API Reference

AWS Telco Network Builder

API Version 2008-10-21

AWS Telco Network Builder: API Reference

Table of Contents

Welcome

AWS Telco Network Builder (TNB) is a network automation service that helps you deploy and manage telecom networks. AWS TNB helps you with the lifecycle management of your telecommunication network functions throughout planning, deployment, and post-deployment activities.

This document was last published on July 22, 2025.

Actions

The following actions are supported:

- CancelSolNetworkOperation
- CreateSolFunctionPackage
- CreateSolNetworkInstance
- CreateSolNetworkPackage
- DeleteSolFunctionPackage
- DeleteSolNetworkInstance
- DeleteSolNetworkPackage
- GetSolFunctionInstance
- GetSolFunctionPackage
- GetSolFunctionPackageContent
- GetSolFunctionPackageDescriptor
- GetSolNetworkInstance
- GetSolNetworkOperation
- GetSolNetworkPackage
- GetSolNetworkPackageContent
- GetSolNetworkPackageDescriptor
- InstantiateSolNetworkInstance
- ListSolFunctionInstances
- ListSolFunctionPackages
- ListSolNetworkInstances
- ListSolNetworkOperations
- ListSolNetworkPackages
- ListTagsForResource
- PutSolFunctionPackageContent
- PutSolNetworkPackageContent
- TagResource
- TerminateSolNetworkInstance

- UntagResource
- UpdateSolFunctionPackage
- UpdateSolNetworkInstance
- UpdateSolNetworkPackage
- ValidateSolFunctionPackageContent
- ValidateSolNetworkPackageContent

CancelSolNetworkOperation

Cancels a network operation.

A network operation is any operation that is done to your network, such as network instance instantiation or termination.

Request Syntax

```
POST /sol/nslcm/v1/ns_lcm_op_occs/nsLcmOpOccId/cancel HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

nsLcmOpOccId

The identifier of the network operation.

Pattern: `no-[a-f0-9]{17}`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 202
```

Response Elements

If the action is successful, the service sends back an HTTP 202 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3

- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateSolFunctionPackage

Creates a function package.

A function package is a .zip file in CSAR (Cloud Service Archive) format that contains a network function (an ETSI standard telecommunication application) and function package descriptor that uses the TOSCA standard to describe how the network functions should run on your network. For more information, see Function packages in the *AWS Telco Network Builder User Guide*.

Creating a function package is the first step for creating a network in AWS TNB. This request creates an empty container with an ID. The next step is to upload the actual CSAR zip file into that empty container. To upload function package content, see PutSolFunctionPackageContent.

Request Syntax

```
POST /sol/vnfpkgm/v1/vnf_packages HTTP/1.1
Content-type: application/json

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

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Pattern: `(?!aws:).{1,128}`

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

Required: No

Response Syntax

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

{
   "arn": "string",
   "id": "string",
   "onboardingState": "string",
   "operationalState": "string",
   "tags": {
      "string" : "string"
   },
   "usageState": "string"
}
```

Response Elements

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

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

arn

Function package ARN.

Type: String

Pattern: `arn:(aws|aws-cn|aws-iso|aws-iso-b|aws-us-gov):tnb:([a-z]{2}(-(gov|isob|iso))?-(east|west|north|south|central){1,2}-[0-9]):\d{12}:(function-package/fp-[a-f0-9]{17})`

id

ID of the function package.

Type: String

Pattern: `fp-[a-f0-9]{17}`

onboardingState

Onboarding state of the function package.

Type: String

Valid Values: `CREATED | ONBOARDED | ERROR`

operationalState

Operational state of the function package.

Type: String

Valid Values: `ENABLED | DISABLED`

tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Pattern: `(?!aws:).{1,128}`

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

usageState

Usage state of the function package.

Type: String

Valid Values: `IN_USE | NOT_IN_USE`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ServiceQuotaExceededException

Service quotas have been exceeded.

HTTP Status Code: 402

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3

- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateSolNetworkInstance

Creates a network instance.

A network instance is a single network created in AWS TNB that can be deployed and on which life-cycle operations (like terminate, update, and delete) can be performed. Creating a network instance is the third step after creating a network package. For more information about network instances, Network instances in the *AWS Telco Network Builder User Guide*.

Once you create a network instance, you can instantiate it. To instantiate a network, see InstantiateSolNetworkInstance.

Request Syntax

```
POST /sol/nslcm/v1/ns_instances HTTP/1.1
Content-type: application/json

{
   "nsDescription": "string",
   "nsdInfoId": "string",
   "nsName": "string",
   "tags": {
      "string" : "string"
   }
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

nsDescription

Network instance description.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 255.

Required: No

nsdInfoId

ID for network service descriptor.

Type: String

Pattern: `np-[a-f0-9]{17}`

Required: Yes

nsName

Network instance name.

Type: String

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

Required: Yes

tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Pattern: `(?!aws:).{1,128}`

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

Required: No

Response Syntax

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

{
   "arn": "string",
```

```
   "id": "string",
   "nsdInfoId": "string",
   "nsInstanceName": "string",
   "tags": {
      "string" : "string"
   }
}
```

Response Elements

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

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

arn

Network instance ARN.

Type: String

Pattern: `arn:(aws|aws-cn|aws-iso|aws-iso-b|aws-us-gov):tnb:([a-z]{2}(-(gov|isob|iso))?-(east|west|north|south|central){1,2}-[0-9]):\d{12}:(network-instance/ni-[a-f0-9]{17})`

id

Network instance ID.

Type: String

Pattern: `ni-[a-f0-9]{17}`

nsdInfoId

Network service descriptor ID.

Type: String

Pattern: `np-[a-f0-9]{17}`

nsInstanceName

Network instance name.

Type: String

tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Pattern: `(?!aws:).{1,128}`

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

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ServiceQuotaExceededException

Service quotas have been exceeded.

HTTP Status Code: 402

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateSolNetworkPackage

Creates a network package.

A network package is a .zip file in CSAR (Cloud Service Archive) format defines the function packages you want to deploy and the AWS infrastructure you want to deploy them on. For more information, see Network instances in the *AWS Telco Network Builder User Guide*.

A network package consists of a network service descriptor (NSD) file (required) and any additional files (optional), such as scripts specific to your needs. For example, if you have multiple function packages in your network package, you can use the NSD to define which network functions should run in certain VPCs, subnets, or EKS clusters.

This request creates an empty network package container with an ID. Once you create a network package, you can upload the network package content using PutSolNetworkPackageContent.

Request Syntax

```
POST /sol/nsd/v1/ns_descriptors HTTP/1.1
Content-type: application/json

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

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Pattern: `(?!aws:).{1,128}`

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

Required: No

Response Syntax

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

{
   "arn": "string",
   "id": "string",
   "nsdOnboardingState": "string",
   "nsdOperationalState": "string",
   "nsdUsageState": "string",
   "tags": {
      "string" : "string"
   }
}
```

Response Elements

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

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

arn

Network package ARN.

Type: String

Pattern: `arn:(aws|aws-cn|aws-iso|aws-iso-b|aws-us-gov):tnb:([a-z]{2}(-(gov|isob|iso))?-(east|west|north|south|central){1,2}-[0-9]):\d{12}:(network-package/np-[a-f0-9]{17})`

id

ID of the network package.

Type: String

Pattern: `np-[a-f0-9]{17}`

nsdOnboardingState

Onboarding state of the network service descriptor in the network package.

Type: String

Valid Values: `CREATED | ONBOARDED | ERROR`

nsdOperationalState

Operational state of the network service descriptor in the network package.

Type: String

Valid Values: `ENABLED | DISABLED`

nsdUsageState

Usage state of the network service descriptor in the network package.

Type: String

Valid Values: `IN_USE | NOT_IN_USE`

tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Pattern: `(?!aws:).{1,128}`

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

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ServiceQuotaExceededException

Service quotas have been exceeded.

