resource definition version.
type: string

Id

The ID of the resource definition version.
type: string

Version

The version of the resource definition version.
type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the resource definition version was created.
type: string

Definition

Information about a resource definition version.
type: object

Resources

A list of resources.
type: array

items: Resource (p. 656)

Resource

Information about a resource.
type: object

required: ["Id", "Name", "ResourceDataContainer"]

Id

The resource ID, used to refer to a resource in the Lambda function configuration. Maximum length is 128 characters with the pattern [a-zA-Z0-9_:]+. This must be unique within a Greengrass group.
type: string

 Arn

The ARN of the resource definition version.

 type: string

 Id

The ID of the resource definition version.

 type: string

 Version

The version of the resource definition version.

 type: string

 CreationTimestamp

The time, in milliseconds since the epoch, when the resource definition version was created.

 type: string

 Definition

Information about a resource definition version.

 type: object

 Resources

A list of resources.

 type: array

 items: Resource (p. 656)
Name

The descriptive resource name, which is displayed on the AWS IoT Greengrass console. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-]+. This must be unique within a Greengrass group.

type: string

ResourceDataContainer

A container for resource data. The container takes only one of the following supported resource data types: LocalDeviceResourceData, LocalVolumeResourceData, SageMakerMachineLearningModelResourceData, S3MachineLearningModelResourceData, SecretsManagerSecretResourceData.

type: object

NOTE: oneOf

LocalDeviceResourceData

Attributes that define a local device resource.

type: object

SourcePath

The local absolute path of the device resource. The source path for a device resource can refer only to a character device or block device under /dev.

type: string

GroupOwnerSetting

Group owner related settings for local resources.

type: object

AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.

type: boolean

GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.

type: string

LocalVolumeResourceData

Attributes that define a local volume resource.

type: object

SourcePath

The local absolute path of the volume resource on the host. The source path for a volume resource type cannot start with /sys.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.
type: string
GroupOwnerSetting

Group owner related settings for local resources.

type: object
AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.

type: boolean
GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.

type: string
SageMakerMachineLearningModelResourceData

Attributes that define an Amazon SageMaker machine learning resource.

type: object
SageMakerJobArn

The ARN of the Amazon SageMaker training job that represents the source model.

type: string
DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string
OwnerSetting

The owner setting for the downloaded machine learning resource.

type: object
required: ["GroupOwner", "GroupPermission"]
GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string
GroupPermission

The permissions that the group owner has to the resource. Valid values are rw (read-write) or ro (read-only).

type: string
enum: ["ro", "rw"]
S3MachineLearningModelResourceData

Attributes that define an Amazon S3 machine learning resource.

type: object
S3Uri

The URI of the source model in an S3 bucket. The model package must be in tar.gz or .zip format.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.

type: object

required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are rw (read-write) or ro (read-only).

type: string

enum: ["ro", "rw"]

SecretsManagerSecretResourceData

Attributes that define a secret resource, which references a secret from AWS Secrets Manager. AWS IoT Greengrass stores a local, encrypted copy of the secret on the Greengrass core, where it can be securely accessed by connectors and Lambda functions.

type: object

ARN

The ARN of the Secrets Manager secret to make available on the core. The value of the secret's latest version (represented by the AWSCURRENT staging label) is included by default.

type: string

AdditionalStagingLabelsToDownload

Optional. The staging labels in addition to AWSCURRENT whose values you want to make available on the core.

type: array

GetServiceRoleForAccountResponse

```json
{
   "AssociatedAt": "string",
   "RoleArn": "string"
}
```
GetServiceRoleForAccountResponse

  type: object

  AssociatedAt

    The time when the service role was associated with the account.

    type: string

  RoleArn

    The ARN of the role associated with the account.

    type: string

GetSubscriptionDefinitionVersionResponse

  Information about a subscription definition version.

  type: object

  Arn

    The ARN of the subscription definition version.

    type: string

  Id

    The ID of the subscription definition version.

    type: string

  Version

    The version of the subscription definition version.

    type: string

  CreationTimestamp

    The time, in milliseconds since the epoch, when the subscription definition version was created.

    type: string

  "NextToken": "string"
Definition

Information about a subscription definition version.

type: object

Subscriptions

A list of subscriptions.

type: array

items: Subscription (p. 672)

Subscription

Information about a subscription.

type: object

required: ["Id", "Source", "Subject", "Target"]

Id

A descriptive or arbitrary ID for the subscription. This value must be unique within the subscription definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-]+.

type: string

Source

The source of the subscription. Can be a thing ARN, the ARN of a Lambda function alias (recommended) or version, a connector ARN, 'cloud' (which represents AWS IoT), or 'GGShadowService'. If you specify a Lambda function, this ARN should match the ARN used to add the function to the Greengrass group.


type: string

Subject

The MQTT topic used to route the message.


