



API Reference, Version 1

AWS IoT Greengrass



AWS IoT Greengrass: API Reference, Version 1

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

.....	xii
Greengrass V1 API	1
Actions	1
AssociateRoleToGroup	4
AssociateServiceRoleToAccount	9
CreateConnectorDefinition	13
CreateConnectorDefinitionVersion	18
CreateCoreDefinition	23
CreateCoreDefinitionVersion	29
CreateDeployment	34
CreateDeviceDefinition	38
CreateDeviceDefinitionVersion	43
CreateFunctionDefinition	48
CreateFunctionDefinitionVersion	59
CreateGroup	68
CreateGroupCertificateAuthority	74
CreateGroupVersion	78
CreateLoggerDefinition	82
CreateLoggerDefinitionVersion	88
CreateResourceDefinition	93
CreateResourceDefinitionVersion	104
CreateSoftwareUpdateJob	115
CreateSubscriptionDefinition	121
CreateSubscriptionDefinitionVersion	127
DeleteConnectorDefinition	131
DeleteCoreDefinition	134
DeleteDeviceDefinition	136
DeleteFunctionDefinition	138
DeleteGroup	141
DeleteLoggerDefinition	143
DeleteResourceDefinition	146
DeleteSubscriptionDefinition	148
DisassociateRoleFromGroup	150
DisassociateServiceRoleFromAccount	154

GetAssociatedRole	156
GetBulkDeploymentStatus	160
GetConnectivityInfo	164
GetConnectorDefinition	169
GetConnectorDefinitionVersion	172
GetCoreDefinition	177
GetCoreDefinitionVersion	181
GetDeploymentStatus	186
GetDeviceDefinition	190
GetDeviceDefinitionVersion	193
GetFunctionDefinition	198
GetFunctionDefinitionVersion	202
GetGroup	211
GetGroupCertificateAuthority	214
GetGroupCertificateConfiguration	218
GetGroupVersion	223
GetLoggerDefinition	227
GetLoggerDefinitionVersion	231
GetResourceDefinition	236
GetResourceDefinitionVersion	239
GetServiceRoleForAccount	249
GetSubscriptionDefinition	251
GetSubscriptionDefinitionVersion	254
GetThingRuntimeConfiguration	259
ListBulkDeploymentDetailedReports	264
ListBulkDeployments	269
ListConnectorDefinitionVersions	272
ListConnectorDefinitions	276
ListCoreDefinitionVersions	279
ListCoreDefinitions	283
ListDeployments	286
ListDeviceDefinitionVersions	290
ListDeviceDefinitions	294
ListFunctionDefinitionVersions	297
ListFunctionDefinitions	301
ListGroupCertificateAuthorities	304

ListGroupVersions	308
ListGroups	312
ListLoggerDefinitionVersions	315
ListLoggerDefinitions	319
ListResourceDefinitionVersions	322
ListResourceDefinitions	326
ListSubscriptionDefinitionVersions	329
ListSubscriptionDefinitions	333
ListTagsForResource	336
ResetDeployments	339
StartBulkDeployment	342
StopBulkDeployment	347
TagResource	349
UntagResource	351
UpdateConnectivityInfo	354
UpdateConnectorDefinition	359
UpdateCoreDefinition	362
UpdateDeviceDefinition	365
UpdateFunctionDefinition	367
UpdateGroup	370
UpdateGroupCertificateConfiguration	373
UpdateLoggerDefinition	378
UpdateResourceDefinition	380
UpdateSubscriptionDefinition	383
UpdateThingRuntimeConfiguration	386
Endpoints	390
/greengrass/bulk/deployments	392
/greengrass/bulk/deployments/BulkDeploymentId/\$stop	400
/greengrass/bulk/deployments/BulkDeploymentId/detailed-reports	403
/greengrass/bulk/deployments/BulkDeploymentId/status	408
/greengrass/definition/connectors	412
/greengrass/definition/connectors/ConnectorDefinitionId	421
/greengrass/definition/connectors/ConnectorDefinitionId/versions	430
/greengrass/definition/connectors/ConnectorDefinitionId/versions/ ConnectorDefinitionVersionId	439
/greengrass/definition/cores	444

/greengrass/definition/cores/CoreDefinitionId	453
/greengrass/definition/cores/CoreDefinitionId/versions	462
/greengrass/definition/cores/CoreDefinitionId/versions/CoreDefinitionVersionId	471
/greengrass/definition/devices	476
/greengrass/definition/devices/DeviceDefinitionId	485
/greengrass/definition/devices/DeviceDefinitionId/versions	494
/greengrass/definition/devices/DeviceDefinitionId/versions/DeviceDefinitionVersionId ...	503
/greengrass/definition/functions	507
/greengrass/definition/functions/FunctionDefinitionId	521
/greengrass/definition/functions/FunctionDefinitionId/versions	530
/greengrass/definition/functions/FunctionDefinitionId/versions/ FunctionDefinitionVersionId	544
/greengrass/definition/loggers	553
/greengrass/definition/loggers/LoggerDefinitionId	562
/greengrass/definition/loggers/LoggerDefinitionId/versions	571
/greengrass/definition/loggers/LoggerDefinitionId/versions/LoggerDefinitionVersionId .	580
/greengrass/definition/resources	585
/greengrass/definition/resources/ResourceDefinitionId	600
/greengrass/definition/resources/ResourceDefinitionId/versions	609
/greengrass/definition/resources/ResourceDefinitionId/versions/ ResourceDefinitionVersionId	623
/greengrass/definition/subscriptions	633
/greengrass/definition/subscriptions/SubscriptionDefinitionId	642
/greengrass/definition/subscriptions/SubscriptionDefinitionId/versions	651
/greengrass/definition/subscriptions/SubscriptionDefinitionId/versions/ SubscriptionDefinitionVersionId	660
/greengrass/groups	665
/greengrass/groups/GroupId	673
/greengrass/groups/GroupId/certificateauthorities	682
/greengrass/groups/GroupId/certificateauthorities/configuration/expiry	691
/greengrass/groups/GroupId/certificateauthorities/CertificateAuthorityId	699
/greengrass/groups/GroupId/deployments	704
/greengrass/groups/GroupId/deployments/\$reset	712
/greengrass/groups/GroupId/deployments/DeploymentId/status	715
/greengrass/groups/GroupId/role	719
/greengrass/groups/GroupId/versions	731

/greengrass/groups/GroupId/versions/GroupVersionId	740
/greengrass/servicerole	744
/greengrass/things/ThingName/connectivityInfo	755
/greengrass/things/ThingName/runtimeconfig	765
/greengrass/updates	772
/tags/resource-arn	778
Parameters	785
AssociateRoleToGroupRequestBody	787
AssociateServiceRoleToAccountRequestBody	787
BulkDeploymentId	788
CertificateAuthorityId	788
ConnectorDefinitionId	789
ConnectorDefinitionVersionId	789
CoreDefinitionId	790
CoreDefinitionVersionId	790
CreateConnectorDefinitionRequestBody	791
CreateConnectorDefinitionVersionRequestBody	792
CreateCoreDefinitionRequestBody	793
CreateCoreDefinitionVersionRequestBody	794
CreateDeploymentRequestBody	796
CreateDeviceDefinitionRequestBody	797
CreateDeviceDefinitionVersionRequestBody	797
CreateFunctionDefinitionRequestBody	799
CreateFunctionDefinitionVersionRequestBody	801
CreateGroupRequestBody	806
CreateGroupVersionRequestBody	807
CreateLoggerDefinitionRequestBody	809
CreateLoggerDefinitionVersionRequestBody	809
CreateResourceDefinitionRequestBody	811
CreateResourceDefinitionVersionRequestBody	813
CreateSoftwareUpdateJobRequestBody	819
CreateSubscriptionDefinitionRequestBody	821
CreateSubscriptionDefinitionVersionRequestBody	821
DeploymentId	823
DeviceDefinitionId	824
DeviceDefinitionVersionId	824

FunctionDefinitionId	824
FunctionDefinitionVersionId	825
GroupId	825
GroupVersionId	826
LoggerDefinitionId	826
LoggerDefinitionVersionId	827
MaxResults	827
NextToken	828
ResetDeploymentsRequestBody	828
ResourceArn	829
ResourceDefinitionId	829
ResourceDefinitionVersionId	829
RuntimeConfigurationUpdateRequestBody	830
StartBulkDeploymentRequestBody	831
SubscriptionDefinitionId	832
SubscriptionDefinitionVersionId	833
tags	833
ThingName	834
TagKeys	834
UpdateConnectivityInfoRequestBody	835
UpdateDefinitionRequestBody	836
UpdateGroupCertificateConfigurationRequestBody	836
X-Amzn-Client-Token	837
Definitions	837
AssociateRoleToGroupRequest	841
AssociateServiceRoleToAccountRequest	841
AssociateServiceRoleToAccountResponse	842
BulkDeployment	842
BulkDeploymentMetrics	843
BulkDeploymentResult	844
BulkDeploymentResults	846
BulkDeploymentStatus	848
BulkDeployments	848
ConnectivityInfo	849
Connector	850
ConnectorDefinitionVersion	851

Core	852
CoreDefinitionVersion	853
CreateDeploymentRequest	855
CreateDeploymentResponse	856
CreateGroupCertificateAuthorityResponse	856
CreateSoftwareUpdateJobRequest	856
CreateSoftwareUpdateJobResponse	858
DefinitionInformation	859
Deployment	860
DeploymentType	861
Deployments	862
Device	863
DeviceDefinitionVersion	864
DisassociateRoleFromGroupResponse	865
DisassociateServiceRoleFromAccountResponse	865
Empty Schema	866
ErrorDetail	866
ErrorDetails	867
Function	867
FunctionConfiguration	872
FunctionConfigurationEnvironment	876
FunctionDefaultConfig	878
FunctionDefaultExecutionConfig	879
FunctionDefinitionVersion	880
FunctionExecutionConfig	885
FunctionIsolationMode	886
FunctionRunAsConfig	887
GeneralError	888
GetAssociatedRoleResponse	889
GetBulkDeploymentStatusResponse	889
GetConnectivityInfoResponse	892
GetConnectorDefinitionVersionResponse	893
GetCoreDefinitionVersionResponse	895
GetDeploymentStatusResponse	897
GetDeviceDefinitionVersionResponse	899
GetFunctionDefinitionVersionResponse	901

GetGroupCertificateAuthorityResponse	907
GetGroupVersionResponse	908
GetLoggerDefinitionVersionResponse	910
GetResourceDefinitionVersionResponse	912
GetServiceRoleForAccountResponse	919
GetSubscriptionDefinitionVersionResponse	919
GetThingRuntimeConfigurationResponse	922
Group	923
Group Certificate Authority Properties	923
GroupCertificateConfiguration	924
GroupInformation	925
GroupOwnerSetting	926
GroupVersion	927
IDList	928
ListBulkDeploymentDetailedReportsResponse	928
ListBulkDeploymentsResponse	931
ListDefinitionsResponse	932
ListDeploymentsRequest	934
ListDeploymentsResponse	935
ListGroupCertificateAuthoritiesResponse	937
ListGroupsResponse	937
ListVersionsResponse	939
LocalDeviceResourceData	941
LocalVolumeResourceData	942
Logger	943
LoggerComponent	944
LoggerDefinitionVersion	944
LoggerLevel	946
LoggerType	946
Permission	946
ResetDeploymentsRequest	946
ResetDeploymentsResponse	947
Resource	947
ResourceAccessPolicy	953
ResourceDataContainer	954
ResourceDefinitionVersion	959

ResourceDownloadOwnerSetting	965
RuntimeConfiguration	966
RuntimeConfigurationUpdate	967
S3MachineLearningModelResourceData	967
S3UrlSignerRole	969
SageMakerMachineLearningModelResourceData	969
SecretsManagerSecretResourceData	970
SoftwareToUpdate	971
StartBulkDeploymentRequest	971
StartBulkDeploymentResponse	972
Subscription	973
SubscriptionDefinitionVersion	974
TelemetryConfiguration	976
TelemetryConfigurationUpdate	976
UpdateAgentLogLevel	977
UpdateConnectivityInfoRequest	977
UpdateConnectivityInfoResponse	978
UpdateTargets	979
UpdateTargetsArchitecture	979
UpdateTargetsOperatingSystem	979
VersionInformation	980
Versions	981
tags	982
updateGroupCertificateConfigurationRequest	982

End of support notice: On October 7th, 2026, AWS will discontinue support for AWS IoT Greengrass Version 1. After October 7th, 2026, you will no longer be able to access the AWS IoT Greengrass V1 resources. For more information, please visit [Migrate from AWS IoT Greengrass Version 1](#).

Greengrass V1 API

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 IoT Greengrass, specify `application/hal+json` for the `accept` request header.

For more information about how AWS IoT Greengrass works, see the [AWS IoT Greengrass Version 1 Developer Guide](#).