HTTP Status Code: 402

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3

- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteSolFunctionPackage

Deletes a function package.

A function package is a .zip file in CSAR (Cloud Service Archive) format that contains a network function (an ETSI standard telecommunication application) and function package descriptor that uses the TOSCA standard to describe how the network functions should run on your network.

To delete a function package, the package must be in a disabled state. To disable a function package, see UpdateSolFunctionPackage.

Request Syntax

```
DELETE /sol/vnfpkgm/v1/vnf_packages/vnfPkgId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

vnfPkgId

ID of the function package.

Pattern: `fp-[a-f0-9]{17}`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 204
```

Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteSolNetworkInstance

Deletes a network instance.

A network instance is a single network created in AWS TNB that can be deployed and on which life-cycle operations (like terminate, update, and delete) can be performed.

To delete a network instance, the instance must be in a stopped or terminated state. To terminate a network instance, see TerminateSolNetworkInstance.

Request Syntax

```
DELETE /sol/nslcm/v1/ns_instances/nsInstanceId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

nsInstanceId

Network instance ID.

Pattern: `ni-[a-f0-9]{17}`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 204
```

Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteSolNetworkPackage

Deletes network package.

A network package is a .zip file in CSAR (Cloud Service Archive) format defines the function packages you want to deploy and the AWS infrastructure you want to deploy them on.

To delete a network package, the package must be in a disable state. To disable a network package, see UpdateSolNetworkPackage.

Request Syntax

```
DELETE /sol/nsd/v1/ns_descriptors/nsdInfoId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

nsdInfoId

ID of the network service descriptor in the network package.

Pattern: `np-[a-f0-9]{17}`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 204
```

Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetSolFunctionInstance

Gets the details of a network function instance, including the instantiation state and metadata from the function package descriptor in the network function package.

A network function instance is a function in a function package .

Request Syntax

```
GET /sol/vnflcm/v1/vnf_instances/vnfInstanceId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

vnfInstanceId

ID of the network function.

Pattern: `fi-[a-f0-9]{17}`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

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

{
   "arn": "string",
   "id": "string",
   "instantiatedVnfInfo": {
      "vnfcResourceInfo": [
         {
            "metadata": {
               "cluster": "string",
```

```
                "helmChart": "string",
                "nodeGroup": "string"
            }
         }
      ],
      "vnfState": "string"
   },
   "instantiationState": "string",
   "metadata": {
      "createdAt": "string",
      "lastModified": "string"
   },
   "nsInstanceId": "string",
   "tags": {
      "string" : "string"
   },
   "vnfdId": "string",
   "vnfdVersion": "string",
   "vnfPkgId": "string",
   "vnfProductName": "string",
   "vnfProvider": "string"
}
```

Response Elements

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

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

arn

Network function instance ARN.

Type: String

Pattern: `arn:(aws|aws-cn|aws-iso|aws-iso-b|aws-us-gov):tnb:([a-z]{2}(-(gov|isob|iso))?-(east|west|north|south|central){1,2}-[0-9]):\d{12}:(function-instance/fi-[a-f0-9]{17})`

id

Network function instance ID.

Type: String

Pattern: `fi-[a-f0-9]{17}`

instantiatedVnfInfo

Information about the network function.

A network function instance is a function in a function package .

Type: GetSolVnfInfo object

instantiationState

Network function instantiation state.

Type: String

Valid Values: `INSTANTIATED | NOT_INSTANTIATED`

metadata

The metadata of a network function instance.

A network function instance is a function in a function package .

Type: GetSolFunctionInstanceMetadata object

nsInstanceId

Network instance ID.

Type: String

Pattern: `ni-[a-f0-9]{17}`

tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Pattern: `(?!aws:).{1,128}`

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

vnfdId

Function package descriptor ID.

Type: String

Pattern: `[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}`

vnfdVersion

Function package descriptor version.

Type: String

vnfPkgId

Function package ID.

Type: String

Pattern: `fp-[a-f0-9]{17}`

vnfProductName

Network function product name.

Type: String

vnfProvider

Network function provider.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetSolFunctionPackage

Gets the details of an individual function package, such as the operational state and whether the package is in use.

A function package is a .zip file in CSAR (Cloud Service Archive) format that contains a network function (an ETSI standard telecommunication application) and function package descriptor that uses the TOSCA standard to describe how the network functions should run on your network..

Request Syntax

```
GET /sol/vnfpkgm/v1/vnf_packages/vnfPkgId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

vnfPkgId

ID of the function package.

Pattern: `fp-[a-f0-9]{17}`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

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

{
   "arn": "string",
   "id": "string",
   "metadata": {
      "createdAt": "string",
      "lastModified": "string",
```

```
      "vnfd": {
         "overrides": [
            {
               "defaultValue": "string",
               "name": "string"
            }
         ]
      }
   },
   "onboardingState": "string",
   "operationalState": "string",
   "tags": {
      "string" : "string"
   },
   "usageState": "string",
   "vnfdId": "string",
   "vnfdVersion": "string",
   "vnfProductName": "string",
   "vnfProvider": "string"
}
```

Response Elements

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

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

arn

Function package ARN.

Type: String

Pattern: `arn:(aws|aws-cn|aws-iso|aws-iso-b|aws-us-gov):tnb:([a-z]{2}(-(gov|isob|iso))?-(east|west|north|south|central){1,2}-[0-9]):\d{12}:(function-package/fp-[a-f0-9]{17})`

id

Function package ID.

Type: String

Pattern: `fp-[a-f0-9]{17}`

metadata

Metadata related to the function package.

A function package is a .zip file in CSAR (Cloud Service Archive) format that contains a network function (an ETSI standard telecommunication application) and function package descriptor that uses the TOSCA standard to describe how the network functions should run on your network.

Type: GetSolFunctionPackageMetadata object

onboardingState

Function package onboarding state.

Type: String

Valid Values: `CREATED | ONBOARDED | ERROR`

operationalState

Function package operational state.

Type: String

Valid Values: `ENABLED | DISABLED`

tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Pattern: `(?!aws:).{1,128}`

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

usageState

Function package usage state.

Type: String

Valid Values: `IN_USE | NOT_IN_USE`

vnfdId

Function package descriptor ID.

Type: String

vnfdVersion

Function package descriptor version.

Type: String

vnfProductName

Network function product name.

Type: String

vnfProvider

Network function provider.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetSolFunctionPackageContent

Gets the contents of a function package.

A function package is a .zip file in CSAR (Cloud Service Archive) format that contains a network function (an ETSI standard telecommunication application) and function package descriptor that uses the TOSCA standard to describe how the network functions should run on your network.

Request Syntax

```
GET /sol/vnfpkgm/v1/vnf_packages/vnfPkgId/package_content HTTP/1.1
Accept: accept
```

URI Request Parameters

The request uses the following URI parameters.

accept

The format of the package that you want to download from the function packages.

Valid Values: `application/zip`

Required: Yes

vnfPkgId

ID of the function package.

Pattern: `fp-[a-f0-9]{17}`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
```

```
Content-Type: contentType

packageContent
```

Response Elements

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

The response returns the following HTTP headers.

contentType

Indicates the media type of the resource.

Valid Values: `application/zip`

The response returns the following as the HTTP body.

packageContent

Contents of the function package.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetSolFunctionPackageDescriptor

Gets a function package descriptor in a function package.

A function package descriptor is a .yaml file in a function package that uses the TOSCA standard to describe how the network function in the function package should run on your network.

A function package is a .zip file in CSAR (Cloud Service Archive) format that contains a network function (an ETSI standard telecommunication application) and function package descriptor that uses the TOSCA standard to describe how the network functions should run on your network.

Request Syntax

```
GET /sol/vnfpkgm/v1/vnf_packages/vnfPkgId/vnfd HTTP/1.1
Accept: accept
```

URI Request Parameters

The request uses the following URI parameters.

accept

Indicates which content types, expressed as MIME types, the client is able to understand.

Valid Values: `text/plain`

Required: Yes

vnfPkgId

ID of the function package.

Pattern: `fp-[a-f0-9]{17}`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-Type: contentType

vnfd
```

Response Elements

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

The response returns the following HTTP headers.

contentType

Indicates the media type of the resource.