type: string

Target

Where the message is sent. Can be a thing ARN, the ARN of a Lambda function alias (recommended) or version, a connector ARN, 'cloud' (which represents AWS IoT), or 'GGShadowService'. If you specify a Lambda function, this ARN should match the ARN used to add the function to the Greengrass group.


type: string

NextToken

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


type: string

Group

{
    "AssociatedAt": "string"
}

638
Group

  type: object

AssociatedAt

  The time, in milliseconds since the epoch, when the role ARN was associated with the group.

  type: string

Group Certificate Authority Properties

{
  "GroupCertificateAuthorityId": "string",
  "GroupCertificateAuthorityArn": "string"
}

Group Certificate Authority Properties

  Information about a certificate authority for a group.

  type: object

GroupCertificateAuthorityId

  The ID of the certificate authority for the group.

  type: string

GroupCertificateAuthorityArn

  The ARN of the certificate authority for the group.

  type: string

GroupCertificateConfiguration

{
  "GroupId": "string",
  "CertificateAuthorityExpiryInMilliseconds": "string",
  "CertificateExpiryInMilliseconds": "string"
}

GroupCertificateConfiguration

  Information about a group certificate configuration.

  type: object

GroupId

  The ID of the group certificate configuration.

  type: string

CertificateAuthorityExpiryInMilliseconds

  The amount of time, in milliseconds, before the certificate authority expires.

  type: string
CertificateExpiryInMilliseconds

The amount of time, in milliseconds, before the certificate expires.

type: string

GroupInformation

```json
{
   "Name": "string",
   "Id": "string",
   "Arn": "string",
   "LastUpdatedTimestamp": "string",
   "CreationTimestamp": "string",
   "LatestVersion": "string",
   "LatestVersionArn": "string"
}
```

GroupInformation

Information about a group.

type: object

Name

The name of the group.

type: string

Id

The ID of the group.

type: string

Arn

The ARN of the group.

type: string

LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the group was last updated.

type: string

CreationTimestamp

The time, in milliseconds since the epoch, when the group was created.

type: string

LatestVersion

The ID of the latest version associated with the group.

type: string

LatestVersionArn

The ARN of the latest version associated with the group.

type: string
GroupOwnerSetting

```json
{
  "AutoAddGroupOwner": true,
  "GroupOwner": "string"
}
```

GroupOwnerSetting

Group owner related settings for local resources.

type: object

AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.

type: boolean

GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.

type: string

GroupVersion

```json
{
  "CoreDefinitionVersionArn": "string",
  "DeviceDefinitionVersionArn": "string",
  "FunctionDefinitionVersionArn": "string",
  "SubscriptionDefinitionVersionArn": "string",
  "LoggerDefinitionVersionArn": "string",
  "ResourceDefinitionVersionArn": "string",
  "ConnectorDefinitionVersionArn": "string"
}
```

GroupVersion

Information about a group version.

type: object

CoreDefinitionVersionArn

The ARN of the core definition version for this group.

type: string

DeviceDefinitionVersionArn

The ARN of the device definition version for this group.

type: string

FunctionDefinitionVersionArn

The ARN of the function definition version for this group.

type: string
IDList
type: object
Ids
A list of IDs.
type: array

ListBulkDeploymentDetailedReportsResponse

{
  "Deployments": [
    {
      "DeploymentId": "string",
      "DeploymentArn": "string",
      "GroupArn": "string",
      "DeploymentType": "NewDeployment|Redeployment|ResetDeployment|ForceResetDeployment",
      "CreatedAt": "string",
      "DeploymentStatus": "string",
      "ErrorMessage": "string",
      "ErrorDetails": [
        {
          "DetailedErrorCode": "string",
          "DetailedErrorMessage": "string"
        }
      ]
    }
  ]
}
AWS Greengrass API Reference

ListBulkDeploymentDetailedReportsResponse

   type: object

Deployments

   type: array

   items: BulkDeploymentResult (p. 584)

BulkDeploymentResult

   Information about an individual group deployment in a bulk deployment operation.

   type: object

DeploymentId

   The ID of the group deployment.

   type: string

DeploymentArn

   The ARN of the group deployment.

   type: string

GroupArn

   The ARN of the Greengrass group.

   type: string

DeploymentType

   The type of deployment. When used for CreateDeployment, only NewDeployment and Redeployment are valid.

   type: string

enum: ["NewDeployment", "Redeployment", "ResetDeployment", "ForceResetDeployment"]

CreatedAt

   The time, in ISO format, when the deployment was created.

   type: string

DeploymentStatus

   The current status of the group deployment: Building, InProgress, Success, or Failure.

   type: string

ErrorMessage

   The error message for a failed deployment.

   type: string

ErrorDetails

   A list of error details.
type: array
items: ErrorDetail (p. 600)

```
ErrorDetail
  Details about the error.
  type: object
DetailedErrorCode
    A detailed error code.
    type: string
DetailedErrorMessage
    A detailed error message.
    type: string
NextToken
    The token for the next set of results, or null if there are no more results.
in: query
type: string
```