Contents

- [Actions](#)
- [Endpoints](#)
- [Parameters](#)
- [Definitions](#)

Actions

The following actions are supported:

- [AssociateRoleToGroup](#)
- [AssociateServiceRoleToAccount](#)
- [CreateConnectorDefinition](#)
- [CreateConnectorDefinitionVersion](#)
- [CreateCoreDefinition](#)
- [CreateCoreDefinitionVersion](#)
- [CreateDeployment](#)
- [CreateDeviceDefinition](#)
- [CreateDeviceDefinitionVersion](#)

- [CreateFunctionDefinition](#)
- [CreateFunctionDefinitionVersion](#)
- [CreateGroup](#)
- [CreateGroupCertificateAuthority](#)
- [CreateGroupVersion](#)
- [CreateLoggerDefinition](#)
- [CreateLoggerDefinitionVersion](#)
- [CreateResourceDefinition](#)
- [CreateResourceDefinitionVersion](#)
- [CreateSoftwareUpdateJob](#)
- [CreateSubscriptionDefinition](#)
- [CreateSubscriptionDefinitionVersion](#)
- [DeleteConnectorDefinition](#)
- [DeleteCoreDefinition](#)
- [DeleteDeviceDefinition](#)
- [DeleteFunctionDefinition](#)
- [DeleteGroup](#)
- [DeleteLoggerDefinition](#)
- [DeleteResourceDefinition](#)
- [DeleteSubscriptionDefinition](#)
- [DisassociateRoleFromGroup](#)
- [DisassociateServiceRoleFromAccount](#)
- [GetAssociatedRole](#)
- [GetBulkDeploymentStatus](#)
- [GetConnectivityInfo](#)
- [GetConnectorDefinition](#)
- [GetConnectorDefinitionVersion](#)
- [GetCoreDefinition](#)
- [GetCoreDefinitionVersion](#)

- [GetDeploymentStatus](#)
- [GetDeviceDefinition](#)
- [GetDeviceDefinitionVersion](#)
- [GetFunctionDefinition](#)
- [GetFunctionDefinitionVersion](#)
- [GetGroup](#)
- [GetGroupCertificateAuthority](#)
- [GetGroupCertificateConfiguration](#)
- [GetGroupVersion](#)
- [GetLoggerDefinition](#)
- [GetLoggerDefinitionVersion](#)
- [GetResourceDefinition](#)
- [GetResourceDefinitionVersion](#)
- [GetServiceRoleForAccount](#)
- [GetSubscriptionDefinition](#)
- [GetSubscriptionDefinitionVersion](#)
- [GetThingRuntimeConfiguration](#)
- [ListBulkDeploymentDetailedReports](#)
- [ListBulkDeployments](#)
- [ListConnectorDefinitionVersions](#)
- [ListConnectorDefinitions](#)
- [ListCoreDefinitionVersions](#)
- [ListCoreDefinitions](#)
- [ListDeployments](#)
- [ListDeviceDefinitionVersions](#)
- [ListDeviceDefinitions](#)
- [ListFunctionDefinitionVersions](#)
- [ListFunctionDefinitions](#)
- [ListGroupCertificateAuthorities](#)

- [ListGroupVersions](#)
- [ListGroups](#)
- [ListLoggerDefinitionVersions](#)
- [ListLoggerDefinitions](#)
- [ListResourceDefinitionVersions](#)
- [ListResourceDefinitions](#)
- [ListSubscriptionDefinitionVersions](#)
- [ListSubscriptionDefinitions](#)
- [ListTagsForResource](#)
- [ResetDeployments](#)
- [StartBulkDeployment](#)
- [StopBulkDeployment](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateConnectivityInfo](#)
- [UpdateConnectorDefinition](#)
- [UpdateCoreDefinition](#)
- [UpdateDeviceDefinition](#)
- [UpdateFunctionDefinition](#)
- [UpdateGroup](#)
- [UpdateGroupCertificateConfiguration](#)
- [UpdateLoggerDefinition](#)
- [UpdateResourceDefinition](#)
- [UpdateSubscriptionDefinition](#)
- [UpdateThingRuntimeConfiguration](#)

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:

```
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"  
}
```

Parameters:

GroupId

The ID of the Greengrass group.

where used: path; required: true

type: string

AssociateRoleToGroupRequestBody

where used: body; required: true

```
{  
  "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](#)

```
{
  "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](#)

```
{
  "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](#)

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](#)

```
{
```

```
"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](#)

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:

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

cli-input-json format:

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

Parameters:

AssociateServiceRoleToAccountRequestBody

where used: body; required: true

```
{
  "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](#)

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

AssociateServiceRoleToAccountResponse

type: object

AssociatedAt

The time when the service role was associated with the account.

type: string

400

Invalid request.

[GeneralError](#)

```
{
  "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](#)

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](#)

```
{
```

```
"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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

CreateConnectorDefinition

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

URI: 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

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](#)

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

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

where used: header; required: false

type: string

Responses:

200 (CreateConnectorDefinitionResponse)

DefinitionInformation

```
{
  "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

```
{
  "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](#)

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

The ID of the connector definition.

where used: path; required: true

type: string

CreateConnectorDefinitionVersionRequestBody

where used: body; required: true

```
{
  "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](#)

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

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

where used: header; required: false

type: string

Responses:

200 (CreateConnectorDefinitionVersionResponse)

VersionInformation

```
{
  "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

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

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.

URI: POST /greengrass/definition/cores

Produces: 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

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](#)

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

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:**200 (CreateCoreDefinitionResponse)**

DefinitionInformation

```
{
  "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

```
{
  "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](#)

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:

```
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:

```
{
```

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

Parameters:

CoreDefinitionId

The ID of the core definition.

where used: path; required: true

type: string

CreateCoreDefinitionVersionRequestBody

where used: body; required: true

```
{
  "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

items: [Core](#)

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](#)

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200 (CreateCoreDefinitionVersionResponse)

VersionInformation

```
{
  "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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

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

The ID of the Greengrass group.

where used: path; required: true

type: string

CreateDeploymentRequestBody

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

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200

Success. The group was deployed.

CreateDeploymentResponse

```
{
  "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

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

CreateDeviceDefinition

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

URI: POST `/greengrass/definition/devices`

Produces: `application/json`

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"
}
```

Parameters:

CreateDeviceDefinitionRequestBody

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"
  }
}
```

schema:

Name

The name of the device definition.

type: string

InitialVersion

Information about a device definition version.

type: object

Devices

A list of client devices in the definition version.

type: array

items: [Device](#)

Device

Information about a client device.

type: object

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

Id

A descriptive or arbitrary ID for the client 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 client device.

type: string

CertificateArn

The ARN of the certificate associated with the client device.

type: string

SyncShadow

If true, the client 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

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200 (CreateDeviceDefinitionResponse)

DefinitionInformation

```
{
  "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

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

CreateDeviceDefinitionVersion

Creates a version of a client device definition that has already been defined.

URI: POST /greengrass/definition/devices/*DeviceDefinitionId*/versions

Produces: application/json

CLI:

```
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:

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

Parameters:

DeviceDefinitionId

The ID of the device definition.

where used: path; required: true

type: string

CreateDeviceDefinitionVersionRequestBody

where used: body; required: true

```
{  
  "Devices": [  

```

```
{
  "Id": "string",
  "ThingArn": "string",
  "CertificateArn": "string",
  "SyncShadow": true
}
```

schema:

DeviceDefinitionVersion

Information about a device definition version.

type: object

Devices

A list of client devices in the definition version.

type: array

items: [Device](#)

Device

Information about a client device.

type: object

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

Id

A descriptive or arbitrary ID for the client 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 client device.

type: string

CertificateArn

The ARN of the certificate associated with the client device.

type: string

SyncShadow

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

type: boolean

X-Amzn-Client-Token

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200 (CreateDeviceDefinitionVersionResponse)

VersionInformation

```
{
  "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

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

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:

```
{
  "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"
          }
        }
      }
    }
  ]
}
```

```

        }
    },
    "FunctionRuntimeOverride": "string"
}
}
],
},
"tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
},
"AmznClientToken": "string"
}

```

Parameters:

CreateFunctionDefinitionRequestBody

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"
        }
      ],
      "AccessSysfs": true,
      "Execution": {
        "IsolationMode": "GreengrassContainer|NoContainer",
        "RunAs": {
          "Uid": 1001,
          "Gid": 1002
        }
      }
    },
    "FunctionRuntimeOverride": "string"
  }
}
]
},
"tags": {
  "additionalProperty0": "string",
  "additionalProperty1": "string",
  "additionalProperty2": "string"
}
}

```

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](#)

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](#)

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

FunctionRuntimeOverride

The Lambda runtime supported by Greengrass which is to be used instead of the one specified in the Lambda function.

type: string

tags

The resource tags.

type: object

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

X-Amzn-Client-Token

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200 (CreateFunctionDefinitionResponse)

DefinitionInformation

```
{
  "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](#)

```
{  
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

CreateFunctionDefinitionVersion

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

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

```

Parameters:

FunctionDefinitionId

The ID of the function definition.

where used: path; required: true

type: string

CreateFunctionDefinitionVersionRequestBody

Information about the function definition version.

where used: body; required: true

```
{
  "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
            }
          }
        }
      }
    }
  ],
}
```

```
    "FunctionRuntimeOverride": "string"
  }
}
]
```

schema:

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](#)

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](#)

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

FunctionRuntimeOverride

The Lambda runtime supported by Greengrass which is to be used instead of the one specified in the Lambda function.

type: string

X-Amzn-Client-Token

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200 (CreateFunctionDefinitionVersionResponse)

VersionInformation

```
{
  "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](#)

```
{
  "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](#)

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/aws-labs/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> \
  [--initial-version <value>] \
  [--tags <value>] \
```

```
[--amzn-client-token <value>] \  
[--cli-input-json <value>] \  
[--generate-cli-skeleton]
```

cli-input-json format:

```
{  
  "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"  
}
```

Parameters:

[CreateGroupRequestBody](#)

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.

type: object

CoreDefinitionVersionArn

The ARN of the core definition version for this group.

type: string

DeviceDefinitionVersionArn

The ARN of the client 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

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

```
{
  "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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

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:

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

cli-input-json format:

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

Parameters:

GroupId

The ID of the Greengrass group.

where used: path; required: true

type: string

X-Amzn-Client-Token

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

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

CreateGroupCertificateAuthorityResponse

type: object

GroupCertificateAuthorityArn

The ARN of the group certificate authority.

type: string

400

Invalid request.

GeneralError

```
{
  "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](#)

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](#)

```
{
```

```
"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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

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

Parameters:

GroupId

The ID of the Greengrass group.

where used: path; required: true

type: string

CreateGroupVersionRequestBody

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 client 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

X-Amzn-Client-Token

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

```
{
  "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

```
{
  "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](#)

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.

URI: POST /greengrass/definition/loggers

Produces: application/json

CLI:

```
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:

```
{  
  "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](#)

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": {
  "additionalProperty0": "string",
  "additionalProperty1": "string",
  "additionalProperty2": "string"
}
}
```

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](#)

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

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200 (CreateLoggerDefinitionResponse)

DefinitionInformation

```
{
  "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](#)

```
{
  "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](#)

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:

```
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:

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

Parameters:

LoggerDefinitionId

The ID of the logger definition.

where used: path; required: true

type: string

CreateLoggerDefinitionVersionRequestBody

where used: body; required: true

```
{  
  "Loggers": [  
    {  
      "Id": "string",  
      "Type": "FileSystem|AWSCloudWatch",  
      "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](#)

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

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200 (CreateLoggerDefinitionVersionResponse)

VersionInformation

```
{
  "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

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

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": [
      {
        "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"
}

```

Parameters:

CreateResourceDefinitionRequestBody

where used: body; required: true

```

{
    "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": {
    "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](#)

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

tags

The resource tags.

type: object

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

X-Amzn-Client-Token

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200 (CreateResourceDefinitionResponse)

DefinitionInformation

```
{
  "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

```
{
  "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](#)

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.

URI: POST /greengrass/definition/resources/*ResourceDefinitionId*/versions

Produces: application/json

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"
}

```

Parameters:

ResourceDefinitionId

The ID of the resource definition.

where used: path; required: true

type: string

CreateResourceDefinitionVersionRequestBody

Information about the resource definition version.

where used: body; required: true

```

{
  "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"
    ]
}
}
]
}

```

schema:

ResourceDefinitionVersion

Information about a resource definition version.

type: object

Resources

A list of resources.

type: array

items: [Resource](#)

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

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200 (CreateResourceDefinitionVersionResponse)

VersionInformation

```
{
  "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

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: 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:

```
{
  "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

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"
}
```

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

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200

Success.

[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

400

Invalid request.

[GeneralError](#)

```
{
  "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](#)

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

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: 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:

```
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:

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

```
"additionalProperty2": "string",
},
"AmznClientToken": "string"
}
```

Parameters:

CreateSubscriptionDefinitionRequestBody

where used: body; required: true

```
{
  "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.

type: array

items: [Subscription](#)

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

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200 (CreateSubscriptionDefinitionResponse)

DefinitionInformation

```
{
  "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](#)

```
{  
  "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](#)

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:

```
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:

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

Parameters:

SubscriptionDefinitionId

The ID of the subscription definition.

where used: path; required: true

type: string

CreateSubscriptionDefinitionVersionRequestBody

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](#)

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

X-Amzn-Client-Token

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200 (CreateSubscriptionDefinitionVersionResponse)

VersionInformation

```
{
  "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](#)

```
{  
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

DeleteConnectorDefinition

Deletes a connector definition.

URI: DELETE /greengrass/definition/connectors/*ConnectorDefinitionId*

Produces: application/json

CLI:

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

cli-input-json format:

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

Parameters:

ConnectorDefinitionId

The ID of the connector definition.

where used: path; required: true

type: string

Responses:

200

Success.

Empty

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

DeleteCoreDefinition

Deletes 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**

The ID of the core definition.

where used: path; required: true

type: string

Responses:

200

Success.

[Empty](#)

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{  
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: 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:


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

Parameters:

DeviceDefinitionId

The ID of the device definition.

where used: path; required: true

type: string

Responses:

200

Success.

Empty

```
{
}
```

Empty Schema

Empty

type: object

400

Invalid request.

GeneralError

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

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

The ID of the function definition.

where used: path; required: true

type: string

Responses:

200

Success.

Empty

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{
  "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](#)

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. Before you delete a group that is deployed, we recommend that you [reset the group's deployments](#). This step removes the group's information from the core, so you can use the core with another Greengrass group.

URI: DELETE /greengrass/groups/*GroupId*

Produces: application/json

CLI:

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

cli-input-json format:

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

Parameters:**GroupId**

The ID of the Greengrass group.

where used: path; required: true

type: string

Responses:

200

Success.

[Empty](#)

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{  
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

DeleteLoggerDefinition

Deletes a logger definition.

URI: DELETE /greengrass/definition/loggers/*LoggerDefinitionId*

Produces: application/json

CLI:

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

```
[--generate-cli-skeleton]
```

cli-input-json format:

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

Parameters:

LoggerDefinitionId

The ID of the logger definition.

where used: path; required: true

type: string

Responses:

200

Success.

Empty

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

GeneralError

```
{
```



```
"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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

DeleteResourceDefinition

Deletes a resource definition.

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

Produces: application/json

CLI:

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

cli-input-json format:

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

Parameters:

ResourceDefinitionId

The ID of the resource definition.

where used: path; required: true

type: string

Responses:

200

Success.

Empty

```
{
}
```

Empty Schema

Empty

type: object

400

Invalid request.

GeneralError

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

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

The ID of the subscription definition.

where used: path; required: true

type: string

Responses:

200

Success.

[Empty](#)

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{  
  "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](#)

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.

URI: DELETE /greengrass/groups/*GroupId*/role

CLI:

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

cli-input-json format:

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

Parameters:

GroupId

The ID of the Greengrass group.

where used: path; required: true

type: string

Responses:

200

Success.

DisassociateRoleFromGroupResponse

```
{
  "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

```
{
  "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](#)

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](#)

```
{
  "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](#)

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](#)

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

DisassociateServiceRoleFromAccountResponse

type: object

DisassociatedAt

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

type: string

500

Server error.