Valid Values: `text/plain`

The response returns the following as the HTTP body.

vnfd

Contents of the function package descriptor.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetSolNetworkInstance

Gets the details of the network instance.

A network instance is a single network created in AWS TNB that can be deployed and on which life-cycle operations (like terminate, update, and delete) can be performed.

Request Syntax

```
GET /sol/nslcm/v1/ns_instances/nsInstanceId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

nsInstanceId

ID of the network instance.

Pattern: `ni-[a-f0-9]{17}`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

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

{
   "arn": "string",
   "id": "string",
   "lcmOpInfo": {
      "nsLcmOpOccId": "string"
   },
   "metadata": {
      "createdAt": "string",
```

```
      "lastModified": "string"
   },
   "nsdId": "string",
   "nsdInfoId": "string",
   "nsInstanceDescription": "string",
   "nsInstanceName": "string",
   "nsState": "string",
   "tags": {
      "string" : "string"
   }
}
```

Response Elements

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

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

arn

Network instance ARN.

Type: String

Pattern: `arn:(aws|aws-cn|aws-iso|aws-iso-b|aws-us-gov):tnb:([a-z]{2}(-(gov|isob|iso))?-(east|west|north|south|central){1,2}-[0-9]):\d{12}:(network-instance/ni-[a-f0-9]{17})`

id

Network instance ID.

Type: String

Pattern: `ni-[a-f0-9]{17}`

lcmOpInfo

Lifecycle management operation details on the network instance.

Lifecycle management operations are deploy, update, or delete operations.

Type: LcmOperationInfo object

metadata

The metadata of a network instance.

A network instance is a single network created in AWS TNB that can be deployed and on which life-cycle operations (like terminate, update, and delete) can be performed.

Type: GetSolNetworkInstanceMetadata object

nsdId

Network service descriptor ID.

Type: String

Pattern: `[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}`

nsdInfoId

Network service descriptor info ID.

Type: String

Pattern: `np-[a-f0-9]{17}`

nsInstanceDescription

Network instance description.

Type: String

nsInstanceName

Network instance name.

Type: String

nsState

Network instance state.

Type: String

Valid Values: `INSTANTIATED | NOT_INSTANTIATED | UPDATED | IMPAIRED | UPDATE_FAILED | STOPPED | DELETED | INSTANTIATE_IN_PROGRESS | INTENT_TO_UPDATE_IN_PROGRESS | UPDATE_IN_PROGRESS | TERMINATE_IN_PROGRESS`

tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Pattern: `(?!aws:).{1,128}`

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

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetSolNetworkOperation

Gets the details of a network operation, including the tasks involved in the network operation and the status of the tasks.

A network operation is any operation that is done to your network, such as network instance instantiation or termination.

Request Syntax

```
GET /sol/nslcm/v1/ns_lcm_op_occs/nsLcmOpOccId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

nsLcmOpOccId

The identifier of the network operation.

Pattern: `no-[a-f0-9]{17}`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

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

{
   "arn": "string",
   "error": {
      "detail": "string",
      "title": "string"
   },
   "id": "string",
```

```
   "lcmOperationType": "string",
   "metadata": {
      "createdAt": "string",
      "instantiateMetadata": {
         "additionalParamsForNs": JSON value,
         "nsdInfoId": "string"
      },
      "lastModified": "string",
      "modifyVnfInfoMetadata": {
         "vnfConfigurableProperties": JSON value,
         "vnfInstanceId": "string"
      },
      "updateNsMetadata": {
         "additionalParamsForNs": JSON value,
         "nsdInfoId": "string"
      }
   },
   "nsInstanceId": "string",
   "operationState": "string",
   "tags": {
      "string" : "string"
   },
   "tasks": [
      {
         "taskContext": {
            "string" : "string"
         },
         "taskEndTime": "string",
         "taskErrorDetails": {
            "cause": "string",
            "details": "string"
         },
         "taskName": "string",
         "taskStartTime": "string",
         "taskStatus": "string"
      }
   ],
   "updateType": "string"
}
```

Response Elements

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

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

arn

Network operation ARN.

Type: String

Pattern: `arn:(aws|aws-cn|aws-iso|aws-iso-b|aws-us-gov):tnb:([a-z]{2}(-(gov|isob|iso))?-(east|west|north|south|central){1,2}-[0-9]):\d{12}:(network-operation/no-[a-f0-9]{17})`

error

Error related to this specific network operation occurrence.

Type: ProblemDetails object

id

ID of this network operation occurrence.

Type: String

Pattern: `no-[a-f0-9]{17}`

lcmOperationType

Type of the operation represented by this occurrence.

Type: String

Valid Values: `INSTANTIATE | UPDATE | TERMINATE`

metadata

Metadata of this network operation occurrence.

Type: GetSolNetworkOperationMetadata object

nsInstanceId

ID of the network operation instance.

Type: String

Pattern: `ni-[a-f0-9]{17}`

operationState

The state of the network operation.

Type: String

Valid Values: `PROCESSING | COMPLETED | FAILED | CANCELLING | CANCELLED`

tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Pattern: `(?!aws:).{1,128}`

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

tasks

All tasks associated with this operation occurrence.

Type: Array of GetSolNetworkOperationTaskDetails objects

updateType

Type of the update. Only present if the network operation lcmOperationType is `UPDATE`.

Type: String

Valid Values: `MODIFY_VNF_INFORMATION | UPDATE_NS`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetSolNetworkPackage

Gets the details of a network package.

A network package is a .zip file in CSAR (Cloud Service Archive) format defines the function packages you want to deploy and the AWS infrastructure you want to deploy them on.

Request Syntax

```
GET /sol/nsd/v1/ns_descriptors/nsdInfoId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

nsdInfoId

ID of the network service descriptor in the network package.

Pattern: np-[a-f0-9]{17}

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

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

{
   "arn": "string",
   "id": "string",
   "metadata": {
      "createdAt": "string",
      "lastModified": "string",
      "nsd": {
```

```
         "overrides": [
            {
               "defaultValue": "string",
               "name": "string"
            }
         ]
      }
   },
   "nsdId": "string",
   "nsdName": "string",
   "nsdOnboardingState": "string",
   "nsdOperationalState": "string",
   "nsdUsageState": "string",
   "nsdVersion": "string",
   "tags": {
      "string" : "string"
   },
   "vnfPkgIds": [ "string" ]
}
```

Response Elements

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

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

arn

Network package ARN.

Type: String

Pattern: `arn:(aws|aws-cn|aws-iso|aws-iso-b|aws-us-gov):tnb:([a-z]{2}(-(gov|isob|iso))?-(east|west|north|south|central){1,2}-[0-9]):\d{12}:(network-package/np-[a-f0-9]{17})`

id

Network package ID.

Type: String

Pattern: `np-[a-f0-9]{17}`

metadata

Metadata associated with a network package.

A network package is a .zip file in CSAR (Cloud Service Archive) format defines the function packages you want to deploy and the AWS infrastructure you want to deploy them on.

Type: GetSolNetworkPackageMetadata object

nsdId

Network service descriptor ID.

Type: String

Pattern: `[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}`

nsdName

Network service descriptor name.

Type: String

nsdOnboardingState

Network service descriptor onboarding state.

Type: String

Valid Values: `CREATED | ONBOARDED | ERROR`

nsdOperationalState

Network service descriptor operational state.

Type: String

Valid Values: `ENABLED | DISABLED`

nsdUsageState

Network service descriptor usage state.

Type: String

Valid Values: `IN_USE | NOT_IN_USE`

nsdVersion

Network service descriptor version.