ListBulkDeploymentsResponse

```
{
  "BulkDeployments": [
    {
      "BulkDeploymentId": "string",
      "BulkDeploymentArn": "string",
      "CreatedAt": "string"
    }
  ],
  "NextToken": "string"
}
```

ListBulkDeploymentsResponse

type: object

BulkDeployments

type: array
items: BulkDeployment (p. 583)

BulkDeployment

Information about a bulk deployment. You cannot start a new bulk deployment while another one is still running or in a non-terminal state.

type: object

BulkDeploymentId

The ID of the bulk deployment.
ListDefinitionsResponse

A list of definitions.

type: object

Definitions

Information about a definition.

type: array

items: DefinitionInformation (p. 594)

DefinitionInformation

Information about a definition.

type: object
Name
   The name of the definition.
   type: string
Id
   The ID of the definition.
   type: string
Arn
   The ARN of the definition.
   type: string
tags
   The resource tags.
   type: object
   additionalProperties: The key-value pair for the resource tag. Type: string
LastUpdatedTimestamp
   The time, in milliseconds since the epoch, when the definition was last updated.
   type: string
CreationTimestamp
   The time, in milliseconds since the epoch, when the definition was created.
   type: string
LatestVersion
   The ID of the latest version associated with the definition.
   type: string
LatestVersionArn
   The ARN of the latest version associated with the definition.
   type: string
NextToken
   The token for the next set of results, or null if there are no more results.
   type: string

ListDeploymentsRequest

```
{
   "MaxResults": 0,
   "NextToken": "string"
}
```

ListDeploymentsRequest
   type: object
MaxResults
   The maximum number of results to be returned per request.
   in: query
   type: integer
   min: 1
   max: 250

NextToken
   The token to retrieve the next set of results.
   in: query
   type: string

ListDeploymentsResponse

{  
   "Deployments": [  
      {  
         "GroupArn": "string",  
         "DeploymentId": "string",  
         "DeploymentArn": "string",  
         "DeploymentType": "NewDeployment|Redeployment|ResetDeployment|ForceResetDeployment",  
         "CreatedAt": "string"  
      },  
   ],  
   "NextToken": "string"  
}

ListDeploymentsResponse
   type: object
Deployments
   type: array
      items: Deployment (p. 596)
Deployment
   Information about a deployment.
   type: object
GroupArn
   The ARN of the group for this deployment.
   type: string
DeploymentId
   The ID of the deployment.
   type: string
DeploymentArn
The ARN of the deployment.
type: string

DeploymentType
The type of deployment. When used for CreateDeployment, only NewDeployment and Redeployment are valid.
type: string
default: ["NewDeployment", "Redeployment", "ResetDeployment", "ForceResetDeployment"]

CreatedAt
The time, in milliseconds since the epoch, when the deployment was created.
type: string

NextToken
The token for the next set of results, or null if there are no more results.
in: query
type: string

**ListGroupCertificateAuthoritiesResponse**

```
{
  "GroupCertificateAuthorities": [
    {
      "GroupCertificateAuthorityId": "string",
      "GroupCertificateAuthorityArn": "string"
    }
  ]
}
```

**ListGroupCertificateAuthoritiesResponse**

type: object

**GroupCertificateAuthorities**

A list of certificate authorities associated with the group.
type: array

items: [GroupCertificateAuthorityProperties (p. 639)]

**Group Certificate Authority Properties**

Information about a certificate authority for a group.
type: object

**GroupCertificateAuthorityId**

The ID of the certificate authority for the group.
type: string
GroupCertificateAuthorityArn

The ARN of the certificate authority for the group.

type: string

ListGroupsResponse

```json
{
    "Groups": [
        {
            "Name": "string",
            "Id": "string",
            "Arn": "string",
            "LastUpdatedTimestamp": "string",
            "CreationTimestamp": "string",
            "LatestVersion": "string",
            "LatestVersionArn": "string"
        }
    ],
    "NextToken": "string"
}
```

ListGroupsResponse

type: object

Groups

Information about a group.

type: array

items: GroupInformation (p. 640)

GroupInformation

Information about a group.

type: object

Name

The name of the group.

type: string

Id

The ID of the group.

type: string

Arn

The ARN of the group.

type: string

LastUpdatedTimestamp

The time, in milliseconds since the epoch, when the group was last updated.

type: string
ListVersionsResponse

```
{
  "Versions": [
    {
      "Arn": "string",
      "Id": "string",
      "Version": "string",
      "CreationTimestamp": "string"
    }
  ],
  "NextToken": "string"
}
```

ListVersionsResponse

A list of versions.

type: object

Versions

Information about a version.

type: array

items: VersionInformation (p. 676)

VersionInformation

Information about a version.

type: object

Arn

The ARN of the version.

type: string
**Id**

The ID of the parent definition that the version is associated with.

type: string

**Version**

The ID of the version.

type: string

**CreationTimestamp**

The time, in milliseconds since the epoch, when the version was created.