[GeneralError](#)

```
{
  "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](#)

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:

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

cli-input-json format:

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

Parameters:**GroupId**

The ID of the Greengrass group.

where used: path; required: true

type: string

Responses:

200

Success.

[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

400

Invalid request.

[GeneralError](#)

```
{
  "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](#)

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](#)

```
{  
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

GetBulkDeploymentStatus

Returns the status of a bulk deployment.

URI: GET /greengrass/bulk/deployments/*BulkDeploymentId*/status

Produces: application/json

CLI:

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

cli-input-json format:

```
{
  "BulkDeploymentId": "string"
}
```

Parameters:

BulkDeploymentId

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

```
{
```



```
"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

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

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](#)

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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

GetConnectivityInfo

Retrieves the connectivity information for a core.

URI: GET /greengrass/things/*ThingName*/connectivityInfo

Produces: application/json

CLI:

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

cli-input-json format:

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

Parameters:

ThingName

The thing name.

where used: path; required: true

type: string

Responses:

200

Success.

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](#)

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](#)

```
{
```

```
"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](#)

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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

GetConnectorDefinition

Retrieves information about a connector definition.

URI: GET /greengrass/definition/connectors/*ConnectorDefinitionId*

Produces: application/json

CLI:

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

cli-input-json format:

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

Parameters:**ConnectorDefinitionId**

The ID of the connector definition.

where used: path; required: true

type: string

Responses:

200 (GetConnectorDefinitionResponse)

DefinitionInformation

```
{
  "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

```
{
  "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](#)

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:

```
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:

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

Parameters:

ConnectorDefinitionId

The ID of the connector definition.

where used: path; required: true

type: 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.

where used: path; required: true

type: string

NextToken

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

```
{
  "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](#)

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

```
{
  "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](#)

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

The ID of the core definition.

where used: path; required: true

type: string

Responses:

200 (GetCoreDefinitionResponse)

DefinitionInformation

```
{  
  "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

```
{
  "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](#)

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:

```
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"
}
```

Parameters:

CoreDefinitionId

The ID of the core definition.

where used: path; required: true

type: 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.

where used: path; required: true

type: string

Responses:

200

Success.

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](#)

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

```
{
  "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](#)

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

Produces: application/json

CLI:

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

cli-input-json format:

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

Parameters:

GroupId

The ID of the Greengrass group.

where used: path; required: true

type: string

DeploymentId

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](#)

```
{
  "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

ErrorDetails

A list of error details.

type: array

items: [ErrorDetail](#)

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](#)

```
{
```

```
"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](#)

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

The ID of the device definition.

where used: path; required: true

type: string

Responses:

200 (GetDeviceDefinitionResponse)

DefinitionInformation

```
{
```

```
"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

```
{
  "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](#)

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

The ID of the device definition.

where used: path; required: true

type: 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.

where used: path; required: true

type: string

NextToken

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 (GetDeviceDefinitionVersionResponse)

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](#)

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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

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

cli-input-json format:

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

Parameters:

FunctionDefinitionId

The ID of the function definition.

where used: path; required: true

type: string

Responses:

200 (GetFunctionDefinitionResponse)

Success.

DefinitionInformation

```
{
  "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

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

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:

```
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:

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

Parameters:

FunctionDefinitionId

The ID of the function definition.

where used: path; required: true

type: 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.

where used: path; required: true

type: string

NextToken

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

```
{
  "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
          }
        }
      },
      "FunctionRuntimeOverride": "string"
    }
  }
],
"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](#)

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](#)

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

FunctionRuntimeOverride

The Lambda runtime supported by Greengrass which is to be used instead of the one specified in the Lambda function.

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](#)

```
{
  "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](#)

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:

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

cli-input-json format:

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

Parameters:

GroupId

The ID of the Greengrass group.

where used: path; required: true

type: string

Responses:

200 (GetGroupResponse)

Success.

DefinitionInformation

```
{
```

```
"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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

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

The ID of the Greengrass group.

where used: path; required: true

type: string

CertificateAuthorityId

The ID of the certificate authority.

where used: path; required: true

type: string

Responses:

200

Success. The response body contains the PKI configuration.

GetGroupCertificateAuthorityResponse

```
{
  "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](#)

```
{
  "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](#)

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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

GetGroupCertificateConfiguration

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

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

Produces: application/json

CLI:

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

cli-input-json format:

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

Parameters:

GroupId

The ID of the Greengrass group.

where used: path; required: true

type: string

Responses:

200

Success. The response body contains the PKI configuration.

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

400

Invalid request.

GeneralError

```
{
  "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](#)

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](#)

```
{
  "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](#)

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

The ID of the Greengrass group.

where used: path; required: true

type: 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.

where used: path; required: true

type: string

Responses:

200

Success.

GetGroupVersionResponse

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

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 client 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](#)

```
{
  "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](#)

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

The ID of the logger definition.

where used: path; required: true

type: string

Responses:

200 (GetLoggerDefinitionResponse)

DefinitionInformation

```
{
  "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

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: 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:

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

Parameters:

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.

where used: path; required: true

type: string

LoggerDefinitionId

The ID of the logger definition.

where used: path; required: true

type: string

NextToken

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

```
{
  "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](#)

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](#)

```
{
```

```
"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](#)

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.

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

Produces: application/json

CLI:

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

cli-input-json format:

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

Parameters:

ResourceDefinitionId

The ID of the resource definition.

where used: path; required: true

type: string

Responses:

200 (GetResourceDefinitionResponse)

Success.

DefinitionInformation

```
{
  "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

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

GetResourceDefinitionVersion

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

URI: 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]
```

cli-input-json format:

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

Parameters:

ResourceDefinitionId

The ID of the resource definition.

where used: path; required: true

type: 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.

where used: path; required: true

type: string

Responses:

200

Success.

GetResourceDefinitionVersionResponse

```
{
  "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](#)

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

400

Invalid request.

[GeneralError](#)

```
{
```

```
"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](#)

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](#)

```
{
  "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

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: 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

The ID of the subscription definition.

where used: path; required: true

type: string

Responses:

200 (GetSubscriptionDefinitionResponse)

DefinitionInformation

```
{
  "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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

GetSubscriptionDefinitionVersion

Retrieves information about a subscription definition version.

URI: GET /greengrass/definition/subscriptions/*SubscriptionDefinitionId*/versions/*SubscriptionDefinitionVersionId*

Produces: application/json

CLI:

```
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:

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

Parameters:

SubscriptionDefinitionId

The ID of the subscription definition.

where used: path; required: true

type: 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.

where used: path; required: true

type: string

NextToken

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

```
{
  "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](#)

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

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

GetThingRuntimeConfiguration

Retrieves the runtime configuration of a Greengrass core. Before you can retrieve the runtime configuration, you must use [UpdateThingRuntimeConfiguration](#) to create a runtime configuration for the core.

URI: GET /greengrass/things/{ThingName}/runtimeconfig

Produces: application/json

CLI:

```
aws greengrass get-thing-runtime-configuration \
```

```
--thing-name <value> \  
[--cli-input-json <value>] \  
[--generate-cli-skeleton]
```

cli-input-json format:

```
{  
  "ThingName": ""  
}
```

Parameters:

[ThingName](#)

The thing name.

where used: path; required: true

type: string

Responses:

200

200 response

[GetThingRuntimeConfigurationResponse](#)

```
{  
  "RuntimeConfiguration": {  
    "TelemetryConfiguration": {  
      "Telemetry": "On|Off",  
      "ConfigurationSyncStatus": "InSync|OutOfSync"  
    }  
  }  
}
```

GetThingRuntimeConfigurationResponse

Information about the runtime configuration for a thing.

type: object

RuntimeConfiguration

The runtime configuration for a thing.

type: object

TelemetryConfiguration

The configuration setting to run telemetry.

type: object

required: ["Telemetry"]

Telemetry

The configuration setting to turn on or turn off telemetry.

type: string

enum: ["On", "Off"]

ConfigurationSyncStatus

The synchronization status of the device-reported configuration with the desired configuration.

type: string

enum: ["InSync", "OutOfSync"]

400

400 response

GeneralError

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

500

500 response

[GeneralError](#)

```
{  
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

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:

```
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"
}
```

Parameters:

BulkDeploymentId

The ID of the bulk deployment.

where used: path; required: true

type: string

MaxResults

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

NextToken

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

```
{
  "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](#)

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](#)

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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

ListBulkDeployments

Returns a list of bulk deployments.

URI: GET /greengrass/bulk/deployments

Produces: application/json

CLI:

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

cli-input-json format:

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

Parameters:

MaxResults

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

NextToken

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 bulk deployments.

[ListBulkDeploymentsResponse](#)

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

ListBulkDeploymentsResponse

type: object

BulkDeployments

type: array

items: [BulkDeployment](#)

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

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

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

Produces: 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

The ID of the connector definition.

where used: path; required: true

type: string

NextToken

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

where used: query; required: false

type: string

MaxResults

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

Responses:

200 (ListConnectorDefinitionVersionsResponse)

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](#)

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](#)

```
{
  "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](#)

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:

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

cli-input-json format:

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

```
}
```

Parameters:

MaxResults

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

NextToken

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

```
{
  "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](#)

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

Produces: application/json

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

The ID of the core definition.

where used: path; required: true

type: string

NextToken

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

where used: query; required: false

type: string

MaxResults

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

Responses:

200 (ListCoreDefinitionVersionsResponse)

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](#)

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](#)

```
{
  "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](#)

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

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

NextToken

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

```
{
  "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](#)

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.

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

The ID of the Greengrass group.

where used: path; required: true

type: string

MaxResults

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

NextToken

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

```
{
  "Deployments": [
    {
      "GroupArn": "string",
      "DeploymentId": "string",
      "DeploymentArn": "string",
      "DeploymentType": "NewDeployment|Redeployment|ResetDeployment|
ForceResetDeployment",
      "CreatedAt": "string"
    }
  ],
  "NextToken": "string"
}
```

ListDeploymentsResponse

type: object

Deployments

type: array

items: [Deployment](#)

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](#)

```
{
  "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](#)

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

The ID of the device definition.

where used: path; required: true

type: string

NextToken

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

where used: query; required: false

type: string

MaxResults

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

Responses:

200 (ListDeviceDefinitionVersionsResponse)

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](#)

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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

ListDeviceDefinitions

Retrieves a list of device definitions.

URI: GET /greengrass/definition/devices

Produces: application/json

CLI:

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

cli-input-json format:

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

```
}
```

Parameters:

MaxResults

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

NextToken

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

```
{
  "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](#)

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.

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

Produces: application/json

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

The ID of the function definition.

where used: path; required: true

type: string

NextToken

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

where used: query; required: false

type: string

MaxResults

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

Responses:

200 (ListFunctionDefinitionVersionsResponse)

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](#)

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

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

ListFunctionDefinitions

Retrieves a list of Lambda function definitions.

URI: GET /greengrass/definition/functions

Produces: 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

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

NextToken

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

```
{
  "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](#)

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

ListGroupCertificateAuthorities

Retrieves the current CAs for a group.

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

Produces: application/json

CLI:

```
aws greengrass list-group-certificate-authorities \
  --group-id <value> \
  [--cli-input-json <value>] \
```



```
[--generate-cli-skeleton]
```

cli-input-json format:

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

Parameters:

GroupId

The ID of the Greengrass group.

where used: path; required: true

type: string

Responses:

200

Success. The response body contains the PKI configuration.

ListGroupCertificateAuthoritiesResponse

```
{  
  "GroupCertificateAuthorities": [  
    {  
      "GroupCertificateAuthorityId": "string",  
      "GroupCertificateAuthorityArn": "string"  
    }  
  ]  
}
```

ListGroupCertificateAuthoritiesResponse

type: object

GroupCertificateAuthorities

A list of certificate authorities associated with the group.

type: array

items: [GroupCertificateAuthorityProperties](#)

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](#)

```
{
  "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](#)

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](#)

```
{
  "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](#)

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:

```
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:

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

Parameters:

GroupId

The ID of the Greengrass group.

where used: path; required: true

type: string

NextToken

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

where used: query; required: false

type: string

MaxResults

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](#)

```
{
  "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](#)

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](#)

```
{
  "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](#)

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

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

NextToken

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 (ListGroupResponse)

ListGroupsResponse

```
{  
  "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](#)

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

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

The ID of the logger definition.

where used: path; required: true

type: string

NextToken

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

where used: query; required: false

type: string

MaxResults

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

Responses:

200 (ListLoggerDefinitionVersionsResponse)

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](#)

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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

ListLoggerDefinitions

Retrieves a list of logger definitions.

URI: GET /greengrass/definition/loggers

Produces: application/json

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"
```

```
}
```

Parameters:

MaxResults

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

NextToken

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

```
{
  "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](#)

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.

URI: GET /greengrass/definition/resources/*ResourceDefinitionId*/versions

Produces: application/json

CLI:

```
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:

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

Parameters:

ResourceDefinitionId

The ID of the resource definition.

where used: path; required: true

type: string

NextToken

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

where used: query; required: false

type: string

MaxResults

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

Responses:

200 (ListResourceDefinitionVersionsResponse)

[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](#)

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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

ListResourceDefinitions

Retrieves a list of resource definitions.

URI: GET /greengrass/definition/resources

Produces: application/json

CLI:

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

cli-input-json format:

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

Parameters:

MaxResults

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

NextToken

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

```
{
  "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](#)

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:

```
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:

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

Parameters:

SubscriptionDefinitionId

The ID of the subscription definition.

where used: path; required: true

type: string

NextToken

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

where used: query; required: false

type: string

MaxResults

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

Responses:

200 (ListSubscriptionDefinitionVersionsResponse)

[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](#)

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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

ListSubscriptionDefinitions

Retrieves a list of subscription definitions.

URI: GET /greengrass/definition/subscriptions

Produces: application/json

CLI:

```
aws greengrass list-subscription-definitions \
  [--max-results <value>] \
  [--next-token <value>] \
```

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

cli-input-json format:

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

Parameters:

MaxResults

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

NextToken

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

```
{  
  "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](#)

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](#)

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](#)

```
{
  "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](#)

```
{
  "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](#)

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:

```
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:

```
{
  "GroupId": "string",
  "Force": "boolean",
  "AmznClientToken": "string"
}
```

Parameters:

GroupId

The ID of the Greengrass group.

where used: path; required: true

type: string

[ResetDeploymentsRequestBody](#)

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](#)

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](#)

```
{
  "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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

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>] \
```

```
[--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"  
}
```

Parameters:

StartBulkDeploymentRequestBody

where used: body; required: true

```
{  
  "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.