Type: String

tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Pattern: `(?!aws:).{1,128}`

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

vnfPkgIds

Identifies the function package for the function package descriptor referenced by the onboarded network package.

Type: Array of strings

Pattern: `fp-[a-f0-9]{17}`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetSolNetworkPackageContent

Gets the contents of a network package.

A network package is a .zip file in CSAR (Cloud Service Archive) format defines the function packages you want to deploy and the AWS infrastructure you want to deploy them on.

Request Syntax

```
GET /sol/nsd/v1/ns_descriptors/nsdInfoId/nsd_content HTTP/1.1
Accept: accept
```

URI Request Parameters

The request uses the following URI parameters.

accept

The format of the package you want to download from the network package.

Valid Values: `application/zip`

Required: Yes

nsdInfoId

ID of the network service descriptor in the network package.

Pattern: `np-[a-f0-9]{17}`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-Type: contentType
```

```
nsdContent
```

Response Elements

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

The response returns the following HTTP headers.

contentType

Indicates the media type of the resource.

Valid Values: `application/zip`

The response returns the following as the HTTP body.

nsdContent

Content of the network service descriptor in the network package.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetSolNetworkPackageDescriptor

Gets the content of the network service descriptor.

A network service descriptor is a .yaml file in a network package that uses the TOSCA standard to describe the network functions you want to deploy and the AWS infrastructure you want to deploy the network functions on.

Request Syntax

```
GET /sol/nsd/v1/ns_descriptors/nsdInfoId/nsd HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

nsdInfoId

ID of the network service descriptor in the network package.

Pattern: `np-[a-f0-9]{17}`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-Type: contentType

nsd
```

Response Elements

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

The response returns the following HTTP headers.

contentType

Indicates the media type of the resource.

Valid Values: `text/plain`

The response returns the following as the HTTP body.

nsd

Contents of the network service descriptor in the network package.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

InstantiateSolNetworkInstance

Instantiates a network instance.

A network instance is a single network created in AWS TNB that can be deployed and on which life-cycle operations (like terminate, update, and delete) can be performed.

Before you can instantiate a network instance, you have to create a network instance. For more information, see CreateSolNetworkInstance.

Request Syntax

```
POST /sol/nslcm/v1/ns_instances/nsInstanceId/instantiate?dry_run=dryRun HTTP/1.1
Content-type: application/json

{
   "additionalParamsForNs": JSON value,
   "tags": {
      "string" : "string"
   }
}
```

URI Request Parameters

The request uses the following URI parameters.

dryRun

A check for whether you have the required permissions for the action without actually making the request and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

nsInstanceId

ID of the network instance.

Pattern: `ni-[a-f0-9]{17}`

Required: Yes

Request Body

The request accepts the following data in JSON format.

additionalParamsForNs

Provides values for the configurable properties.

Type: JSON value

Required: No

tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. When you use this API, the tags are only applied to the network operation that is created. These tags are not applied to the network instance. Use tags to search and filter your resources or track your AWS costs.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Pattern: `(?!aws:).{1,128}`

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

Required: No

Response Syntax

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

{
   "nsLcmOpOccId": "string",
   "tags": {
      "string" : "string"
   }
}
```

Response Elements

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

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

nsLcmOpOccId

The identifier of the network operation.

Type: String

Pattern: `no-[a-f0-9]{17}`

tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. When you use this API, the tags are only applied to the network operation that is created. These tags are not applied to the network instance. Use tags to search and filter your resources or track your AWS costs.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Pattern: `(?!aws:).{1,128}`

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

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ServiceQuotaExceededException

Service quotas have been exceeded.

HTTP Status Code: 402

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListSolFunctionInstances

Lists network function instances.

A network function instance is a function in a function package .

Request Syntax

```
GET /sol/vnflcm/v1/vnf_instances?
max_results=maxResults&nextpage_opaque_marker=nextToken HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

maxResults

The maximum number of results to include in the response.

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

nextToken

The token for the next page of results.

Request Body

The request does not have a request body.

Response Syntax

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

{
   "functionInstances": [
      {
         "arn": "string",
         "id": "string",
         "instantiatedVnfInfo": {
            "vnfState": "string"
```

```
         },
         "instantiationState": "string",
         "metadata": {
            "createdAt": "string",
            "lastModified": "string"
         },
         "nsInstanceId": "string",
         "vnfPkgId": "string",
         "vnfPkgName": "string"
      }
   ],
   "nextToken": "string"
}
```

Response Elements

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

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

functionInstances

Network function instances.

Type: Array of ListSolFunctionInstanceInfo objects

nextToken

The token to use to retrieve the next page of results. This value is `null` when there are no more results to return.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListSolFunctionPackages

Lists information about function packages.

A function package is a .zip file in CSAR (Cloud Service Archive) format that contains a network function (an ETSI standard telecommunication application) and function package descriptor that uses the TOSCA standard to describe how the network functions should run on your network.

Request Syntax

```
GET /sol/vnfpkgm/v1/vnf_packages?
max_results=maxResults&nextpage_opaque_marker=nextToken HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

maxResults

The maximum number of results to include in the response.

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

nextToken

The token for the next page of results.

Request Body

The request does not have a request body.

Response Syntax

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

{
   "functionPackages": [
      {
         "arn": "string",
```

```
            "id": "string",
            "metadata": {
               "createdAt": "string",
               "lastModified": "string"
            },
            "onboardingState": "string",
            "operationalState": "string",
            "usageState": "string",
            "vnfdId": "string",
            "vnfdVersion": "string",
            "vnfProductName": "string",
            "vnfProvider": "string"
         }
      ],
      "nextToken": "string"
}
```

Response Elements

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

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

functionPackages

Function packages. A function package is a .zip file in CSAR (Cloud Service Archive) format that contains a network function (an ETSI standard telecommunication application) and function package descriptor that uses the TOSCA standard to describe how the network functions should run on your network.

Type: Array of ListSolFunctionPackageInfo objects

nextToken

The token to use to retrieve the next page of results. This value is `null` when there are no more results to return.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListSolNetworkInstances

Lists your network instances.

A network instance is a single network created in AWS TNB that can be deployed and on which life-cycle operations (like terminate, update, and delete) can be performed.

Request Syntax

```
GET /sol/nslcm/v1/ns_instances?max_results=maxResults&nextpage_opaque_marker=nextToken
 HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

maxResults

The maximum number of results to include in the response.

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

nextToken

The token for the next page of results.

Request Body

The request does not have a request body.

Response Syntax

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

{
   "networkInstances": [
      {
         "arn": "string",
         "id": "string",
```

```
            "metadata": {
               "createdAt": "string",
               "lastModified": "string"
            },
            "nsdId": "string",
            "nsdInfoId": "string",
            "nsInstanceDescription": "string",
            "nsInstanceName": "string",
            "nsState": "string"
         }
      ],
      "nextToken": "string"
}
```

Response Elements

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

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

networkInstances

Lists network instances.

Type: Array of ListSolNetworkInstanceInfo objects

nextToken

The token to use to retrieve the next page of results. This value is `null` when there are no more results to return.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListSolNetworkOperations

Lists details for a network operation, including when the operation started and the status of the operation.

A network operation is any operation that is done to your network, such as network instance instantiation or termination.

Request Syntax

```
GET /sol/nslcm/v1/ns_lcm_op_occs?
max_results=maxResults&nextpage_opaque_marker=nextToken&nsInstanceId=nsInstanceId
 HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

maxResults

The maximum number of results to include in the response.

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

nextToken

The token for the next page of results.

nsInstanceId

Network instance id filter, to retrieve network operations associated to a network instance.

Pattern: `ni-[a-f0-9]{17}`

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
```

```
Content-type: application/json

{
   "networkOperations": [
      {
         "arn": "string",
         "error": {
            "detail": "string",
            "title": "string"
         },
         "id": "string",
         "lcmOperationType": "string",
         "metadata": {
            "createdAt": "string",
            "lastModified": "string",
            "nsdInfoId": "string",
            "vnfInstanceId": "string"
         },
         "nsInstanceId": "string",
         "operationState": "string",
         "updateType": "string"
      }
   ],
   "nextToken": "string"
}
```

Response Elements

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

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

networkOperations

Lists network operation occurrences. Lifecycle management operations are deploy, update, or delete operations.