type: string

**NextToken**

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

type: string

---

**LocalDeviceResourceData**

```
{
  "SourcePath": "string",
  "GroupOwnerSetting": {
    "AutoAddGroupOwner": true,
    "GroupOwner": "string"
  }
}
```

**LocalDeviceResourceData**

Attributes that define a local device resource.

type: object

**SourcePath**

The local absolute path of the device resource. The source path for a device resource can refer only to a character device or block device under /dev.

type: string

**GroupOwnerSetting**

Group owner related settings for local resources.

type: object

**AutoAddGroupOwner**

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.

type: boolean

**GroupOwner**

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.
LocalVolumeResourceData

Attributes that define a local volume resource.

type: object

SourcePath

The local absolute path of the volume resource on the host. The source path for a volume resource type cannot start with /sys.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

GroupOwnerSetting

Group owner related settings for local resources.

type: object

AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.

type: boolean

GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.

type: string

Logger

{  "Id": "string",  "Type": "FileSystem|AWSCloudWatch",  "Component": "GreengrassSystem|Lambda",  "Level": "DEBUG|INFO|WARN|ERROR|FATAL"}
Logger

Information about a logger
type: object
required: ["Id", "Type", "Component", "Level"]

Id
A descriptive or arbitrary ID for the logger. This value must be unique within the logger definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-]+.
type: string

Type
type: string
enum: ["FileSystem", "AWSCloudWatch"]

Component
type: string
enum: ["GreengrassSystem", "Lambda"]

Level
type: string
enum: ["DEBUG", "INFO", "WARN", "ERROR", "FATAL"]

Space
The amount of file space, in KB, to use if the local file system is used for logging purposes.
type: integer

LoggerComponent

{
}

type: string
enum: ["GreengrassSystem", "Lambda"]

LoggerDefinitionVersion

{
  "Loggers": [
    {
      "Id": "string",
      "Type": "FileSystem|AWSCloudWatch",
      "Space": 0
    }
  ]
}
LoggerDefinitionVersion

Information about a logger definition version.

- type: object

Loggers

A list of loggers.

- type: array

  - items: Logger (p. 652)

Logger

Information about a logger

- type: object

  - required: ["Id", "Type", "Component", "Level"]

Id

A descriptive or arbitrary ID for the logger. This value must be unique within the logger definition version. Maximum length is 128 characters with the pattern `[a-zA-Z0-9_:.-]+`.

- type: string

Type

- type: string

  - enum: ["FileSystem", "AWSCloudWatch"]

Component

- type: string

  - enum: ["GreengrassSystem", "Lambda"]

Level

- type: string

  - enum: ["DEBUG", "INFO", "WARN", "ERROR", "FATAL"]

Space

The amount of file space, in KB, to use if the local file system is used for logging purposes.

- type: integer

(LoggerLevel)


LoggerType

```json
{
  type: string
  enum: ['DEBUG', 'INFO', 'WARN', 'ERROR', 'FATAL']
}
```

Permission

```json
{
  type: string
  enum: ['FileSystem', 'AWSCloudWatch']
}
```

The type of permission a function has to access a resource.

```json
{
  type: string
  enum: ['ro', 'rw']
}
```

ResetDeploymentsRequest

```json
{
  "Force": true
}
```

ResetDeploymentsRequest

- Information about a group reset request.
- type: object
- Force
  - If true, performs a best-effort only core reset.
  - type: boolean

ResetDeploymentsResponse

```json
{
  "DeploymentArn": "string",
}
```
"DeploymentId": "string"
}

ResetDeploymentsResponse

type: object

DeploymentArn

The ARN of the deployment.

type: string

DeploymentId

The ID of the deployment.

type: string

Resource

{
"Id": "string",
"Name": "string",
"ResourceDataContainer": {
"*-- NOTE:": "choose one of the following --*",
"LocalDeviceResourceData": {
"SourcePath": "string",
"GroupOwnerSetting": {
"AutoAddGroupOwner": true,
"GroupOwner": "string"
}
},
"LocalVolumeResourceData": {
"SourcePath": "string",
"DestinationPath": "string",
"GroupOwnerSetting": {
"AutoAddGroupOwner": true,
"GroupOwner": "string"
}
},
"SageMakerMachineLearningModelResourceData": {
"SageMakerJobArn": "string",
"DestinationPath": "string",
"OwnerSetting": {
"GroupOwner": "string",
"GroupPermission": "ro|rw"
}
},
"S3MachineLearningModelResourceData": {
"S3Uri": "string",
"DestinationPath": "string",
"OwnerSetting": {
"GroupOwner": "string",
"GroupPermission": "ro|rw"
}
},
"SecretsManagerSecretResourceData": {
"ARN": "string",
"AdditionalStagingLabelsToDownload": [ "string"
] } }
Resource

Information about a resource.

type: object

required: ["Id", "Name", "ResourceDataContainer"]

Id

The resource ID, used to refer to a resource in the Lambda function configuration. Maximum length is 128 characters with the pattern [a-zA-Z0-9_:][-+]. This must be unique within a Greengrass group.