```
{"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

[X-Amzn-Client-Token](#)

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Responses:

200

Success.

[StartBulkDeploymentResponse](#)

```
{
  "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](#)

```
{
  "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](#)

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

cli-input-json format:

```
{
  "BulkDeploymentId": "string"
}
```

Parameters:

BulkDeploymentId

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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

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

The Amazon Resource Name (ARN) of the resource to attach the tags to.

where used: path; required: true

type: string

[tags](#)

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](#)

```
{
  "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](#)

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](#)

An array of tag keys to delete.

where used: query; required: true

type: array of strings

[ResourceArn](#)

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](#)

```
{
```



```
"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](#)

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. Client 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>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "ThingName": "string",
  "ConnectivityInfo": [
    {
      "Id": "string",
      "HostAddress": "string",
      "PortNumber": "int32-integer",
      "Metadata": "string"
    }
  ]
}
```

Parameters:

ThingName

The thing name.

where used: path; required: true

type: string

UpdateConnectivityInfoRequestBody

The connectivity information.

where used: body; required:

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

schema:

Information required to update a Greengrass core's connectivity.

ConnectivityInfo

A list of connectivity info.

type: array

items: [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

Responses:

200

Success.

[UpdateConnectivityInfoResponse](#)

```
{
  "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](#)

```
{
  "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](#)

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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

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:

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

cli-input-json format:

```
{
  "ConnectorDefinitionId": "string",
  "Name": "string"
}
```

Parameters:**ConnectorDefinitionId**

The ID of the connector definition.

where used: path; required: true

type: string

UpdateDefinitionRequestBody

where used: body; required: true

```
{  
  "Name": "string"  
}
```

schema:

Name

The name of the definition.

required: true

type: string

Responses:

200

Success.

Empty

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

GeneralError

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

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

cli-input-json format:

```
{
  "CoreDefinitionId": "string",
  "Name": "string"
}
```

Parameters:

CoreDefinitionId

The ID of the core definition.

where used: path; required: true

type: string

UpdateDefinitionRequestBody

where used: body; required: true

```
{  
  "Name": "string"  
}
```

schema:

Name

The name of the definition.

required: true

type: string

Responses:

200

Success.

[Empty](#)

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{  
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

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

The ID of the device definition.

where used: path; required: true

type: string

UpdateDefinitionRequestBody

where used: body; required: true

```
{  
  "Name": "string"  
}
```

schema:**Name**

The name of the definition.

required: true

type: string

Responses:**200**

Success.

[Empty](#)

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{  
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

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:

```
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"  
}
```

Parameters:

FunctionDefinitionId

The ID of the function definition.

where used: path; required: true

type: string

UpdateDefinitionRequestBody

where used: body; required: true

```
{  
  "Name": "string"  
}
```

schema:

Name

The name of the definition.

required: true

type: string

Responses:

200

Success.

[Empty](#)

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

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

GeneralError

General error information.

type: object

required: ["Message"]


```
[--generate-cli-skeleton]
```

cli-input-json format:

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

Parameters:

GroupId

The ID of the Greengrass group.

where used: path; required: true

type: string

UpdateDefinitionRequestBody

where used: body; required: true

```
{
  "Name": "string"
}
```

schema:

Name

The name of the definition.

required: true

type: string

Responses:

200

Success.

Empty

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

GeneralError

```
{  
  "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](#)

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:

```
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"
}
```

Parameters:

GroupId

The ID of the Greengrass group.

where used: path; required: true

type: string

UpdateGroupCertificateConfigurationRequestBody

where used: body; required: true

```
{
  "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

```
{
  "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](#)

```
{
  "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](#)

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](#)

```
{  
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: 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:

```
{
  "LoggerDefinitionId": "string",
  "Name": "string"
}
```

Parameters:

LoggerDefinitionId

The ID of the logger definition.

where used: path; required: true

type: string

UpdateDefinitionRequestBody

where used: body; required: true

```
{
  "Name": "string"
}
```

schema:**Name**

The name of the definition.

required: true

type: string

Responses:**200**

Success.

[Empty](#)

```
{
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{
  "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](#)

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:

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

cli-input-json format:

```
{
  "ResourceDefinitionId": "string",
  "Name": "string"
}
```

Parameters:

ResourceDefinitionId

The ID of the resource definition.

where used: path; required: true

type: string

UpdateDefinitionRequestBody

where used: body; required: true

```
{
  "Name": "string"
}
```

schema:

Name

The name of the definition.

required: true

type: string

Responses:

200

Success.

[Empty](#)

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{  
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

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 \
  --subscription-definition-id <value> \
  [--name <value>] \
```

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

cli-input-json format:

```
{  
  "SubscriptionDefinitionId": "string",  
  "Name": "string"  
}
```

Parameters:

SubscriptionDefinitionId

The ID of the subscription definition.

where used: path; required: true

type: string

UpdateDefinitionRequestBody

where used: body; required: true

```
{  
  "Name": "string"  
}
```

schema:

Name

The name of the definition.

required: true

type: string

Responses:

200

Success.

[Empty](#)

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{  
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

UpdateThingRuntimeConfiguration

Updates the runtime configuration of a Greengrass core to turn on or turn off telemetry.

URI: PUT /greengrass/things/{ThingName}/runtimeconfig

produces: application/json

CLI:

```
aws greengrass update-thing-runtime-configuration \
  --thing-name <value> \
  [--telemetry-configuration <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "TelemetryConfiguration": {
    "Telemetry": "On|Off"
```

```
},  
  "ThingName": ""  
}
```

Parameters:

[RuntimeConfigurationUpdateRequestBody](#)

The configuration settings to run telemetry.

where used: body; required: true

```
{  
  "TelemetryConfiguration": {  
    "Telemetry": "On|Off"  
  }  
}
```

schema:

RuntimeConfigurationUpdate

Information about the runtime configuration for a thing.

type: object

TelemetryConfiguration

The configuration settings to run telemetry.

type: object

required: ["Telemetry"]

Telemetry

The configuration setting to turn on or turn off telemetry.

type: string

enum: ["On", "Off"]

[ThingName](#)

The thing name.

where used: path; required: true

type: string

Responses:

200

200 response

[Empty](#)

```
{  
}
```

Empty Schema

Empty

type: object

400

400 response

[GeneralError](#)

```
{  
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

500

500 response

[GeneralError](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

Endpoints

The following endpoints are supported:

- [/greengrass/bulk/deployments](#)

- [/greengrass/bulk/deployments/BulkDeploymentId/\\$stop](#)
- [/greengrass/bulk/deployments/BulkDeploymentId/detailed-reports](#)
- [/greengrass/bulk/deployments/BulkDeploymentId/status](#)
- [/greengrass/definition/connectors](#)
- [/greengrass/definition/connectors/ConnectorDefinitionId](#)
- [/greengrass/definition/connectors/ConnectorDefinitionId/versions](#)
- [/greengrass/definition/connectors/ConnectorDefinitionId/versions/ConnectorDefinitionVersionId](#)
- [/greengrass/definition/cores](#)
- [/greengrass/definition/cores/CoreDefinitionId](#)
- [/greengrass/definition/cores/CoreDefinitionId/versions](#)
- [/greengrass/definition/cores/CoreDefinitionId/versions/CoreDefinitionVersionId](#)
- [/greengrass/definition/devices](#)
- [/greengrass/definition/devices/DeviceDefinitionId](#)
- [/greengrass/definition/devices/DeviceDefinitionId/versions](#)
- [/greengrass/definition/devices/DeviceDefinitionId/versions/DeviceDefinitionVersionId](#)
- [/greengrass/definition/functions](#)
- [/greengrass/definition/functions/FunctionDefinitionId](#)
- [/greengrass/definition/functions/FunctionDefinitionId/versions](#)
- [/greengrass/definition/functions/FunctionDefinitionId/versions/FunctionDefinitionVersionId](#)
- [/greengrass/definition/loggers](#)
- [/greengrass/definition/loggers/LoggerDefinitionId](#)
- [/greengrass/definition/loggers/LoggerDefinitionId/versions](#)
- [/greengrass/definition/loggers/LoggerDefinitionId/versions/LoggerDefinitionVersionId](#)
- [/greengrass/definition/resources](#)
- [/greengrass/definition/resources/ResourceDefinitionId](#)
- [/greengrass/definition/resources/ResourceDefinitionId/versions](#)
- [/greengrass/definition/resources/ResourceDefinitionId/versions/ResourceDefinitionVersionId](#)
- [/greengrass/definition/subscriptions](#)
- [/greengrass/definition/subscriptions/SubscriptionDefinitionId](#)

- [/greengrass/definition/subscriptions/SubscriptionDefinitionId/versions](#)
- [/greengrass/definition/subscriptions/SubscriptionDefinitionId/versions/SubscriptionDefinitionVersionId](#)
- [/greengrass/groups](#)
- [/greengrass/groups/GroupId](#)
- [/greengrass/groups/GroupId/certificateauthorities](#)
- [/greengrass/groups/GroupId/certificateauthorities/configuration/expiry](#)
- [/greengrass/groups/GroupId/certificateauthorities/CertificateAuthorityId](#)
- [/greengrass/groups/GroupId/deployments](#)
- [/greengrass/groups/GroupId/deployments/\\$reset](#)
- [/greengrass/groups/GroupId/deployments/DeploymentId/status](#)
- [/greengrass/groups/GroupId/role](#)
- [/greengrass/groups/GroupId/versions](#)
- [/greengrass/groups/GroupId/versions/GroupVersionId](#)
- [/greengrass/servicerole](#)
- [/greengrass/things/ThingName/connectivityInfo](#)
- [/greengrass/things/ThingName/runtimeconfig](#)
- [/greengrass/updates](#)
- [/tags/resource-arn](#)

/greengrass/bulk/deployments

POST

POST /greengrass/bulk/deployments

Operation ID: [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.

Produces: application/json

Body Parameters

StartBulkDeploymentRequestBody

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 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

Header Parameters

[X-Amzn-Client-Token](#)

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>] \  
  --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](#)

```
{
  "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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

GET

GET /greengrass/bulk/deployments

Operation ID: [ListBulkDeployments](#)

Returns a list of bulk deployments.

Produces: application/json

Query Parameters

[MaxResults](#)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

[NextToken](#)

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-deployments \  
  [--max-results <value>] \  
  [--next-token <value>] \  
  [--cli-input-json <value>] \  
  --cli-input-json <value>
```

```
[--generate-cli-skeleton]
```

cli-input-json format:

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

Responses

200

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

[ListBulkDeploymentsResponse](#)

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

ListBulkDeploymentsResponse

type: object

BulkDeployments

type: array

items: [BulkDeployment](#)

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

```
{
  "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](#)

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/\$stop

PUT

PUT /greengrass/bulk/deployments/*BulkDeploymentId*/\$stop

Operation ID: [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.

Produces: application/json

Path Parameters

BulkDeploymentId

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

```
{
```

```
"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](#)

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](#)

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

The ID of the bulk deployment.

where used: path; required: true

type: string

Query Parameters

MaxResults

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

NextToken

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](#)

```
{
  "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](#)

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](#)

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](#)

```
{
  "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](#)

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/status

GET

GET /greengrass/bulk/deployments/*BulkDeploymentId*/status

Operation ID: [GetBulkDeploymentStatus](#)

Returns the status of a bulk deployment.

Produces: application/json

Path Parameters

BulkDeploymentId

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](#)

```
{
  "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

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

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](#)

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](#)

```
{
  "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](#)

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](#)

Retrieves a list of connector definitions.

Produces: application/json

Query Parameters

[MaxResults](#)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

[NextToken](#)

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>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

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

Responses

200 (ListConnectorDefinitionsResponse)

[ListDefinitionsResponse](#)

```
{
  "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](#)

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](#)

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](#)

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"
  }
}
```

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](#)

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

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

where used: header; required: false

type: string

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"
}
```

Responses

200 (CreateConnectorDefinitionResponse)

DefinitionInformation

```
{
  "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

```
{
  "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](#)

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

GET

GET /greengrass/definition/connectors/*ConnectorDefinitionId*

Operation ID: [GetConnectorDefinition](#)

Retrieves information about a connector definition.

Produces: application/json

Path Parameters

ConnectorDefinitionId

The ID of the connector definition.

where used: path; required: true

type: string

CLI

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

cli-input-json format:

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

Responses

200 (GetConnectorDefinitionResponse)

DefinitionInformation

```
{
  "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

```
{
  "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](#)

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/connectors/*ConnectorDefinitionId*

Operation ID: [DeleteConnectorDefinition](#)

Deletes a connector definition.

Produces: application/json

Path Parameters

[ConnectorDefinitionId](#)

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](#)

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{  
  "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](#)

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](#)

Updates the name of a connector definition. To update the list of available connectors, use `CreateConnectorDefinitionVersion`.

Produces: application/json

Body Parameters

UpdateDefinitionRequestBody

where used: body; required: true

```
{
  "Name": "string"
}
```

Name

The name of the definition.

required: true

type: string

Path Parameters

ConnectorDefinitionId

The ID of the connector definition.

where used: path; required: true

type: string

CLI

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

cli-input-json format:

```
{
  "ConnectorDefinitionId": "string",
  "Name": "string"
}
```

Responses

200

Success.

[Empty](#)

```
{
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{
  "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](#)

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

GET

GET /greengrass/definition/connectors/*ConnectorDefinitionId*/versions

Operation ID: [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.

Produces: application/json

Path Parameters

ConnectorDefinitionId

The ID of the connector definition.

where used: path; required: true

type: string

Query Parameters

NextToken

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

where used: query; required: false

type: string

MaxResults

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

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"
}
```

Responses

200 (ListConnectorDefinitionVersionsResponse)

[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](#)

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

```
{
  "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](#)

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](#)

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

Produces: application/json

Body Parameters

[CreateConnectorDefinitionVersionRequestBody](#)

where used: body; required: true

```
{
  "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](#)

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

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

where used: header; required: false

type: string

Path Parameters

ConnectorDefinitionId

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

```
{  
  "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

```
{
  "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](#)

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](#)

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

The ID of the connector definition.

where used: path; required: true

type: 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.

where used: path; required: true

type: string

Query Parameters

NextToken

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



```
[--generate-cli-skeleton]
```

cli-input-json format:

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

Responses

200 (GetConnectorDefinitionVersionResponse)

GetConnectorDefinitionVersionResponse

```
{
  "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](#)

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

```
{
  "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](#)

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](#)

Retrieves a list of core definitions.