Type: Array of ListSolNetworkOperationsInfo objects

nextToken

The token to use to retrieve the next page of results. This value is `null` when there are no more results to return.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2

- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListSolNetworkPackages

Lists network packages.

A network package is a .zip file in CSAR (Cloud Service Archive) format defines the function packages you want to deploy and the AWS infrastructure you want to deploy them on.

Request Syntax

```
GET /sol/nsd/v1/ns_descriptors?max_results=maxResults&nextpage_opaque_marker=nextToken
 HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

maxResults

The maximum number of results to include in the response.

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

nextToken

The token for the next page of results.

Request Body

The request does not have a request body.

Response Syntax

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

{
   "networkPackages": [
      {
         "arn": "string",
         "id": "string",
```

```
            "metadata": {
               "createdAt": "string",
               "lastModified": "string"
            },
            "nsdDesigner": "string",
            "nsdId": "string",
            "nsdInvariantId": "string",
            "nsdName": "string",
            "nsdOnboardingState": "string",
            "nsdOperationalState": "string",
            "nsdUsageState": "string",
            "nsdVersion": "string",
            "vnfPkgIds": [ "string" ]
         }
      ],
      "nextToken": "string"
}
```

Response Elements

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

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

networkPackages

Network packages. A network package is a .zip file in CSAR (Cloud Service Archive) format defines the function packages you want to deploy and the AWS infrastructure you want to deploy them on.

Type: Array of ListSolNetworkPackageInfo objects

nextToken

The token to use to retrieve the next page of results. This value is `null` when there are no more results to return.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListTagsForResource

Lists tags for AWS TNB resources.

Request Syntax

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

URI Request Parameters

The request uses the following URI parameters.

resourceArn

Resource ARN.

Pattern: `arn:aws:tnb:[a-z0-9-]+:[^:]*:.*`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

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

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

Response Elements

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

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

tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Pattern: `(?!aws:).{1,128}`

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

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

PutSolFunctionPackageContent

Uploads the contents of a function package.

A function package is a .zip file in CSAR (Cloud Service Archive) format that contains a network function (an ETSI standard telecommunication application) and function package descriptor that uses the TOSCA standard to describe how the network functions should run on your network.

Request Syntax

```
PUT /sol/vnfpkgm/v1/vnf_packages/vnfPkgId/package_content HTTP/1.1
Content-Type: contentType

file
```

URI Request Parameters

The request uses the following URI parameters.

contentType

Function package content type.

Valid Values: `application/zip`

vnfPkgId

Function package ID.

Pattern: `fp-[a-f0-9]{17}`

Required: Yes

Request Body

The request accepts the following binary data.

file

Function package file.

Required: Yes

Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
   "id": "string",
   "metadata": {
      "vnfd": {
         "overrides": [
            {
               "defaultValue": "string",
               "name": "string"
            }
         ]
      }
   },
   "vnfdId": "string",
   "vnfdVersion": "string",
   "vnfProductName": "string",
   "vnfProvider": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 202 response.

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

id

Function package ID.

Type: String

Pattern: `fp-[a-f0-9]{17}`

metadata

Function package metadata.

Type: PutSolFunctionPackageContentMetadata object

vnfdId

Function package descriptor ID.

Type: String

Pattern: `[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}`

vnfdVersion

Function package descriptor version.

Type: String

vnfProductName

Function product name.

Type: String

vnfProvider

Function provider.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

PutSolNetworkPackageContent

Uploads the contents of a network package.

A network package is a .zip file in CSAR (Cloud Service Archive) format defines the function packages you want to deploy and the AWS infrastructure you want to deploy them on.

Request Syntax

```
PUT /sol/nsd/v1/ns_descriptors/nsdInfoId/nsd_content HTTP/1.1
Content-Type: contentType

file
```

URI Request Parameters

The request uses the following URI parameters.

contentType

Network package content type.

Valid Values: `application/zip`

nsdInfoId

Network service descriptor info ID.

Pattern: `np-[a-f0-9]{17}`

Required: Yes

Request Body

The request accepts the following binary data.

file

Network package file.

Required: Yes

Response Syntax

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

{
   "arn": "string",
   "id": "string",
   "metadata": {
      "nsd": {
         "overrides": [
            {
               "defaultValue": "string",
               "name": "string"
            }
         ]
      }
   },
   "nsdId": "string",
   "nsdName": "string",
   "nsdVersion": "string",
   "vnfPkgIds": [ "string" ]
}
```

Response Elements

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

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

arn

Network package ARN.

Type: String

Pattern: `arn:(aws|aws-cn|aws-iso|aws-iso-b|aws-us-gov):tnb:([a-z]{2}(-(gov|isob|iso))?-(east|west|north|south|central){1,2}-[0-9]):\d{12}:(network-package/np-[a-f0-9]{17})`

id

Network package ID.

Type: String

Pattern: `np-[a-f0-9]{17}`

metadata

Network package metadata.

Type: PutSolNetworkPackageContentMetadata object

nsdId

Network service descriptor ID.

Type: String

Pattern: `[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}`

nsdName

Network service descriptor name.

Type: String

nsdVersion

Network service descriptor version.

Type: String

vnfPkgIds

Function package IDs.

Type: Array of strings

Pattern: `fp-[a-f0-9]{17}`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python

- AWS SDK for Ruby V3

TagResource

Tags an AWS TNB resource.

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Request Syntax

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

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

URI Request Parameters

The request uses the following URI parameters.

resourceArn

Resource ARN.

Pattern: `arn:aws:tnb:[a-z0-9-]+:[^:]*:.*`

Required: Yes

Request Body

The request accepts the following data in JSON format.

tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Pattern: `(?!aws:).{1,128}`

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

Required: Yes

Response Syntax

```
HTTP/1.1 200
```

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

TerminateSolNetworkInstance

Terminates a network instance.

A network instance is a single network created in AWS TNB that can be deployed and on which life-cycle operations (like terminate, update, and delete) can be performed.

You must terminate a network instance before you can delete it.

Request Syntax

```
POST /sol/nslcm/v1/ns_instances/nsInstanceId/terminate HTTP/1.1
Content-type: application/json

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

URI Request Parameters

The request uses the following URI parameters.

nsInstanceId

ID of the network instance.

Pattern: `ni-[a-f0-9]{17}`

Required: Yes

Request Body

The request accepts the following data in JSON format.

tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. When you use this API, the tags are only applied to the network operation that is created.

These tags are not applied to the network instance. Use tags to search and filter your resources or track your AWS costs.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Pattern: `(?!aws:).{1,128}`

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

Required: No

Response Syntax

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

{
   "nsLcmOpOccId": "string",
   "tags": {
      "string" : "string"
   }
}
```

Response Elements

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

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

nsLcmOpOccId

The identifier of the network operation.

Type: String

Pattern: `no-[a-f0-9]{17}`

tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. When you use this API, the tags are only applied to the network operation that is created.

These tags are not applied to the network instance. Use tags to search and filter your resources or track your AWS costs.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Pattern: `(?!aws:).{1,128}`

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

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ServiceQuotaExceededException

Service quotas have been exceeded.

HTTP Status Code: 402

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UntagResource

Untags an AWS TNB resource.

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Request Syntax

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

URI Request Parameters

The request uses the following URI parameters.

resourceArn

Resource ARN.

Pattern: `arn:aws:tnb:[a-z0-9-]+:[^:]*:.*`

Required: Yes

tagKeys

Tag keys.

Array Members: Minimum number of 0 items. Maximum number of 200 items.

Pattern: `(?!aws:).{1,128}`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
```

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateSolFunctionPackage

Updates the operational state of function package.