type: string

Name

The descriptive resource name, which is displayed on the AWS IoT Greengrass console. Maximum length is 128 characters with the pattern [a-zA-Z0-9_:][-+]. This must be unique within a Greengrass group.

type: string

ResourceDataContainer

A container for resource data. The container takes only one of the following supported resource data types: LocalDeviceResourceData, LocalVolumeResourceData, SageMakerMachineLearningModelResourceData, S3MachineLearningModelResourceData, SecretsManagerSecretResourceData.

type: object

NOTE: oneOf

LocalDeviceResourceData

Attributes that define a local device resource.

type: object

SourcePath

The local absolute path of the device resource. The source path for a device resource can refer only to a character device or block device under /dev.

type: string

GroupOwnerSetting

Group owner related settings for local resources.

type: object

AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.

type: boolean
GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.

type: string

LocalVolumeResourceData

Attributes that define a local volume resource.

type: object

SourcePath

The local absolute path of the volume resource on the host. The source path for a volume resource type cannot start with `/sys`.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

GroupOwnerSetting

Group owner related settings for local resources.

type: object

AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.

type: boolean

GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.

type: string

SageMakerMachineLearningModelResourceData

Attributes that define an Amazon SageMaker machine learning resource.

type: object

SageMakerJobArn

The ARN of the Amazon SageMaker training job that represents the source model.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.
type: object

required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are `rw` (read-write) or `ro` (read-only).

type: string

enum: ["ro", "rw"]

S3MachineLearningModelResourceData

Attributes that define an Amazon S3 machine learning resource.

type: object

S3Uri

The URI of the source model in an S3 bucket. The model package must be in tar.gz or .zip format.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.

type: object

required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are `rw` (read-write) or `ro` (read-only).

type: string

enum: ["ro", "rw"]

SecretsManagerSecretResourceData

Attributes that define a secret resource, which references a secret from AWS Secrets Manager. AWS IoT Greengrass stores a local, encrypted copy of the secret on the Greengrass core, where it can be securely accessed by connectors and Lambda functions.
ResourceAccessPolicy

```
{  
  "ResourceId": "string",  
  "Permission": "ro|rw"  
}
```

A policy used by the function to access a resource.

type: object

- **ResourceId**
  The ID of the resource. (This ID is assigned to the resource when you create the resource definition.)
  type: string
- **Permission**
  The type of permission a function has to access a resource.
  type: string
  enum: ["ro", "rw"]

ResourceDataContainer

```
"*-- NOTE:" : "choose one of the following --*
"LocalDeviceResourceData": {  
  "SourcePath": "string",  
  "GroupOwnerSetting": {  
    "AutoAddGroupOwner": true,  
    "GroupOwner": "string"  
  }  
},  
"LocalVolumeResourceData": {  
  "SourcePath": "string",  
  "DestinationPath": "string"  
}
```

- **SourcePath**
- **DestinationPath**
- **GroupOwnerSetting**
  - **AutoAddGroupOwner**
  - **GroupOwner**

ARN

The ARN of the Secrets Manager secret to make available on the core. The value of the secret's latest version (represented by the `AWSCURRENT` staging label) is included by default.

type: string

AdditionalStagingLabelsToDownload

Optional. The staging labels in addition to `AWSCURRENT` whose values you want to make available on the core.

type: array
A container for resource data. The container takes only one of the
following supported resource data types: LocalDeviceResourceData,
LocalVolumeResourceData, SageMakerMachineLearningModelResourceData,

type: object

NOTE: oneOf

LocalDeviceResourceData

Attributes that define a local device resource.

type: object

SourcePath

The local absolute path of the device resource. The source path for a device resource can refer only
to a character device or block device under /dev.

type: string

GroupOwnerSetting

Group owner related settings for local resources.

type: object

AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda
process privileges. The Lambda process then has the file access permissions of the added Linux
group.

type: boolean
GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.

type: string

LocalVolumeResourceData

Attributes that define a local volume resource.

type: object

SourcePath

The local absolute path of the volume resource on the host. The source path for a volume resource type cannot start with /sys.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

GroupOwnerSetting

Group owner related settings for local resources.

type: object

AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.

 type: boolean

GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.

type: string

SageMakerMachineLearningModelResourceData

Attributes that define an Amazon SageMaker machine learning resource.

type: object

SageMakerJobArn

The ARN of the Amazon SageMaker training job that represents the source model.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.
ResourceDataContainer

type: object

required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are \texttt{rw} (read-write) or \texttt{ro} (read-only).

type: string

enum: ["ro", "rw"]

S3MachineLearningModelResourceData

Attributes that define an Amazon S3 machine learning resource.

type: object

S3Uri

The URI of the source model in an S3 bucket. The model package must be in tar.gz or .zip format.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.

type: object

required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are \texttt{rw} (read-write) or \texttt{ro} (read-only).

type: string

enum: ["ro", "rw"]