Produces: application/json

Query Parameters

[MaxResults](#)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

[NextToken](#)

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](#)

```
{
  "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](#)

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](#)

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](#)

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"
  }
}
```

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](#)

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

A client token used to correlate requests and responses.

where used: header; required: false

type: string

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"
}
```

Responses

200 (CreateCoreDefinitionResponse)

DefinitionInformation

```
{
  "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](#)

```
{
  "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](#)

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

GET

GET /greengrass/definition/cores/*CoreDefinitionId*

Operation ID: [GetCoreDefinition](#)

Retrieves information about a core definition version.

Produces: application/json

Path Parameters

CoreDefinitionId

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

```
{
  "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

```
{
  "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](#)

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](#)

Deletes a core definition.

Produces: application/json

Path Parameters

[CoreDefinitionId](#)

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](#)

```
{
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{
  "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](#)

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](#)

Updates the name of a core definition. To update the list of available cores, use `CreateCoreDefinitionVersion`.

Produces: application/json

Body Parameters

UpdateDefinitionRequestBody

where used: body; required: true

```
{
  "Name": "string"
}
```

Name

The name of the definition.

required: true

type: string

Path Parameters

CoreDefinitionId

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](#)

```
{
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{
  "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](#)

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](#)

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

Produces: application/json

Body Parameters

CreateCoreDefinitionVersionRequestBody

where used: body; required: true

```
{
  "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](#)

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

Header Parameters

X-Amzn-Client-Token

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Path Parameters

CoreDefinitionId

The ID of the core definition.

where used: path; required: true

type: string

CLI

```
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:

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

Responses

200 (CreateCoreDefinitionVersionResponse)

VersionInformation

```
{  
  "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](#)

```
{
  "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](#)

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](#)

Lists the versions of a core definition.

Produces: application/json

Path Parameters

[CoreDefinitionId](#)

The ID of the core definition.

where used: path; required: true

type: string

Query Parameters

[NextToken](#)

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

where used: query; required: false

type: string

[MaxResults](#)

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](#)

```
{
```

```
"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](#)

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

```
{
  "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](#)

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/ CoreDefinitionVersionId

GET

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

Operation ID: [GetCoreDefinitionVersion](#)

Retrieves information about a core definition version.

Produces: application/json

Path Parameters

[CoreDefinitionId](#)

The ID of the core definition.

where used: path; required: true

type: 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.

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

```
{  
  "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](#)

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](#)

```
{
  "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](#)

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](#)

Retrieves a list of device definitions.

Produces: application/json

Query Parameters

[MaxResults](#)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

[NextToken](#)

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>] \
[--generate-cli-skeleton]
```

cli-input-json format:

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

Responses

200 (ListDeviceDefinitionsResponse)

[ListDefinitionsResponse](#)

```
{
  "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](#)

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/devices

Operation ID: [CreateDeviceDefinition](#)

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](#)

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](#)

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

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

```
{  
  "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](#)

```
{
  "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](#)

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](#)

Retrieves information about a device definition.

Produces: application/json

Path Parameters

DeviceDefinitionId

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

```
{
  "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

```
{
  "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](#)

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](#)

Deletes a device definition.

Produces: application/json

Path Parameters

DeviceDefinitionId

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

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{
  "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](#)

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](#)

Updates the name of a device definition. To update the list of available devices, use `CreateDeviceDefinitionVersion`.

Produces: application/json

Body Parameters

[UpdateDefinitionRequestBody](#)

where used: body; required: true

```
{
  "Name": "string"
}
```

Name

The name of the definition.

required: true

type: string

Path Parameters

[DeviceDefinitionId](#)

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](#)

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{
  "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](#)

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/versions

POST

POST /greengrass/definition/devices/*DeviceDefinitionId*/versions

Operation ID: [CreateDeviceDefinitionVersion](#)

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

Produces: application/json

Body Parameters

[CreateDeviceDefinitionVersionRequestBody](#)

where used: body; required: true

```
{
  "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](#)

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](#)

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Path Parameters

DeviceDefinitionId

The ID of the device definition.

where used: path; required: true

type: string

CLI

```
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:

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

Responses

200 (CreateDeviceDefinitionVersionResponse)

VersionInformation

```
{
```



```
"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

```
{  
  "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](#)

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/devices/*DeviceDefinitionId*/versions

Operation ID: [ListDeviceDefinitionVersions](#)

Lists the versions of a device definition.

Produces: application/json

Path Parameters

[DeviceDefinitionId](#)

The ID of the device definition.

where used: path; required: true

type: string

Query Parameters

[NextToken](#)

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

where used: query; required: false

type: string

[MaxResults](#)

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](#)

```
{  
  "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](#)

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](#)

```
{
  "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](#)

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**/versions/**DeviceDefinitionVersionId**

GET

GET /greengrass/definition/devices/*DeviceDefinitionId*/versions/*DeviceDefinitionVersionId*

Operation ID: [GetDeviceDefinitionVersion](#)

Retrieves information about a device definition version.

Produces: application/json

Path Parameters

DeviceDefinitionId

The ID of the device definition.

where used: path; required: true

type: 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.

where used: path; required: true

type: string

Query Parameters

NextToken

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-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"
}
```

Responses

200 (GetDeviceDefinitionVersionResponse)

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](#)

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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

/greengrass/definition/functions

GET

GET /greengrass/definition/functions

Operation ID: [ListFunctionDefinitions](#)

Retrieves a list of Lambda function definitions.

Produces: application/json

Query Parameters

MaxResults

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

NextToken

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-function-definitions \  
  [--max-results <value>] \  
  [--next-token <value>] \  
  [--cli-input-json <value>] \  
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{  
  "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](#)

```
{
  "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](#)

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/functions

Operation ID: [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`.

Produces: application/json

Body Parameters

[CreateFunctionDefinitionRequestBody](#)

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"
    }
  ],
  "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](#)

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](#)

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

Header Parameters

X-Amzn-Client-Token

A client token used to correlate requests and responses.

where used: header; required: false

type: string

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:

```
{  
  "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](#)

```
{
  "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

```
{
  "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](#)

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](#)

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

Produces: application/json

Path Parameters

FunctionDefinitionId

The ID of the function definition.

where used: path; required: true

type: string

CLI

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

cli-input-json format:

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

Responses

200 (GetFunctionDefinitionResponse)

Success.

DefinitionInformation

```
{  
  "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](#)

```
{
  "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](#)

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/functions/*FunctionDefinitionId*

Operation ID: [DeleteFunctionDefinition](#)

Deletes a Lambda function definition.

Produces: application/json

Path Parameters**FunctionDefinitionId**

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](#)

```
{
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{
  "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](#)

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](#)

Updates the name of a function definition. To update the list of available functions, use `CreateFunctionDefinitionVersion`.

Produces: application/json

Body Parameters

UpdateDefinitionRequestBody

where used: body; required: true

```
{
  "Name": "string"
}
```

Name

The name of the definition.

required: true

type: string

Path Parameters

FunctionDefinitionId

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](#)

```
{
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{
  "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](#)

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/versions

POST

POST /greengrass/definition/functions/*FunctionDefinitionId*/versions

Operation ID: [CreateFunctionDefinitionVersion](#)

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

Produces: application/json

Body Parameters

CreateFunctionDefinitionVersionRequestBody

Information about the function definition version.

where used: body; required: true

```
{
  "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](#)

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](#)

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

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Path Parameters

FunctionDefinitionId

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:

```
{
  "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"
          }
        }
      }
    }
  ]
}
```

```
    }
  }
},
"AmznClientToken": "string"
}
```

Responses

200 (CreateFunctionDefinitionVersionResponse)

VersionInformation

```
{
  "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](#)

```
{
  "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](#)

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/functions/*FunctionDefinitionId*/versions

Operation ID: [ListFunctionDefinitionVersions](#)

Lists the versions of a Lambda function definition.

Produces: application/json

Path Parameters

FunctionDefinitionId

The ID of the function definition.

where used: path; required: true

type: string

Query Parameters

NextToken

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

where used: query; required: false

type: string

MaxResults

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

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"
}
```

Responses

200 (ListFunctionDefinitionVersionsResponse)

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](#)

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](#)

```
{
  "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](#)

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/versions/FunctionDefinitionVersionId

GET

GET /greengrass/definition/functions/*FunctionDefinitionId*/versions/*FunctionDefinitionVersionId*

Operation ID: [GetFunctionDefinitionVersion](#)

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

The ID of the function definition.

where used: path; required: true

type: 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.

where used: path; required: true

type: string

Query Parameters

NextToken

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-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:

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

Responses

200

Success.

GetFunctionDefinitionVersionResponse

```
{
  "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  
      }  
    },  
    "FunctionRuntimeOverride": "string"  
  }  
]  
,  
"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](#)

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](#)

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

FunctionRuntimeOverride

The Lambda runtime supported by Greengrass which is to be used instead of the one specified in the Lambda function.

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](#)

```
{
  "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](#)

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](#)

Retrieves a list of logger definitions.

Produces: application/json

Query Parameters

[MaxResults](#)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

[NextToken](#)

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](#)

```
{
  "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](#)

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](#)

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](#)

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](#)

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

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 <value>] \  
[--generate-cli-skeleton]
```

cli-input-json format:

```
{  
  "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

```
{  
  "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](#)

```
{
  "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](#)

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

GET

GET /greengrass/definition/loggers/*LoggerDefinitionId*

Operation ID: [GetLoggerDefinition](#)

Retrieves information about a logger definition.

Produces: application/json

Path Parameters

LoggerDefinitionId

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:

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

Responses

200 (GetLoggerDefinitionResponse)

DefinitionInformation

```
{
  "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

```
{
  "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](#)

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](#)

Deletes a logger definition.

Produces: application/json

Path Parameters

[LoggerDefinitionId](#)

The ID of the logger definition.

where used: path; required: true

type: string

CLI

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

cli-input-json format:

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

Responses

200

Success.

[Empty](#)

```
{
```

```
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{
  "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](#)

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](#)

Updates the name of a logger definition. To update the list of available loggers, use `CreateLoggerDefinitionVersion`.

Produces: application/json

Body Parameters

[UpdateDefinitionRequestBody](#)

where used: body; required: true

```
{
  "Name": "string"
}
```

Name

The name of the definition.

required: true

type: string

Path Parameters

LoggerDefinitionId

The ID of the logger definition.

where used: path; required: true

type: string

CLI

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

cli-input-json format:

```
{  
  "LoggerDefinitionId": "string",  
  "Name": "string"  
}
```

Responses

200

Success.

Empty

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{
  "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](#)

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](#)

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

Produces: application/json

Body Parameters

[CreateLoggerDefinitionVersionRequestBody](#)

where used: body; required: true

```
{
  "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](#)

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

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Path Parameters

LoggerDefinitionId

The ID of the logger definition.

where used: path; required: true

type: string

CLI

```
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:

```
{  
  "LoggerDefinitionId": "string",  
  "Loggers": [  
    {
```

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

Responses

200 (CreateLoggerDefinitionVersionResponse)

VersionInformation

```
{
  "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

```
{
  "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](#)

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](#)

Lists the versions of a logger definition.

Produces: application/json

Path Parameters

LoggerDefinitionId

The ID of the logger definition.

where used: path; required: true

type: string

Query Parameters

NextToken

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

where used: query; required: false

type: string

MaxResults

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>] \  
  [--generate-cli-skeleton]
```

cli-input-json format:

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

Responses

200 (ListLoggerDefinitionVersionsResponse)

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](#)

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](#)

```
{
  "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](#)

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](#)

Retrieves information about a logger definition version.

Produces: application/json

Path Parameters

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.

where used: path; required: true

type: string

LoggerDefinitionId

The ID of the logger definition.

where used: path; required: true

type: string

Query Parameters

NextToken

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]
```

cli-input-json format:

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

Responses

200 (GetLoggerDefinitionVersionResponse)

Success.

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](#)

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](#)

```
{
  "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](#)

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](#)

Retrieves a list of resource definitions.

Produces: application/json

Query Parameters

[MaxResults](#)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

[NextToken](#)

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-resource-definitions \  
  [--max-results <value>] \  
  [--next-token <value>] \  
  [--cli-input-json <value>] \  
  [--generate-cli-skeleton]
```

cli-input-json format:

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

Responses

200 (ListResourceDefinitionsResponse)

The IDs of all the Greengrass resource definitions in this account.

[ListDefinitionsResponse](#)

```
{  
  "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](#)

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](#)

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

where used: body; required: true

```
{
  "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](#)

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

tags

The resource tags.

type: object

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

Header Parameters

X-Amzn-Client-Token

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>] \  
  [--tags <value>] \  
  [--amzn-client-token <value>] \  
  [--cli-input-json <value>] \  
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "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](#)

```
{
  "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

```
{
  "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](#)

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](#)

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

Produces: application/json

Path Parameters

ResourceDefinitionId

The ID of the resource definition.

where used: path; required: true

type: string

CLI

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

cli-input-json format:

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

Responses

200 (GetResourceDefinitionResponse)

Success.

[DefinitionInformation](#)


```
{
  "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](#)

```
{
  "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](#)

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](#)

Deletes a resource definition.

Produces: application/json

Path Parameters

[ResourceDefinitionId](#)

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:

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

Responses

200

Success.

[Empty](#)

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{
```

```
"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](#)

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/resources/*ResourceDefinitionId*

Operation ID: [UpdateResourceDefinition](#)

Updates the name of a resource definition. To update the list of available resources, use `CreateResourceDefinitionVersion`.