A function package is a .zip file in CSAR (Cloud Service Archive) format that contains a network function (an ETSI standard telecommunication application) and function package descriptor that uses the TOSCA standard to describe how the network functions should run on your network.

Request Syntax

```
PATCH /sol/vnfpkgm/v1/vnf_packages/vnfPkgId HTTP/1.1
Content-type: application/json

{
   "operationalState": "string"
}
```

URI Request Parameters

The request uses the following URI parameters.

vnfPkgId

ID of the function package.

Pattern: `fp-[a-f0-9]{17}`

Required: Yes

Request Body

The request accepts the following data in JSON format.

operationalState

Operational state of the function package.

Type: String

Valid Values: `ENABLED | DISABLED`

Required: Yes

Response Syntax

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

{
   "operationalState": "string"
}
```

Response Elements

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

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

operationalState

Operational state of the function package.

Type: String

Valid Values: `ENABLED | DISABLED`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateSolNetworkInstance

Update a network instance.

A network instance is a single network created in AWS TNB that can be deployed and on which life-cycle operations (like terminate, update, and delete) can be performed.

Choose the *updateType* parameter to target the necessary update of the network instance.

Request Syntax

```
POST /sol/nslcm/v1/ns_instances/nsInstanceId/update HTTP/1.1
Content-type: application/json

{
   "modifyVnfInfoData": {
      "vnfConfigurableProperties": JSON value,
      "vnfInstanceId": "string"
   },
   "tags": {
      "string" : "string"
   },
   "updateNs": {
      "additionalParamsForNs": JSON value,
      "nsdInfoId": "string"
   },
   "updateType": "string"
}
```

URI Request Parameters

The request uses the following URI parameters.

nsInstanceId

ID of the network instance.

Pattern: `ni-[a-f0-9]{17}`

Required: Yes

Request Body

The request accepts the following data in JSON format.

modifyVnfInfoData

Identifies the network function information parameters and/or the configurable properties of the network function to be modified.

Include this property only if the update type is `MODIFY_VNF_INFORMATION`.

Type: UpdateSolNetworkModify object

Required: No

tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. When you use this API, the tags are only applied to the network operation that is created. These tags are not applied to the network instance. Use tags to search and filter your resources or track your AWS costs.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Pattern: `(?!aws:).{1,128}`

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

Required: No

updateNs

Identifies the network service descriptor and the configurable properties of the descriptor, to be used for the update.

Include this property only if the update type is `UPDATE_NS`.

Type: UpdateSolNetworkServiceData object

Required: No

updateType

The type of update.

- Use the MODIFY_VNF_INFORMATION update type, to update a specific network function configuration, in the network instance.
- Use the UPDATE_NS update type, to update the network instance to a new network service descriptor.

Type: String

Valid Values: MODIFY_VNF_INFORMATION | UPDATE_NS

Required: Yes

Response Syntax

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

{
   "nsLcmOpOccId": "string",
   "tags": {
      "string" : "string"
   }
}
```

Response Elements

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

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

nsLcmOpOccId

The identifier of the network operation.

Type: String

Pattern: no-[a-f0-9]{17}

tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. When you use this API, the tags are only applied to the network operation that is created.

These tags are not applied to the network instance. Use tags to search and filter your resources or track your AWS costs.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Pattern: `(?!aws:).{1,128}`

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

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ServiceQuotaExceededException

Service quotas have been exceeded.

HTTP Status Code: 402

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateSolNetworkPackage

Updates the operational state of a network package.

A network package is a .zip file in CSAR (Cloud Service Archive) format defines the function packages you want to deploy and the AWS infrastructure you want to deploy them on.

A network service descriptor is a .yaml file in a network package that uses the TOSCA standard to describe the network functions you want to deploy and the AWS infrastructure you want to deploy the network functions on.

Request Syntax

```
PATCH /sol/nsd/v1/ns_descriptors/nsdInfoId HTTP/1.1
Content-type: application/json

{
   "nsdOperationalState": "string"
}
```

URI Request Parameters

The request uses the following URI parameters.

nsdInfoId

ID of the network service descriptor in the network package.

Pattern: `np-[a-f0-9]{17}`

Required: Yes

Request Body

The request accepts the following data in JSON format.

nsdOperationalState

Operational state of the network service descriptor in the network package.

Type: String

Valid Values: ENABLED | DISABLED

Required: Yes

Response Syntax

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

{
   "nsdOperationalState": "string"
}
```

Response Elements

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

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

nsdOperationalState

Operational state of the network service descriptor in the network package.

Type: String

Valid Values: ENABLED | DISABLED

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ValidateSolFunctionPackageContent

Validates function package content. This can be used as a dry run before uploading function package content with PutSolFunctionPackageContent.

A function package is a .zip file in CSAR (Cloud Service Archive) format that contains a network function (an ETSI standard telecommunication application) and function package descriptor that uses the TOSCA standard to describe how the network functions should run on your network.

Request Syntax

```
PUT /sol/vnfpkgm/v1/vnf_packages/vnfPkgId/package_content/validate HTTP/1.1
Content-Type: contentType

file
```

URI Request Parameters

The request uses the following URI parameters.

contentType

Function package content type.

Valid Values: `application/zip`

vnfPkgId

Function package ID.

Pattern: `fp-[a-f0-9]{17}`

Required: Yes

Request Body

The request accepts the following binary data.

file

Function package file.

Required: Yes

Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
   "id": "string",
   "metadata": {
      "vnfd": {
         "overrides": [
            {
               "defaultValue": "string",
               "name": "string"
            }
         ]
      }
   },
   "vnfdId": "string",
   "vnfdVersion": "string",
   "vnfProductName": "string",
   "vnfProvider": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 202 response.

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

id

Function package ID.

Type: String

Pattern: `fp-[a-f0-9]{17}`

metadata

Function package metadata.

Type: ValidateSolFunctionPackageContentMetadata object

vnfdId

Function package descriptor ID.

Type: String

Pattern: `[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}`

vnfdVersion

Function package descriptor version.

Type: String

vnfProductName

Network function product name.

Type: String

vnfProvider

Network function provider.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ValidateSolNetworkPackageContent

Validates network package content. This can be used as a dry run before uploading network package content with PutSolNetworkPackageContent.

A network package is a .zip file in CSAR (Cloud Service Archive) format defines the function packages you want to deploy and the AWS infrastructure you want to deploy them on.

Request Syntax

```
PUT /sol/nsd/v1/ns_descriptors/nsdInfoId/nsd_content/validate HTTP/1.1
Content-Type: contentType

file
```

URI Request Parameters

The request uses the following URI parameters.

contentType

Network package content type.

Valid Values: `application/zip`

nsdInfoId

Network service descriptor file.

Pattern: `np-[a-f0-9]{17}`

Required: Yes

Request Body

The request accepts the following binary data.

file

Network package file.

Required: Yes

Response Syntax

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

{
   "arn": "string",
   "id": "string",
   "metadata": {
      "nsd": {
         "overrides": [
            {
               "defaultValue": "string",
               "name": "string"
            }
         ]
      }
   },
   "nsdId": "string",
   "nsdName": "string",
   "nsdVersion": "string",
   "vnfPkgIds": [ "string" ]
}
```

Response Elements

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

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

arn

Network package ARN.

Type: String

Pattern: `arn:(aws|aws-cn|aws-iso|aws-iso-b|aws-us-gov):tnb:([a-z]{2}(-(gov|isob|iso))?-(east|west|north|south|central){1,2}-[0-9]):\d{12}:(network-package/np-[a-f0-9]{17})`

id

Network package ID.

Type: String

Pattern: `np-[a-f0-9]{17}`

metadata

Network package metadata.