SecretsManagerSecretResourceData

Attributes that define a secret resource, which references a secret from AWS Secrets Manager. AWS IoT Greengrass stores a local, encrypted copy of the secret on the Greengrass core, where it can be securely accessed by connectors and Lambda functions.

type: object
ARN

The ARN of the Secrets Manager secret to make available on the core. The value of the secret's latest version (represented by the AWSCURRENT staging label) is included by default.

type: string

AdditionalStagingLabelsToDownload

Optional. The staging labels in addition to AWSCURRENT whose values you want to make available on the core.

type: array

ResourceDefinitionVersion

```json
{
    "Id": "string",
    "Name": "string",
    "ResourceDataContainer": {
        "*-- NOTE:*": "choose one of the following --*",
        "LocalDeviceResourceData": {
            "SourcePath": "string",
            "GroupOwnerSetting": {
                "AutoAddGroupOwner": true,
                "GroupOwner": "string"
            }
        },
        "LocalVolumeResourceData": {
            "SourcePath": "string",
            "DestinationPath": "string",
            "GroupOwnerSetting": {
                "AutoAddGroupOwner": true,
                "GroupOwner": "string"
            }
        },
        "SageMakerMachineLearningModelResourceData": {
            "SageMakerJobArn": "string",
            "DestinationPath": "string",
            "OwnerSetting": {
                "GroupOwner": "string",
                "GroupPermission": "ro|rw"
            }
        },
        "S3MachineLearningModelResourceData": {
            "S3Uri": "string",
            "DestinationPath": "string",
            "OwnerSetting": {
                "GroupOwner": "string",
                "GroupPermission": "ro|rw"
            }
        },
        "SecretsManagerSecretResourceData": {
            "ARN": "string",
            "AdditionalStagingLabelsToDownload": ["string"
        ]
    }
}
```
ResourceDefinitionVersion

Information about a resource definition version.

type: object

Resources

A list of resources.

type: array

items: Resource (p. 656)

Resource

Information about a resource.

type: object

required: ["Id", "Name", "ResourceDataContainer"]

Id

The resource ID, used to refer to a resource in the Lambda function configuration. Maximum length is 128 characters with the pattern \[a-zA-Z0-9_:.-]+\+. This must be unique within a Greengrass group.

type: string

Name

The descriptive resource name, which is displayed on the AWS IoT Greengrass console. Maximum length is 128 characters with the pattern \[a-zA-Z0-9_:.-]+\+. This must be unique within a Greengrass group.

type: string

ResourceDataContainer

A container for resource data. The container takes only one of the following supported resource data types: LocalDeviceResourceData, LocalVolumeResourceData, SageMakerMachineLearningModelResourceData, S3MachineLearningModelResourceData, SecretsManagerSecretResourceData.

type: object

NOTE: oneOf

LocalDeviceResourceData

Attributes that define a local device resource.

type: object

SourcePath

The local absolute path of the device resource. The source path for a device resource can refer only to a character device or block device under /dev.

type: string

GroupOwnerSetting

Group owner related settings for local resources.
type: object
AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.

type: boolean
GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.

type: string
LocalVolumeResourceData

Attributes that define a local volume resource.

type: object
SourcePath

The local absolute path of the volume resource on the host. The source path for a volume resource type cannot start with /sys.

type: string
DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string
GroupOwnerSetting

Group owner related settings for local resources.

type: object
AutoAddGroupOwner

If true, AWS IoT Greengrass adds the specified Linux OS group owner of the resource to the Lambda process privileges. The Lambda process then has the file access permissions of the added Linux group.

type: boolean
GroupOwner

The name of the Linux OS group whose privileges are added to the Lambda process. This field is optional.

type: string
SageMakerMachineLearningModelResourceData

Attributes that define an Amazon SageMaker machine learning resource.

type: object
SageMakerJobArn

The ARN of the Amazon SageMaker training job that represents the source model.

type: string
DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.

type: object

required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are \texttt{rw} (read-write) or \texttt{ro} (read-only).

type: string

enum: ["ro", "rw"]

S3MachineLearningModelResourceData

Attributes that define an Amazon S3 machine learning resource.

type: object

S3Uri

The URI of the source model in an S3 bucket. The model package must be in tar.gz or .zip format.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.

type: object

required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are \texttt{rw} (read-write) or \texttt{ro} (read-only).

type: string
enum: ["ro", "rw"]

SecretsManagerSecretResourceData

Attributes that define a secret resource, which references a secret from AWS Secrets Manager. AWS IoT Greengrass stores a local, encrypted copy of the secret on the Greengrass core, where it can be securely accessed by connectors and Lambda functions.

type: object

ARN

The ARN of the Secrets Manager secret to make available on the core. The value of the secret's latest version (represented by the AWSCURRENT staging label) is included by default.

type: string

AdditionalStagingLabelsToDownload

Optional. The staging labels in addition to AWSCURRENT whose values you want to make available on the core.