Produces: application/json

Body Parameters

[UpdateDefinitionRequestBody](#)

where used: body; required: true

```
{
  "Name": "string"
}
```

Name

The name of the definition.

required: true

type: string

Path Parameters

[ResourceDefinitionId](#)

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 <value>] \  
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{  
  "ResourceDefinitionId": "string",  
  "Name": "string"  
}
```

Responses

200

Success.

[Empty](#)

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{  
  "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](#)

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](#)

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

Produces: application/json

Body Parameters

[CreateResourceDefinitionVersionRequestBody](#)

Information about the resource definition version.

where used: body; required: true

```
{
  "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"
    ]
  }
}
]
}

```

ResourceDefinitionVersion

Information about a resource definition version.

type: object

Resources

A list of resources.

type: array

items: [Resource](#)

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

Header Parameters

X-Amzn-Client-Token

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Path Parameters

ResourceDefinitionId

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](#)

```
{
  "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

```
{
  "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](#)

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](#)

Lists the versions of a resource definition.

Produces: application/json

Path Parameters

[ResourceDefinitionId](#)

The ID of the resource definition.

where used: path; required: true

type: string

Query Parameters

NextToken

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

where used: query; required: false

type: string

MaxResults

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

CLI

```
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:

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

Responses

200 (ListResourceDefinitionVersionsResponse)

[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](#)

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](#)

```
{
  "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](#)

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/ ResourceDefinitionVersionId

GET

GET /greengrass/definition/resources/*ResourceDefinitionId*/
versions/*ResourceDefinitionVersionId*

Operation ID: [GetResourceDefinitionVersion](#)

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

Produces: application/json

Path Parameters

ResourceDefinitionId

The ID of the resource definition.

where used: path; required: true

type: 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.

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

```
{
  "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](#)

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

400

Invalid request.

[GeneralError](#)

```
{
```

```
"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](#)

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](#)

Retrieves a list of subscription definitions.

Produces: application/json

Query Parameters

[MaxResults](#)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

[NextToken](#)

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:

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

Responses

200 (ListSubscriptionDefinitionsResponse)

ListDefinitionsResponse

```
{
  "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](#)

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](#)

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](#)

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](#)

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

A client token used to correlate requests and responses.

where used: header; required: false

type: string

CLI

```
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:

```
{  
  "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

```
{  
  "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](#)

```
{
  "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](#)

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

GET

GET /greengrass/definition/subscriptions/*SubscriptionDefinitionId*

Operation ID: [GetSubscriptionDefinition](#)

Retrieves information about a subscription definition.

Produces: application/json

Path Parameters

SubscriptionDefinitionId

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

```
{
  "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

```
{
  "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](#)

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/subscriptions/*SubscriptionDefinitionId*

Operation ID: [DeleteSubscriptionDefinition](#)

Deletes a subscription definition.

Produces: application/json

Path Parameters

SubscriptionDefinitionId

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

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{
  "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](#)

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](#)

Updates the name of a subscription definition. To update the list of available subscriptions, use `CreateSubscriptionDefinitionVersion`.

Produces: application/json

Body Parameters

[UpdateDefinitionRequestBody](#)

where used: body; required: true

```
{
  "Name": "string"
}
```

Name

The name of the definition.

required: true

type: string

Path Parameters

[SubscriptionDefinitionId](#)

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](#)

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{
```

```
"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](#)

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

POST

POST /greengrass/definition/subscriptions/*SubscriptionDefinitionId*/versions

Operation ID: [CreateSubscriptionDefinitionVersion](#)

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

Produces: application/json

Body Parameters

[CreateSubscriptionDefinitionVersionRequestBody](#)

where used: body; required: true

```
{
  "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](#)

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

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Path Parameters

SubscriptionDefinitionId

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:

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

Responses

200 (CreateSubscriptionDefinitionVersionResponse)

[VersionInformation](#)

```
{
  "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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string


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

cli-input-json format:

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

Responses

200 (ListSubscriptionDefinitionVersionsResponse)

[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](#)

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](#)

```
{
  "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](#)

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](#)

Retrieves information about a subscription definition version.

Produces: application/json

Path Parameters

SubscriptionDefinitionId

The ID of the subscription definition.

where used: path; required: true

type: 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.

where used: path; required: true

type: string

Query Parameters

NextToken

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-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:

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

Responses

200 (GetSubscriptionDefinitionVersionResponse)

GetSubscriptionDefinitionVersionResponse

```
{  
  "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](#)

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](#)

```
{
  "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](#)

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](#)

Retrieves a list of groups.

Produces: application/json

Query Parameters

[MaxResults](#)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

[NextToken](#)

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 (ListGroupResponse)

[ListGroupResponse](#)

```
{  
  "Groups": [  
    {  
      "Name": "string",  
      "Id": "string",  
      "Arn": "string",  
      "LastUpdatedTimestamp": "string",  
      "CreationTimestamp": "string",  
      "LatestVersion": "string",  
      "LatestVersionArn": "string"  
    }  
  ],  
  "NextToken": "string"  
}
```

ListGroupResponse

type: object

Groups

Information about a group.

type: array

items: [GroupInformation](#)

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](#)

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/aws-labs/aws-greengrass-group-setup>) as a library or command-line application to create and deploy Greengrass groups.

Produces: application/json

Body Parameters

[CreateGroupRequestBody](#)

where used: body; required: true

```
{
  "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.

type: string

DeviceDefinitionVersionArn

The ARN of the client 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

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 <value>] \  
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{  
  "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](#)

```
{
  "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

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

/greengrass/groups/GroupId

GET

GET /greengrass/groups/*GroupId*

Operation ID: [GetGroup](#)

Retrieves information about a group.

Produces: application/json

Path Parameters

[GroupId](#)

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](#)

```
{
  "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

```
{
  "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](#)

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](#)

Deletes a group.

Produces: application/json

Path Parameters

[GroupId](#)

The ID of the Greengrass group.

where used: path; required: true

type: string

CLI

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

cli-input-json format:

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

Responses

200

Success.

[Empty](#)

```
{  
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{  
  "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](#)

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](#)

Updates the name of a group. To update group components, use `CreateGroupVersion`.

Produces: `application/json`

Body Parameters

[UpdateDefinitionRequestBody](#)

where used: body; required: true

```
{
  "Name": "string"
}
```

Name

The name of the definition.

required: true

type: string

Path Parameters

[GroupId](#)

The ID of the Greengrass group.

where used: path; required: true

type: string

CLI

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


cli-input-json format:

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

Responses

200

Success.

[Empty](#)

```
{
}
```

Empty Schema

Empty

type: object

400

Invalid request.

[GeneralError](#)

```
{
  "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](#)

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](#)

Retrieves the current CAs for a group.

Produces: application/json

Path Parameters

GroupId

The ID of the Greengrass group.

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

```
{
  "GroupCertificateAuthorities": [
    {
      "GroupCertificateAuthorityId": "string",
      "GroupCertificateAuthorityArn": "string"
    }
  ]
}
```

ListGroupCertificateAuthoritiesResponse

type: object

GroupCertificateAuthorities

A list of certificate authorities associated with the group.

type: array

items: [GroupCertificateAuthorityProperties](#)

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](#)

```
{
  "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](#)

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](#)

```
{
```

```
"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](#)

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](#)

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

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Path Parameters

GroupId

The ID of the Greengrass group.

where used: path; required: true

type: string

CLI

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

cli-input-json format:

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

Responses

200

Success. The response body contains the new, active CA ARN.

[CreateGroupCertificateAuthorityResponse](#)

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

CreateGroupCertificateAuthorityResponse

type: object

GroupCertificateAuthorityArn

The ARN of the group certificate authority.

type: string

400

Invalid request.

[GeneralError](#)

```
{
  "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](#)

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](#)

```
{  
  "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](#)

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](#)

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

Produces: application/json

Path Parameters

GroupId

The ID of the Greengrass group.

where used: path; required: true

type: string

CLI

```
aws greengrass get-group-certificate-configuration \  
  --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.

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

400

Invalid request.

GeneralError

```
{
  "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](#)

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](#)

```
{
  "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](#)

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*/certificateauthorities/configuration/expiry

Operation ID: [UpdateGroupCertificateConfiguration](#)

Updates the certificate expiry time for a group.

Produces: application/json

Body Parameters

[UpdateGroupCertificateConfigurationRequestBody](#)

where used: body; required: true

```
{
  "CertificateExpiryInMilliseconds": "string"
}
```

updateGroupCertificateConfigurationRequest

type: object

required: ["CertificateExpiryInMilliseconds"]

CertificateExpiryInMilliseconds

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

type: string

Path Parameters

[GroupId](#)

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](#)

```
{
  "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

```
{
  "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](#)

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](#)

```
{
  "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](#)

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](#)

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

Produces: application/json

Path Parameters

GroupId

The ID of the Greengrass group.

where used: path; required: true

type: string

CertificateAuthorityId

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

```
{
  "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

```
{
  "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](#)

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](#)

```
{  
  "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](#)

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

POST

POST /greengrass/groups/*GroupId*/deployments

Operation ID: [CreateDeployment](#)

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](#)

where used: body; required: true

```
{
  "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

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Path Parameters

GroupId

The ID of the Greengrass group.

where used: path; required: true

type: string

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"
}
```

Responses

200

Success. The group was deployed.

[CreateDeploymentResponse](#)

```
{
  "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](#)

```
{
  "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](#)

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](#)

Returns a history of deployments for the group.

Produces: application/json

Path Parameters

[GroupId](#)

The ID of the Greengrass group.

where used: path; required: true

type: string

Query Parameters

[MaxResults](#)

The maximum number of results to be returned per request.

where used: query; required: false

type: integer

[NextToken](#)

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-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"  
}
```

Responses

200

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

[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](#)

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](#)

```
{
  "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](#)

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/\$reset

POST

POST /greengrass/groups/*GroupId*/deployments/\$reset

Operation ID: [ResetDeployments](#)

Resets a group's deployments.

Produces: application/json

Body Parameters

[ResetDeploymentsRequestBody](#)

Information required to reset deployments.

where used: body; required: true

```
{
  "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

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Path Parameters

GroupId

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:

```
{  
  "GroupId": "string",  
  "Force": "boolean",  
  "AmznClientToken": "string"  
}
```

Responses

200

Success. The group's deployments were reset.

[ResetDeploymentsResponse](#)

```
{
  "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](#)

```
{
  "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](#)

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](#)

Returns the status of a deployment.

Produces: application/json

Path Parameters

GroupId

The ID of the Greengrass group.

where used: path; required: true

type: string

DeploymentId

The ID of the deployment.

where used: path; required: true

type: string

CLI

```
aws greengrass get-deployment-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

```
{  
  "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

ErrorDetails

A list of error details.

type: array

items: [ErrorDetail](#)

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](#)

```
{
  "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](#)

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](#)

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

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

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](#)

```
{
  "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](#)

```
{
  "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](#)

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](#)

```
{  
  "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](#)

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*/role

Operation ID: [DisassociateRoleFromGroup](#)

Disassociates the role from a group.

Path Parameters

[GroupId](#)

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](#)

```
{  
  "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](#)

```
{
  "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](#)

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](#)

```
{
  "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](#)

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](#)

Retrieves the role associated with a group.

Path Parameters

GroupId

The ID of the Greengrass group.

where used: path; required: true

type: string

CLI

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

cli-input-json format:

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

Responses

200

Success.

[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

400

Invalid request.

[GeneralError](#)

```
{
  "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](#)

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

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

/greengrass/groups/GroupId/versions

POST

POST /greengrass/groups/*GroupId*/versions

Operation ID: [CreateGroupVersion](#)

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

Produces: application/json

Body Parameters

[CreateGroupVersionRequestBody](#)

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 client 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

A client token used to correlate requests and responses.

where used: header; required: false

type: string

Path Parameters

GroupId

The ID of the Greengrass group.

where used: path; required: true

type: string

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

Responses

200 (CreateGroupVersionResponse)

Success. The response contains information about the group version.

VersionInformation

```
{
  "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

```
{
  "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](#)

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](#)

Lists the versions of a group.

Produces: application/json

Path Parameters

GroupId

The ID of the Greengrass group.

where used: path; required: true

type: string

Query Parameters

NextToken

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

where used: query; required: false

type: string

MaxResults

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:

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

Responses

200 (ListGroupVersionsResponse)

Success. The response contains the list of versions and metadata for the given group.

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](#)

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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

/greengrass/groups/GroupId/versions/GroupVersionId

GET

GET /greengrass/groups/*GroupId*/versions/*GroupVersionId*

Operation ID: [GetGroupVersion](#)

Retrieves information about a group version.

Produces: application/json

Path Parameters

GroupId

The ID of the Greengrass group.

where used: path; required: true

type: 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.

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:

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

Responses

200

Success.

[GetGroupVersionResponse](#)

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

```
"ConnectorDefinitionVersionArn": "string"
}
```

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 client 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

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

/greengrass/servicerole

GET

GET /greengrass/servicerole

Operation ID: [GetServiceRoleForAccount](#)

Produces: application/json

CLI

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

Responses

200

Success.

[GetServiceRoleForAccountResponse](#)

```
{
  "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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

PUT

PUT /greengrass/servicerole

Operation ID: [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`.

Body Parameters

[AssociateServiceRoleToAccountRequestBody](#)

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

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

cli-input-json format:

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

Responses

200

Success.

[AssociateServiceRoleToAccountResponse](#)

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

AssociateServiceRoleToAccountResponse

type: object

AssociatedAt

The time when the service role was associated with the account.

type: string

400

Invalid request.

[GeneralError](#)

```
{
  "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](#)

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](#)

```
{  
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

DELETE

DELETE /greengrass/servicerole

Operation ID: [DisassociateServiceRoleFromAccount](#)

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

CLI

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

Responses

200

Success.

[DisassociateServiceRoleFromAccountResponse](#)

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

DisassociateServiceRoleFromAccountResponse

type: object

DisassociatedAt

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

type: string

500

Server error.

[GeneralError](#)

```
{
```

```
"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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

GET

GET /greengrass/servicerole

Operation ID: [GetServiceRoleForAccount](#)

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](#)

```
{
  "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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

/greengrass/things/ThingName/connectivityInfo

GET

GET /greengrass/things/*ThingName*/connectivityInfo

Operation ID: [GetConnectivityInfo](#)

Retrieves the connectivity information for a core.

Produces: application/json

Path Parameters

ThingName

The thing name.

where used: path; required: true

type: string

CLI

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

cli-input-json format:

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

Responses

200

Success.