Type: ValidateSolNetworkPackageContentMetadata object

nsdId

Network service descriptor ID.

Type: String

Pattern: `[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}`

nsdName

Network service descriptor name.

Type: String

nsdVersion

Network service descriptor version.

Type: String

vnfPkgIds

Function package IDs.

Type: Array of strings

Pattern: `fp-[a-f0-9]{17}`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Insufficient permissions to make request.

HTTP Status Code: 403

InternalServerException

Unexpected error occurred. Problem on the server.

HTTP Status Code: 500

ResourceNotFoundException

Request references a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

Exception caused by throttling.

HTTP Status Code: 429

ValidationException

Unable to process the request because the client provided input failed to satisfy request constraints.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3

- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

Data Types

The AWS Telco Network Builder API contains several data types that various actions use. This section describes each data type in detail.

> **Note**
>
> The order of each element in a data type structure is not guaranteed. Applications should not assume a particular order.

The following data types are supported:

- ErrorInfo
- FunctionArtifactMeta
- GetSolFunctionInstanceMetadata
- GetSolFunctionPackageMetadata
- GetSolInstantiatedVnfInfo
- GetSolNetworkInstanceMetadata
- GetSolNetworkOperationMetadata
- GetSolNetworkOperationTaskDetails
- GetSolNetworkPackageMetadata
- GetSolVnfcResourceInfo
- GetSolVnfcResourceInfoMetadata
- GetSolVnfInfo
- InstantiateMetadata
- LcmOperationInfo
- ListSolFunctionInstanceInfo
- ListSolFunctionInstanceMetadata
- ListSolFunctionPackageInfo
- ListSolFunctionPackageMetadata
- ListSolNetworkInstanceInfo
- ListSolNetworkInstanceMetadata

- ListSolNetworkOperationsInfo
- ListSolNetworkOperationsMetadata
- ListSolNetworkPackageInfo
- ListSolNetworkPackageMetadata
- ModifyVnfInfoMetadata
- NetworkArtifactMeta
- ProblemDetails
- PutSolFunctionPackageContentMetadata
- PutSolNetworkPackageContentMetadata
- ToscaOverride
- UpdateNsMetadata
- UpdateSolNetworkModify
- UpdateSolNetworkServiceData
- ValidateSolFunctionPackageContentMetadata
- ValidateSolNetworkPackageContentMetadata

ErrorInfo

Provides error information.

Contents

cause

Error cause.

Type: String

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

Required: No

details

Error details.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 10240.

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FunctionArtifactMeta

Metadata for function package artifacts.

Artifacts are the contents of the package descriptor file and the state of the package.

Contents

overrides

Lists of function package overrides.

Type: Array of ToscaOverride objects

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GetSolFunctionInstanceMetadata

The metadata of a network function instance.

A network function instance is a function in a function package .

Contents

createdAt

The date that the resource was created.

Type: Timestamp

Required: Yes

lastModified

The date that the resource was last modified.

Type: Timestamp

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GetSolFunctionPackageMetadata

Metadata related to the function package.

A function package is a .zip file in CSAR (Cloud Service Archive) format that contains a network function (an ETSI standard telecommunication application) and function package descriptor that uses the TOSCA standard to describe how the network functions should run on your network.

Contents

createdAt

The date that the resource was created.

Type: Timestamp

Required: Yes

lastModified

The date that the resource was last modified.

Type: Timestamp

Required: Yes

vnfd

Metadata related to the function package descriptor of the function package.

Type: FunctionArtifactMeta object

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GetSolInstantiatedVnfInfo

Information about a network function.

A network instance is a single network created in AWS TNB that can be deployed and on which life-cycle operations (like terminate, update, and delete) can be performed.

Contents

vnfState

State of the network function.

Type: String

Valid Values: `STARTED | STOPPED`

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GetSolNetworkInstanceMetadata

The metadata of a network instance.

A network instance is a single network created in AWS TNB that can be deployed and on which life-cycle operations (like terminate, update, and delete) can be performed.

Contents

createdAt

The date that the resource was created.

Type: Timestamp

Required: Yes

lastModified

The date that the resource was last modified.

Type: Timestamp

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GetSolNetworkOperationMetadata

Metadata related to a network operation occurrence.

A network operation is any operation that is done to your network, such as network instance instantiation or termination.

Contents

createdAt

The date that the resource was created.

Type: Timestamp

Required: Yes

lastModified

The date that the resource was last modified.

Type: Timestamp

Required: Yes

instantiateMetadata

Metadata related to the network operation occurrence for network instantiation. This is populated only if the lcmOperationType is `INSTANTIATE`.

Type: InstantiateMetadata object

Required: No

modifyVnfInfoMetadata

Metadata related to the network operation occurrence for network function updates in a network instance. This is populated only if the lcmOperationType is `UPDATE` and the updateType is `MODIFY_VNF_INFORMATION`.

Type: ModifyVnfInfoMetadata object

Required: No

updateNsMetadata

Metadata related to the network operation occurrence for network instance updates. This is populated only if the lcmOperationType is UPDATE and the updateType is UPDATE_NS.

Type: UpdateNsMetadata object

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GetSolNetworkOperationTaskDetails

Gets the details of a network operation.

A network operation is any operation that is done to your network, such as network instance instantiation or termination.

Contents

taskContext

Context for the network operation task.

Type: String to string map

Required: No

taskEndTime

Task end time.

Type: Timestamp

Required: No

taskErrorDetails

Task error details.

Type: ErrorInfo object

Required: No

taskName

Task name.

Type: String

Required: No

taskStartTime

Task start time.

Type: Timestamp

Required: No

taskStatus

Task status.

Type: String

Valid Values: `SCHEDULED | STARTED | IN_PROGRESS | COMPLETED | ERROR | SKIPPED | CANCELLED`

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GetSolNetworkPackageMetadata

Metadata associated with a network package.

A network package is a .zip file in CSAR (Cloud Service Archive) format defines the function packages you want to deploy and the AWS infrastructure you want to deploy them on.

Contents

createdAt

The date that the resource was created.

Type: Timestamp

Required: Yes

lastModified

The date that the resource was last modified.

Type: Timestamp

Required: Yes

nsd

Metadata related to the onboarded network service descriptor in the network package.

Type: NetworkArtifactMeta object

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GetSolVnfcResourceInfo

Details of resource associated with a network function.

A network instance is a single network created in AWS TNB that can be deployed and on which life-cycle operations (like terminate, update, and delete) can be performed.

Contents

metadata

The metadata of the network function compute.

Type: GetSolVnfcResourceInfoMetadata object

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GetSolVnfcResourceInfoMetadata

The metadata of a network function.

A network instance is a single network created in AWS TNB that can be deployed and on which life-cycle operations (like terminate, update, and delete) can be performed.

Contents

cluster

Information about the cluster.

Type: String

Required: No

helmChart

Information about the helm chart.

Type: String

Required: No

nodeGroup

Information about the node group.

Type: String

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GetSolVnfInfo

Information about the network function.

A network function instance is a function in a function package .

Contents

vnfcResourceInfo

Compute info used by the network function instance.

Type: Array of GetSolVnfcResourceInfo objects

Required: No

vnfState

State of the network function instance.

Type: String

Valid Values: `STARTED | STOPPED`

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

InstantiateMetadata

Metadata related to the configuration properties used during instantiation of the network instance.

Contents

nsdInfoId

The network service descriptor used for instantiating the network instance.

Type: String

Pattern: `np-[a-f0-9]{17}`

Required: Yes

additionalParamsForNs

The configurable properties used during instantiation.

Type: JSON value

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LcmOperationInfo

Lifecycle management operation details on the network instance.

Lifecycle management operations are deploy, update, or delete operations.

Contents

nsLcmOpOccId

The identifier of the network operation.

Type: String

Pattern: `no-[a-f0-9]{17}`

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ListSolFunctionInstanceInfo

Lists information about a network function instance.