type: array

ResourceDownloadOwnerSetting

```
{
  "GroupOwner": "string",
  "GroupPermission": "ro|rw"
}
```

ResourceDownloadOwnerSetting

The owner setting for the downloaded machine learning resource.

type: object

required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are rw (read-write) or ro (read-only).

type: string

enum: ["ro", "rw"]

S3MachineLearningModelResourceData

```
{
  "S3Uri": "string",
  "DestinationPath": "string",
  "OwnerSetting": {
```


S3MachineLearningModelResourceData

Attributes that define an Amazon S3 machine learning resource.

type: object

S3Uri

The URI of the source model in an S3 bucket. The model package must be in tar.gz or .zip format.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.

type: object

required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are rw (read-write) or ro (read-only).

type: string

enum: ["ro", "rw"]

S3UrlSignerRole

The IAM role that Greengrass uses to create presigned URLs that point to the update artifact.

type: string

SageMakerMachineLearningModelResourceData

{ "SageMakerJobArn": "string",}
SageMakerMachineLearningModelResourceData

Attributes that define an Amazon SageMaker machine learning resource.

type: object

SageMakerJobArn

The ARN of the Amazon SageMaker training job that represents the source model.

type: string

DestinationPath

The absolute local path of the resource inside the Lambda environment.

type: string

OwnerSetting

The owner setting for the downloaded machine learning resource.

type: object

required: ["GroupOwner", "GroupPermission"]

GroupOwner

The group owner of the resource. This is the group ID (GID) of an existing Linux OS group on the system. The group's permissions are added to the Lambda process.

type: string

GroupPermission

The permissions that the group owner has to the resource. Valid values are rw (read-write) or ro (read-only).

type: string

enum: ["ro", "rw"]

SecretsManagerSecretResourceData

Attributes that define a secret resource, which references a secret from AWS Secrets Manager. AWS IoT Greengrass stores a local, encrypted copy of the secret on the Greengrass core, where it can be securely accessed by connectors and Lambda functions.
type: object

**ARN**

The ARN of the Secrets Manager secret to make available on the core. The value of the secret's latest version (represented by the `AWSCURRENT` staging label) is included by default.

type: string

**AdditionalStagingLabelsToDownload**

Optional. The staging labels in addition to `AWSCURRENT` whose values you want to make available on the core.

type: array

---

**SoftwareToUpdate**

The piece of software on the Greengrass core that will be updated.

type: string

enum: ['core', 'ota_agent']

---

**StartBulkDeploymentRequest**

Information about a bulk deployment. You cannot start a new bulk deployment while another one is still running or in a non-terminal state.

type: object

required: ['InputFileUri', 'ExecutionRoleArn']

**InputFileUri**

The URI of the input file in the S3 bucket. The execution role must have `getObject` permissions on this bucket to access the input file. The input file is a JSON-serialized, line-delimited file with UTF-8 encoding that provides a list of group and version IDs and the deployment type. This file must be less than 100 MB. Currently, AWS IoT Greengrass supports only `NewDeployment` deployment types.

```json
{"GroupId":"groupId1", "GroupVersionId":"groupVersionId1", "DeploymentType":"NewDeployment"}
```
StartBulkDeploymentResponse

```json
{
    "BulkDeploymentId": "string",
    "BulkDeploymentArn": "string"
}
```

**StartBulkDeploymentResponse**

- **BulkDeploymentId**
  - The ID of the bulk deployment.
  - type: string
- **BulkDeploymentArn**
  - The ARN of the bulk deployment.
  - type: string

For more information, including the input file format, see Create Bulk Deployments For Groups.

**ExecutionRoleArn**

The ARN of the execution role to associate with the bulk deployment operation. This IAM role must allow the `greengrass:CreateDeployment` action for all group versions that are listed in the input file. This IAM role must have access to the S3 bucket that contains the input file.

- type: string

**tags**

The resource tags.

- type: object
  - additionalProperties: The key-value pair for the resource tag. Type: string

**Subscription**

```json
{
    "Id": "string",
    "Source": "string",
    "Subject": "string",
    "Target": "string"
}
```

**Subscription**

- **Id**
  - The ID of the subscription.
  - type: string
- **Source**
  - The source of the subscription.
  - type: string
- **Subject**
  - The subject of the subscription.
  - type: string
- **Target**
  - The target of the subscription.
  - type: string
Subscription

Information about a subscription.

type: object

required: ["Id", "Source", "Subject", "Target"]

Id

A descriptive or arbitrary ID for the subscription. This value must be unique within the subscription definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-\-\+].

type: string

Source

The source of the subscription. Can be a thing ARN, the ARN of a Lambda function alias (recommended) or version, a connector ARN, 'cloud' (which represents AWS IoT), or 'GGShadowService'. If you specify a Lambda function, this ARN should match the ARN used to add the function to the Greengrass group.

type: string

Subject

The MQTT topic used to route the message.