[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](#)

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](#)

```
{
  "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](#)

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](#)

```
{
  "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](#)

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](#)

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

The connectivity information.

where used: body; required:

```
{
  "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](#)

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

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](#)

```
{  
  "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](#)

```
{  
  "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](#)

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](#)

```
{  
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

/greengrass/things/ThingName/runtimeconfig

GET

GET /greengrass/things/{ThingName}/runtimeconfig

Operation ID: [GetThingRuntimeConfiguration](#)

Produces: application/json

CLI

```
aws greengrass get-thing-runtime-configuration \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

Responses

200

200 response

[GetThingRuntimeConfigurationResponse](#)

```
{
  "RuntimeConfiguration": {
    "TelemetryConfiguration": {
      "Telemetry": "On|Off",
      "ConfigurationSyncStatus": "InSync|OutOfSync"
    }
  }
}
```

GetThingRuntimeConfigurationResponse

The runtime configuration for a thing.

type: object

RuntimeConfiguration

Runtime configuration for a thing.

type: object

TelemetryConfiguration

The configuration setting for running telemetry.

type: object

required: ["Telemetry"]

Telemetry

The configuration setting to turn on or turn off telemetry.

type: string

enum: ["On", "Off"]

ConfigurationSyncStatus

The synchronization status of the device-reported configuration with the desired configuration.

type: string

enum: ["InSync", "OutOfSync"]

400

400 response

[GeneralError](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

500

500 response

[GeneralError](#)

```
{
  "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](#)

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}/runtimeconfig

Operation ID: [UpdateThingRuntimeConfiguration](#)

Produces: application/json

CLI

```
aws greengrass update-thing-runtime-configuration \  
  [--telemetry-configuration <value>] \  
  [--cli-input-json <value>] \  
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{  
  "TelemetryConfiguration": {  
    "Telemetry": "On|Off"  
  }  
}
```

Responses

200

200 response

[Empty](#)

```
{  
}
```

Empty Schema

Empty

type: object

400

400 response

[GeneralError](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

500

500 response

[GeneralError](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

/greengrass/updates

POST

POST /greengrass/updates

Operation ID: [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.

Produces: application/json

Body Parameters

[CreateSoftwareUpdateJobRequestBody](#)

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

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"]

Header Parameters

X-Amzn-Client-Token

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](#)

```
{
  "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](#)

```
{
  "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](#)

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](#)

```
{  
  "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](#)

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](#)

Lists tags for a Greengrass resource. Valid resources are Group, ConnectorDefinition, CoreDefinition, DeviceDefinition, FunctionDefinition, LoggerDefinition, ResourceDefinition, SubscriptionDefinition, and BulkDeployment.

Produces: application/json

CLI

```
aws greengrass list-tags-for-resource \
  --resource-arn <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

cli-input-json format:

```
{
  "ResourceArn": "string"
}
```

Parameters:

[ResourceArn](#)

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](#)

```
{
  "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](#)

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

POST

POST /tags/*resource-arn*

Operation ID: [TagResource](#)

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

The Amazon Resource Name (ARN) of the resource to attach the tags to.

where used: path; required: true

type: string

tags

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

```
{
```

```
"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](#)

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](#)

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](#)

An array of tag keys to delete.

where used: query; required: true

type: array of strings

[ResourceArn](#)

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

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

Parameters

The following parameters are used:

- [AssociateRoleToGroupRequestBody](#)
- [AssociateServiceRoleToAccountRequestBody](#)
- [BulkDeploymentId](#)
- [CertificateAuthorityId](#)
- [ConnectorDefinitionId](#)
- [ConnectorDefinitionVersionId](#)
- [CoreDefinitionId](#)
- [CoreDefinitionVersionId](#)
- [CreateConnectorDefinitionRequestBody](#)
- [CreateConnectorDefinitionVersionRequestBody](#)
- [CreateCoreDefinitionRequestBody](#)
- [CreateCoreDefinitionVersionRequestBody](#)
- [CreateDeploymentRequestBody](#)
- [CreateDeviceDefinitionRequestBody](#)

- [CreateDeviceDefinitionVersionRequestBody](#)
- [CreateFunctionDefinitionRequestBody](#)
- [CreateFunctionDefinitionVersionRequestBody](#)
- [CreateGroupRequestBody](#)
- [CreateGroupVersionRequestBody](#)
- [CreateLoggerDefinitionRequestBody](#)
- [CreateLoggerDefinitionVersionRequestBody](#)
- [CreateResourceDefinitionRequestBody](#)
- [CreateResourceDefinitionVersionRequestBody](#)
- [CreateSoftwareUpdateJobRequestBody](#)
- [CreateSubscriptionDefinitionRequestBody](#)
- [CreateSubscriptionDefinitionVersionRequestBody](#)
- [DeploymentId](#)
- [DeviceDefinitionId](#)
- [DeviceDefinitionVersionId](#)
- [FunctionDefinitionId](#)
- [FunctionDefinitionVersionId](#)
- [GroupId](#)
- [GroupVersionId](#)
- [LoggerDefinitionId](#)
- [LoggerDefinitionVersionId](#)
- [MaxResults](#)
- [NextToken](#)
- [ResetDeploymentsRequestBody](#)
- [ResourceArn](#)
- [ResourceDefinitionId](#)
- [ResourceDefinitionVersionId](#)
- [RuntimeConfigurationUpdateRequestBody](#)
- [StartBulkDeploymentRequestBody](#)
- [SubscriptionDefinitionId](#)

- [SubscriptionDefinitionVersionId](#)
- [tags](#)
- [ThingName](#)
- [TagKeys](#)
- [UpdateConnectivityInfoRequestBody](#)
- [UpdateDefinitionRequestBody](#)
- [UpdateGroupCertificateConfigurationRequestBody](#)
- [X-Amzn-Client-Token](#)

AssociateRoleToGroupRequestBody

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

AssociateRoleToGroupRequestBody

in: body

required: true

schema: [AssociateRoleToGroupRequest](#)

AssociateRoleToGroupRequest

type: object

required: ["RoleArn"]

RoleArn

The ARN of the role to associate with this group.

type: string

AssociateServiceRoleToAccountRequestBody

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

```
}
```

AssociateServiceRoleToAccountRequestBody

in: body

required: true

schema: [AssociateServiceRoleToAccountRequest](#)

AssociateServiceRoleToAccountRequest

type: object

required: ["RoleArn"]

RoleArn

The ARN of the service role to associate with your account.

type: string

BulkDeploymentId

```
{  
  "BulkDeploymentId": "string"  
}
```

BulkDeploymentId

The ID of the bulk deployment.

in: path

required: true

type: string

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

type: string

CoreDefinitionId

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

CoreDefinitionId

The ID of the core definition.

in: path

required: true

type: string

CoreDefinitionVersionId

```
{
  "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

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

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

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

CreateConnectorDefinitionVersionRequestBody

in: body

required: true

schema: [ConnectorDefinitionVersion](#)

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](#)

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.

CreateCoreDefinitionRequestBody

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

```
"additionalProperty2": "string"
}
```

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

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

CreateCoreDefinitionVersionRequestBody

in: body

required: true

schema: [CoreDefinitionVersion](#)

CoreDefinitionVersion

Information about a core definition version.

type: object

Cores

A list of cores in the core definition version.

type: array

items: [Core](#)

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

CreateDeploymentRequestBody

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

CreateDeploymentRequestBody

in: body

required: true

schema: [CreateDeploymentRequest](#)

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

```
{
  "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](#)

DeviceDefinitionVersion

Information about a device definition version.

type: object

Devices

A list of devices in the definition version.

type: array

items: [Device](#)

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

CreateFunctionDefinitionRequestBody

```
{
  "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
        }
      }
    },
    "FunctionRuntimeOverride": "string"
  }
}
]
```

CreateFunctionDefinitionRequestBody

in: body

required: true

properties

Name: The name of the function definition. Type: string

InitialVersion: Information about the initial version of the function definition. Type: **FunctionDefinitionVersion**

tags: The tags to attach to the new resource. Type: **tags**

CreateFunctionDefinitionVersionRequestBody

```
{
  "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
        }
    },
    "FunctionRuntimeOverride": "string"
}
]
```

CreateFunctionDefinitionVersionRequestBody

Information about the function definition version.

in: body

required: true

schema: [FunctionDefinitionVersion](#)

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](#)

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](#)

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

FunctionRuntimeOverride

The Lambda runtime supported by Greengrass which is to be used instead of the one specified in the Lambda function.

type: string

CreateGroupRequestBody

```
{
  "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

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

CreateGroupVersionRequestBody

in: body

required: true

schema: [GroupVersion](#)

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 client 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

CreateLoggerDefinitionRequestBody

```
{
  "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"
  }
}
```

CreateLoggerDefinitionRequestBody

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

```
{
  "Loggers": [
```

```
{
  "Id": "string",
  "Type": "FileSystem|AWSCloudWatch",
  "Component": "GreengrassSystem|Lambda",
  "Level": "DEBUG|INFO|WARN|ERROR|FATAL",
  "Space": 0
}
```

CreateLoggerDefinitionVersionRequestBody

in: body

required: true

schema: [LoggerDefinitionVersion](#)

LoggerDefinitionVersion

Information about a logger definition version.

type: object

Loggers

A list of loggers.

type: array

items: [Logger](#)

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

CreateResourceDefinitionRequestBody

```
{
  "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": {
  "additionalProperty0": "string",
  "additionalProperty1": "string",
  "additionalProperty2": "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": {
            "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"
      ]
    }
  ]
}
```

CreateResourceDefinitionVersionRequestBody

Information about the resource definition version.

in: body

required: true

schema: [ResourceDefinitionVersion](#)

ResourceDefinitionVersion

Information about a resource definition version.

type: object

Resources

A list of resources.

type: array

items: [Resource](#)

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

CreateSoftwareUpdateJobRequestBody

```
{
  "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](#)

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"]
```

CreateSubscriptionDefinitionRequestBody

```
{
  "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

```
{
```

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

CreateSubscriptionDefinitionVersionRequestBody

in: body

required: true

schema: [SubscriptionDefinitionVersion](#)

SubscriptionDefinitionVersion

Information about a subscription definition version.

type: object

Subscriptions

A list of subscriptions.

type: array

items: [Subscription](#)

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

```
{
  "DeploymentId": "string"
}
```

DeploymentId

The ID of the deployment.

in: path

required: true

type: string

DeviceDefinitionId

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

DeviceDefinitionId

The ID of the device definition.

in: path

required: true

type: string

DeviceDefinitionVersionId

```
{  
  "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

type: string

FunctionDefinitionId

```
{
```



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

FunctionDefinitionId

The ID of the function definition.

in: path

required: true

type: string

FunctionDefinitionVersionId

```
{
  "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

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

GroupId

The ID of the Greengrass group.

in: path

required: true

type: string

GroupVersionId

```
{  
  "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

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

LoggerDefinitionId

The ID of the logger definition.

in: path

required: true

type: string

LoggerDefinitionVersionId

```
{  
  "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

```
{  
  "MaxResults": 0  
}
```

MaxResults

The maximum number of results to be returned per request.

in: query

required: false

min: 1

type: integer

NextToken

```
{  
  "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

```
{  
  "Force": true  
}
```

ResetDeploymentsRequestBody

Information required to reset deployments.

in: body

required: true

schema: [ResetDeploymentsRequest](#)

ResetDeploymentsRequest

Information about a group reset request.

type: object

Force

If true, performs a best-effort only core reset.

type: boolean

ResourceArn

```
{  
  "ResourceArn": "string"  
}
```

ResourceArn

The Amazon Resource Name (ARN) of the target resource for tagging operations.

in: path

required: true

type: string

ResourceDefinitionId

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

ResourceDefinitionId

The ID of the resource definition.

in: path

required: true

type: string

ResourceDefinitionVersionId

```
{  
  "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

required: true

type: string

RuntimeConfigurationUpdateRequestBody

```
{
  "TelemetryConfiguration": {
    "Telemetry": "On|Off"
  }
}
```

RuntimeConfigurationUpdateRequestBody

Information about the runtime configuration for a thing.

type: object

[RuntimeConfigurationUpdate](#)

Runtime configuration for a thing.

type: object

TelemetryConfiguration

The configuration settings to run telemetry.

type: object

required: ["Telemetry"]

Telemetry

The configuration setting to turn on or turn off telemetry.

type: string

enum: ["On", "Off"]

StartBulkDeploymentRequestBody

```
{
  "InputFileUri": "string",
  "ExecutionRoleArn": "string",
  "tags": {
    "additionalProperty0": "string",
    "additionalProperty1": "string",
    "additionalProperty2": "string"
  }
}
```

StartBulkDeploymentRequestBody

in: body

required: true

schema: [StartBulkDeploymentRequest](#)

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.

```
{"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

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

SubscriptionDefinitionId

The ID of the subscription definition.

in: path

required: true

type: string

SubscriptionDefinitionVersionId

```
{  
  "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

```
{  
  "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"
    }
  ]
}
```

UpdateConnectivityInfoRequestBody

The connectivity information.

in: body

schema: [UpdateConnectivityInfoRequest](#)

Information required to update a Greengrass core's connectivity.

ConnectivityInfo

A list of connectivity info.

type: array

items: [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

UpdateDefinitionRequestBody

```
{
  "Name": "string"
}
```

UpdateDefinitionRequestBody

in: body

required: true

properties

Name: The name of the definition. Required: true; Type: string

UpdateGroupCertificateConfigurationRequestBody

```
{
  "CertificateExpiryInMilliseconds": "string"
}
```

UpdateGroupCertificateConfigurationRequestBody

in: body

required: true

schema: [UpdateGroupCertificateConfigurationRequest](#)

updateGroupCertificateConfigurationRequest

type: object

required: ["CertificateExpiryInMilliseconds"]

CertificateExpiryInMilliseconds

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

type: string

X-Amzn-Client-Token

```
{
  "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](#)
- [AssociateServiceRoleToAccountRequest](#)
- [AssociateServiceRoleToAccountResponse](#)
- [BulkDeployment](#)
- [BulkDeploymentMetrics](#)
- [BulkDeploymentResult](#)

- [BulkDeploymentResults](#)
- [BulkDeploymentStatus](#)
- [BulkDeployments](#)
- [ConnectivityInfo](#)
- [Connector](#)
- [ConnectorDefinitionVersion](#)
- [Core](#)
- [CoreDefinitionVersion](#)
- [CreateDeploymentRequest](#)
- [CreateDeploymentResponse](#)
- [CreateGroupCertificateAuthorityResponse](#)
- [CreateSoftwareUpdateJobRequest](#)
- [CreateSoftwareUpdateJobResponse](#)
- [DefinitionInformation](#)
- [Deployment](#)
- [DeploymentType](#)
- [Deployments](#)
- [Device](#)
- [DeviceDefinitionVersion](#)
- [DisassociateRoleFromGroupResponse](#)
- [DisassociateServiceRoleFromAccountResponse](#)
- [Empty Schema](#)
- [ErrorDetail](#)
- [ErrorDetails](#)
- [Function](#)
- [FunctionConfiguration](#)
- [FunctionConfigurationEnvironment](#)
- [FunctionDefaultConfig](#)
- [FunctionDefaultExecutionConfig](#)
- [FunctionDefinitionVersion](#)

- [FunctionExecutionConfig](#)
- [FunctionIsolationMode](#)
- [FunctionRunAsConfig](#)
- [GeneralError](#)
- [GetAssociatedRoleResponse](#)
- [GetBulkDeploymentStatusResponse](#)
- [GetConnectivityInfoResponse](#)
- [GetConnectorDefinitionVersionResponse](#)
- [GetCoreDefinitionVersionResponse](#)
- [GetDeploymentStatusResponse](#)
- [GetDeviceDefinitionVersionResponse](#)
- [GetFunctionDefinitionVersionResponse](#)
- [GetGroupCertificateAuthorityResponse](#)
- [GetGroupVersionResponse](#)
- [GetLoggerDefinitionVersionResponse](#)
- [GetResourceDefinitionVersionResponse](#)
- [GetServiceRoleForAccountResponse](#)
- [GetSubscriptionDefinitionVersionResponse](#)
- [GetThingRuntimeConfigurationResponse](#)
- [Group](#)
- [Group Certificate Authority Properties](#)
- [GroupCertificateConfiguration](#)
- [GroupInformation](#)
- [GroupOwnerSetting](#)
- [GroupVersion](#)
- [IDList](#)
- [ListBulkDeploymentDetailedReportsResponse](#)
- [ListBulkDeploymentsResponse](#)
- [ListDefinitionsResponse](#)
- [ListDeploymentsRequest](#)

- [ListDeploymentsResponse](#)
- [ListGroupCertificateAuthoritiesResponse](#)
- [ListGroupsResponse](#)
- [ListVersionsResponse](#)
- [LocalDeviceResourceData](#)
- [LocalVolumeResourceData](#)
- [Logger](#)
- [LoggerComponent](#)
- [LoggerDefinitionVersion](#)
- [LoggerLevel](#)
- [LoggerType](#)
- [Permission](#)
- [ResetDeploymentsRequest](#)
- [ResetDeploymentsResponse](#)
- [Resource](#)
- [ResourceAccessPolicy](#)
- [ResourceDataContainer](#)
- [ResourceDefinitionVersion](#)
- [ResourceDownloadOwnerSetting](#)
- [RuntimeConfiguration](#)
- [RuntimeConfigurationUpdate](#)
- [S3MachineLearningModelResourceData](#)
- [S3UrlSignerRole](#)
- [SageMakerMachineLearningModelResourceData](#)
- [SecretsManagerSecretResourceData](#)
- [SoftwareToUpdate](#)
- [StartBulkDeploymentRequest](#)
- [StartBulkDeploymentResponse](#)
- [Subscription](#)
- [SubscriptionDefinitionVersion](#)

- [TelemetryConfiguration](#)
- [TelemetryConfigurationUpdate](#)
- [UpdateAgentLogLevel](#)
- [UpdateConnectivityInfoRequest](#)
- [UpdateConnectivityInfoResponse](#)
- [UpdateTargets](#)
- [UpdateTargetsArchitecture](#)
- [UpdateTargetsOperatingSystem](#)
- [VersionInformation](#)
- [Versions](#)
- [tags](#)
- [updateGroupCertificateConfigurationRequest](#)

AssociateRoleToGroupRequest

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

AssociateRoleToGroupRequest

type: object

required: ["RoleArn"]

RoleArn

The ARN of the role to associate with this group.

type: string

AssociateServiceRoleToAccountRequest

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

AssociateServiceRoleToAccountRequest

type: object

required: ["RoleArn"]

RoleArn

The ARN of the service role to associate with your account.

type: string

AssociateServiceRoleToAccountResponse

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

AssociateServiceRoleToAccountResponse

type: object

AssociatedAt

The time when the service role was associated with the account.

type: string

BulkDeployment

```
{
  "BulkDeploymentId": "string",
  "BulkDeploymentArn": "string",
  "CreatedAt": "string"
}
```

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

BulkDeploymentMetrics

```
{
  "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

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

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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

BulkDeploymentResults

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

type: array

items: [BulkDeploymentResult](#)

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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

BulkDeploymentStatus

```
{  
}
```

The current status of the bulk deployment.

type: string

enum: ["Initializing", "Running", "Completed", "Stopping", "Stopped", "Failed"]

BulkDeployments

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



```
]
```

type: array

items: [BulkDeployment](#)

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

ConnectivityInfo

```
{
  "Id": "string",
  "HostAddress": "string",
  "PortNumber": 0x01,
  "Metadata": "string"
}
```

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"
  }
}
```

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.

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](#)

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

```
{
```

```
"Id": "string",
"ThingArn": "string",
"CertificateArn": "string",
"SyncShadow": true
}
```

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

CoreDefinitionVersion

```
{
  "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](#)

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

CreateDeploymentRequest

```
{
  "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

CreateDeploymentResponse

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

CreateDeploymentResponse

type: object

DeploymentArn

The ARN of the deployment.

type: string

DeploymentId

The ID of the deployment.

type: string

CreateGroupCertificateAuthorityResponse

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

CreateGroupCertificateAuthorityResponse

type: object

GroupCertificateAuthorityArn

The ARN of the group certificate authority.

type: string

CreateSoftwareUpdateJobRequest

```
{
```



```
"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

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"]

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",
  "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](#)

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](#)

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

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

DisassociateRoleFromGroupResponse

type: object

DisassociatedAt

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

type: string

DisassociateServiceRoleFromAccountResponse

```
{
  "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.

type: array

items: [ErrorDetail](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

Function

```
{
  "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
    }
  }
},
"FunctionRuntimeOverride": "string"
}
}

```

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](#)

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

FunctionRuntimeOverride

The Lambda runtime supported by Greengrass which is to be used instead of the one specified in the Lambda function.

type: 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
      }
    }
  },
}
```



```
"FunctionRuntimeOverride": "string"
}
```

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](#)

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.

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

FunctionRuntimeOverride

The Lambda runtime supported by Greengrass which is to be used instead of the one specified in the Lambda function.

type: string

FunctionConfigurationEnvironment

```
{
  "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 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](#)

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

FunctionDefaultConfig

```
{
  "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.

type: object

Execution

Configuration information that specifies how a Lambda function runs.

FunctionDefaultExecutionConfig

```
{  
  "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

```
{
  "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
    }
  }
},
"FunctionRuntimeOverride": "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](#)

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](#)

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

FunctionRuntimeOverride

The Lambda runtime supported by Greengrass which is to be used instead of the one specified in the Lambda function.

type: string

FunctionExecutionConfig

```
{
  "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

FunctionIsolationMode

```
{
```

```
}
```

FunctionIsolationMode

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"]

FunctionRunAsConfig

```
{  
  "Uid": 1001,  
  "Gid": 1002  
}
```

FunctionRunAsConfig

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

GeneralError

```
{
  "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](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

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

```
{
  "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

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

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](#)

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](#)

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

GetConnectorDefinitionVersionResponse

```
{
  "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](#)

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

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](#)

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",
  "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

ErrorDetails

A list of error details.

type: array

items: [ErrorDetail](#)

ErrorDetail

Details about the error.

type: object

DetailedErrorCode

A detailed error code.

type: string

DetailedErrorMessage

A detailed error message.

type: string

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](#)

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

```
{
  "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
        }
      }
    },
    "FunctionRuntimeOverride": "string"
  }
},
"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](#)

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](#)

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

FunctionRuntimeOverride

The Lambda runtime supported by Greengrass which is to be used instead of the one specified in the Lambda function.

type: string

NextToken

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

type: string

GetGroupCertificateAuthorityResponse

```
{
  "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

GetGroupVersionResponse

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

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 client 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](#)

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

```
{
  "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](#)

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

GetServiceRoleForAccountResponse

```
{
  "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

```
{
```

```
"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](#)

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

GetThingRuntimeConfigurationResponse

```
{
  "RuntimeConfiguration": {
    "TelemetryConfiguration": {
      "Telemetry": "On|Off",
      "ConfigurationSyncStatus": "InSync|OutOfSync"
    }
  }
}
```

GetThingRuntimeConfigurationResponse

The runtime configuration for a thing.

type: object

RuntimeConfiguration

Runtime configuration for a thing.

type: object

items: [TelemetryConfiguration](#)

TelemetryConfiguration

The configuration settings to run telemetry.

type: object

required: ["Telemetry"]

Telemetry

The configuration setting to turn on or turn off telemetry.

type: string

enum: ["On", "Off"]

ConfigurationSyncStatus

The synchronization status of the device-reported configuration with the desired configuration.

type: string

enum: ["InSync", "OutOfSync"]

Group

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

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

```
{
  "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

```
{
  "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

```
{
  "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 client 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

IDList

```
{
  "Ids": [
    "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"
      }
    ]
  },
  "NextToken": "string"
}
```

ListBulkDeploymentDetailedReportsResponse

type: object

Deployments

type: array

items: [BulkDeploymentResult](#)

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](#)

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](#)

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

ListDefinitionsResponse

```
{
  "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](#)

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](#)

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

ListGroupCertificateAuthoritiesResponse

```
{
  "GroupCertificateAuthorities": [
    {
      "GroupCertificateAuthorityId": "string",
      "GroupCertificateAuthorityArn": "string"
    }
  ]
}
```

ListGroupCertificateAuthoritiesResponse

type: object

GroupCertificateAuthorities

A list of certificate authorities associated with the group.

type: array

items: [GroupCertificateAuthorityProperties](#)

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

```
{
```

```
"Groups": [  
  {  
    "Name": "string",  
    "Id": "string",  
    "Arn": "string",  
    "LastUpdatedTimestamp": "string",  
    "CreationTimestamp": "string",  
    "LatestVersion": "string",  
    "LatestVersionArn": "string"  
  },  
  {  
    "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](#)

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

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](#)

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.

type: string

LocalVolumeResourceData

```
{
  "SourcePath": "string",
  "DestinationPath": "string",
  "GroupOwnerSetting": {
    "AutoAddGroupOwner": true,
    "GroupOwner": "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

Logger

```
{
  "Id": "string",
  "Type": "FileSystem|AWSCloudWatch",
  "Component": "GreengrassSystem|Lambda",
  "Level": "DEBUG|INFO|WARN|ERROR|FATAL",
  "Space": 0
}
```

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",  
      "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](#)

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

LogLevel

```
{  
}
```

type: string

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

LoggerType

```
{  
}
```

type: string

enum: ["FileSystem", "AWSCloudWatch"]

Permission

```
{  
}
```

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

type: string

enum: ["ro", "rw"]

ResetDeploymentsRequest

```
{  
  "Force": true  
}
```

ResetDeploymentsRequest

Information about a group reset request.

type: object

Force

If true, performs a best-effort only core reset.

type: boolean

ResetDeploymentsResponse

```
{
  "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.

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

ResourceAccessPolicy

```
{  
  "ResourceId": "string",  
  "Permission": "ro|rw"  
}
```

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"]

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"  
    ]  
  }  
}
```

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

ResourceDefinitionVersion

```
{
  "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"
      ]
    }
  }
}
```

ResourceDefinitionVersion

Information about a resource definition version.

type: object

Resources

A list of resources.

type: array

items: [Resource](#)

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

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"]

RuntimeConfiguration

```
{
  "TelemetryConfiguration": {
    "Telemetry": "On|Off",
    "ConfigurationSyncStatus": "InSync|OutOfSync"
  }
}
```

RuntimeConfiguration

Runtime configuration for a thing.

type: object

items: [TelemetryConfiguration](#)

TelemetryConfiguration

The configuration settings to run telemetry.

type: object

required: ["Telemetry"]

Telemetry

The configuration setting to turn on or turn off telemetry.

type: string

enum: ["On", "Off"]

ConfigurationSyncStatus

The synchronization status of the device-reported configuration with the desired configuration.

type: string

enum: ["InSync", "OutOfSync"]

RuntimeConfigurationUpdate

```
{
  "TelemetryConfiguration": {
    "Telemetry": "On|Off"
  }
}
```

RuntimeConfigurationUpdate

Runtime configuration for a thing.

type: object

[TelemetryConfiguration](#)

The configuration settings to run telemetry.

type: object

required: ["Telemetry"]

Telemetry

The configuration setting to turn on or turn off telemetry.

type: string

enum: ["On", "Off"]

S3MachineLearningModelResourceData

```
{
  "S3Uri": "string",
  "DestinationPath": "string",
  "OwnerSetting": {
    "GroupOwner": "string",
    "GroupPermission": "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"]

S3UrlSignerRole

```
{  
}
```

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

type: string

SageMakerMachineLearningModelResourceData

```
{  
  "SageMakerJobArn": "string",  
  "DestinationPath": "string",  
  "OwnerSetting": {  
    "GroupOwner": "string",  
    "GroupPermission": "ro|rw"  
  }  
}
```

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

```
{
  "ARN": "string",
  "AdditionalStagingLabelsToDownload": [
    "string"
  ]
}
```

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

```
{  
  "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 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

StartBulkDeploymentResponse

```
{  
  "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

Subscription

```
{
  "Id": "string",
  "Source": "string",
  "Subject": "string",
  "Target": "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](#)

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

TelemetryConfiguration

```
{
  "Telemetry": "On|Off",
  "ConfigurationSyncStatus": "InSync|OutOfSync"
}
```

TelemetryConfiguration

The configuration settings to run telemetry.

type: object

required: ["Telemetry"]

Telemetry

The configuration setting to turn on or turn off telemetry.

type: string

enum: ["On", "Off"]

ConfigurationSyncStatus

The synchronization status of the device-reported configuration with the desired configuration.

type: string

enum: ["InSync", "OutOfSync"]

TelemetryConfigurationUpdate

```
{
  "Telemetry": "On|Off"
}
```

TelemetryConfigurationUpdate

The configuration settings to run telemetry.

type: object

required: ["Telemetry"]

Telemetry

The configuration setting to turn on or turn off telemetry.

type: string

enum: ["On", "Off"]

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,  
      "Metadata": "string"  
    }  
  ]  
}
```

Information required to update a Greengrass core's connectivity.

ConnectivityInfo

A list of connectivity info.

type: array

items: [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

UpdateConnectivityInfoResponse

```
{  
  "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.

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](#)

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

tags

```
{  
  "additionalProperty0": "string",  
  "additionalProperty1": "string",  
  "additionalProperty2": "string"  
}
```

The resource tags.

type: object

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

updateGroupCertificateConfigurationRequest

```
{  
  "CertificateExpiryInMilliseconds": "string"  
}
```

updateGroupCertificateConfigurationRequest

type: object

required: ["CertificateExpiryInMilliseconds"]

CertificateExpiryInMilliseconds

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

type: string