type: string

Target

Where the message is sent. Can be a thing ARN, the ARN of a Lambda function alias (recommended) or version, a connector ARN, 'cloud' (which represents AWS IoT), or 'GGShadowService'. If you specify a Lambda function, this ARN should match the ARN used to add the function to the Greengrass group.

type: string

SubscriptionDefinitionVersion

{
  "Subscriptions": [
    {
      "Id": "string",
      "Source": "string",
      "Subject": "string",
      "Target": "string"
    }
  ]
}

SubscriptionDefinitionVersion

Information about a subscription definition version.

type: object

Subscriptions

A list of subscriptions.

type: array
items: Subscription (p. 672)

Subscription

Information about a subscription.

type: object

required: ["Id", "Source", "Subject", "Target"]

Id

A descriptive or arbitrary ID for the subscription. This value must be unique within the subscription
definition version. Maximum length is 128 characters with the pattern [a-zA-Z0-9:_-\]+.

type: string

Source

The source of the subscription. Can be a thing ARN, the ARN of a Lambda function alias
(recommended) or version, a connector ARN, ‘cloud’ (which represents AWS IoT), or
‘GGShadowService’. If you specify a Lambda function, this ARN should match the ARN used to add
the function to the Greengrass group.

type: string

Subject

The MQTT topic used to route the message.

type: string

Target

Where the message is sent. Can be a thing ARN, the ARN of a Lambda function alias (recommended)
or version, a connector ARN, ‘cloud’ (which represents AWS IoT), or ‘GGShadowService’. If you specify
a Lambda function, this ARN should match the ARN used to add the function to the Greengrass
group.

type: string

UpdateAgentLogLevel

{

}

The minimum level of log statements that should be logged by the OTA agent during an update.

type: string

enum: ["NONE", "TRACE", "DEBUG", "VERBOSE", "INFO", "WARN", "ERROR", "FATAL"]

UpdateConnectivityInfoRequest

{
  "ConnectivityInfo": [
    {
      "Id": "string",
      "HostAddress": "string",
      "PortNumber": 0x01,
    }
  ]
}
UpdateConnectivityInfoResponse

```json

{  
  "Metadata": "string"
}
```

Information required to update a Greengrass core's connectivity.

**ConnectivityInfo**

- A list of connectivity info.
  - type: array
  - items: [ConnectivityInfo](p. 588)

**Information about a Greengrass core's connectivity.**

Required: ["Id", "HostAddress"]

**Id**

- The ID of the connectivity information.
  - type: string

**HostAddress**

- The endpoint for the Greengrass core. Can be an IP address or DNS address.
  - type: string

**PortNumber**

- The port of the Greengrass core, usually 8883.
  - type: integer
  - format: int32

**Metadata**

- Metadata for this endpoint.
  - type: string

**UpdateConnectivityInfoResponse**

```json

{  
  "message": "string",
  "Version": "string"
}
```

**message**

- A message about the connectivity info update request.
  - type: string

**Version**

- The new version of the connectivity info.
  - type: string
UpdateTargets

```
{
}
```

The ARNs of the targets (IoT things or IoT thing groups) that this update is applied to.

- type: array

UpdateTargetsArchitecture

```
{
}
```

The architecture of the cores that are the targets of an update.

- type: string
- enum: ["armv6l", "armv7l", "x86_64", "aarch64"]

UpdateTargetsOperatingSystem

```
{
}
```

The operating system of the cores that are the targets of an update.

- type: string
- enum: ["ubuntu", "raspbian", "amazon_linux", "openwrt"]

VersionInformation

```
{
"Arn": "string",
"Id": "string",
"Version": "string",
"CreationTimestamp": "string"
}
```

VersionInformation

- Information about a version.

- type: object

Arn

- The ARN of the version.
Versions

- **type**: string
- **Id**: The ID of the parent definition that the version is associated with.
  - **type**: string
- **Version**: The ID of the version.
  - **type**: string
- **CreationTimestamp**: The time, in milliseconds since the epoch, when the version was created.
  - **type**: string

**Versions**

```
{
  "Versions": [
    {
      "Arn": "string",
      "Id": "string",
      "Version": "string",
      "CreationTimestamp": "string"
    }
  ]
}
```

**Versions**

- **type**: object
- **Versions**: A list of versions.
  - **type**: array
    - **items**: [VersionInformation](p. 676)

**VersionInformation**

- **type**: object
- **Arn**: The ARN of the version.
  - **type**: string
- **Id**: The ID of the parent definition that the version is associated with.
  - **type**: string
- **Version**: The ID of the version.
tags

```json
{
  "additionalProperty0": "string",
  "additionalProperty1": "string",
  "additionalProperty2": "string"
}
```

The resource tags.

- type: object
- additionalProperties: The key-value pair for the resource tag. Type: string

updateGroupCertificateConfigurationRequest

```json
{
  "CertificateExpiryInMilliseconds": "string"
}
```

- type: object
- required: ["CertificateExpiryInMilliseconds"]

CertificateExpiryInMilliseconds

The amount of time, in milliseconds, before the certificate expires.

- type: string