A network function instance is a function in a function package .

Contents

arn

Network function instance ARN.

Type: String

Pattern: `arn:(aws|aws-cn|aws-iso|aws-iso-b|aws-us-gov):tnb:([a-z]{2}(-(gov|isob|iso))?-(east|west|north|south|central){1,2}-[0-9]):\d{12}:(function-instance/fi-[a-f0-9]{17})`

Required: Yes

id

Network function instance ID.

Type: String

Pattern: `fi-[a-f0-9]{17}`

Required: Yes

instantiationState

Network function instance instantiation state.

Type: String

Valid Values: `INSTANTIATED | NOT_INSTANTIATED`

Required: Yes

metadata

Network function instance metadata.

Type: ListSolFunctionInstanceMetadata object

Required: Yes

nsInstanceId

Network instance ID.

Type: String

Pattern: `ni-[a-f0-9]{17}`

Required: Yes

vnfPkgId

Function package ID.

Type: String

Pattern: `fp-[a-f0-9]{17}`

Required: Yes

instantiatedVnfInfo

Information about a network function.

A network instance is a single network created in AWS TNB that can be deployed and on which life-cycle operations (like terminate, update, and delete) can be performed.

Type: GetSolInstantiatedVnfInfo object

Required: No

vnfPkgName

Function package name.

Type: String

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ListSolFunctionInstanceMetadata

Lists network function instance metadata.

A network function instance is a function in a function package .

Contents

createdAt

When the network function instance was created.

Type: Timestamp

Required: Yes

lastModified

When the network function instance was last modified.

Type: Timestamp

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ListSolFunctionPackageInfo

Information about a function package.

A function package is a .zip file in CSAR (Cloud Service Archive) format that contains a network function (an ETSI standard telecommunication application) and function package descriptor that uses the TOSCA standard to describe how the network functions should run on your network.

Contents

arn

Function package ARN.

Type: String

Pattern: `arn:(aws|aws-cn|aws-iso|aws-iso-b|aws-us-gov):tnb:([a-z]{2}(-(gov|isob|iso))?-(east|west|north|south|central){1,2}-[0-9]):\d{12}:(function-package/fp-[a-f0-9]{17})`

Required: Yes

id

ID of the function package.

Type: String

Pattern: `fp-[a-f0-9]{17}`

Required: Yes

onboardingState

Onboarding state of the function package.

Type: String

Valid Values: `CREATED | ONBOARDED | ERROR`

Required: Yes

operationalState

Operational state of the function package.

Type: String

Valid Values: `ENABLED | DISABLED`

Required: Yes

usageState

Usage state of the function package.

Type: String

Valid Values: `IN_USE | NOT_IN_USE`

Required: Yes

metadata

The metadata of the function package.

Type: ListSolFunctionPackageMetadata object

Required: No

vnfdId

Identifies the function package and the function package descriptor.

Type: String

Required: No

vnfdVersion

Identifies the version of the function package descriptor.

Type: String

Required: No

vnfProductName

The product name for the network function.

Type: String

Required: No

vnfProvider

Provider of the function package and the function package descriptor.

Type: String

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ListSolFunctionPackageMetadata

Details for the function package metadata.

A function package is a .zip file in CSAR (Cloud Service Archive) format that contains a network function (an ETSI standard telecommunication application) and function package descriptor that uses the TOSCA standard to describe how the network functions should run on your network.

Contents

createdAt

The date that the resource was created.

Type: Timestamp

Required: Yes

lastModified

The date that the resource was last modified.

Type: Timestamp

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ListSolNetworkInstanceInfo

Info about the specific network instance.

A network instance is a single network created in AWS TNB that can be deployed and on which life-cycle operations (like terminate, update, and delete) can be performed.

Contents

arn

Network instance ARN.

Type: String

Pattern: `arn:(aws|aws-cn|aws-iso|aws-iso-b|aws-us-gov):tnb:([a-z]{2}(-(gov|isob|iso))?-(east|west|north|south|central){1,2}-[0-9]):\d{12}:(network-instance/ni-[a-f0-9]{17})`

Required: Yes

id

ID of the network instance.

Type: String

Pattern: `ni-[a-f0-9]{17}`

Required: Yes

metadata

The metadata of the network instance.

Type: ListSolNetworkInstanceMetadata object

Required: Yes

nsdId

ID of the network service descriptor in the network package.

Type: String

Pattern: `[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}`

Required: Yes

nsdInfoId

ID of the network service descriptor in the network package.

Type: String

Pattern: `np-[a-f0-9]{17}`

Required: Yes

nsInstanceDescription

Human-readable description of the network instance.

Type: String

Required: Yes

nsInstanceName

Human-readable name of the network instance.

Type: String

Required: Yes

nsState

The state of the network instance.

Type: String

Valid Values: `INSTANTIATED | NOT_INSTANTIATED | UPDATED | IMPAIRED | UPDATE_FAILED | STOPPED | DELETED | INSTANTIATE_IN_PROGRESS | INTENT_TO_UPDATE_IN_PROGRESS | UPDATE_IN_PROGRESS | TERMINATE_IN_PROGRESS`

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ListSolNetworkInstanceMetadata

Metadata details for a network instance.

A network instance is a single network created in AWS TNB that can be deployed and on which life-cycle operations (like terminate, update, and delete) can be performed.

Contents

createdAt

The date that the resource was created.

Type: Timestamp

Required: Yes

lastModified

The date that the resource was last modified.

Type: Timestamp

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ListSolNetworkOperationsInfo

Information parameters for a network operation.

Contents

arn

Network operation ARN.

Type: String

Pattern: `arn:(aws|aws-cn|aws-iso|aws-iso-b|aws-us-gov):tnb:([a-z]{2}(-(gov|isob|iso))?-(east|west|north|south|central){1,2}-[0-9]):\d{12}:(network-operation/no-[a-f0-9]{17})`

Required: Yes

id

ID of this network operation.

Type: String

Pattern: `no-[a-f0-9]{17}`

Required: Yes

lcmOperationType

Type of lifecycle management network operation.

Type: String

Valid Values: `INSTANTIATE | UPDATE | TERMINATE`

Required: Yes

nsInstanceId

ID of the network instance related to this operation.

Type: String

Pattern: `ni-[a-f0-9]{17}`

Required: Yes

operationState

The state of the network operation.

Type: String

Valid Values: `PROCESSING | COMPLETED | FAILED | CANCELLING | CANCELLED`

Required: Yes

error

Error related to this specific network operation.

Type: ProblemDetails object

Required: No

metadata

Metadata related to this network operation.

Type: ListSolNetworkOperationsMetadata object

Required: No

updateType

Type of the update. Only present if the network operation lcmOperationType is `UPDATE`.

Type: String

Valid Values: `MODIFY_VNF_INFORMATION | UPDATE_NS`

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2

- AWS SDK for Ruby V3

ListSolNetworkOperationsMetadata

Metadata related to a network operation.

A network operation is any operation that is done to your network, such as network instance instantiation or termination.

Contents

createdAt

The date that the resource was created.

Type: Timestamp

Required: Yes

lastModified

The date that the resource was last modified.

Type: Timestamp

Required: Yes

nsdInfoId

The network service descriptor id used for the operation.

Only present if the updateType is `UPDATE_NS`.

Type: String

Pattern: `np-[a-f0-9]{17}`

Required: No

vnfInstanceId

The network function id used for the operation.

Only present if the updateType is `MODIFY_VNF_INFO`.

Type: String

Pattern: `fi-[a-f0-9]{17}`

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ListSolNetworkPackageInfo

Details of a network package.

A network package is a .zip file in CSAR (Cloud Service Archive) format defines the function packages you want to deploy and the AWS infrastructure you want to deploy them on.

Contents

arn

Network package ARN.

Type: String

Pattern: `arn:(aws|aws-cn|aws-iso|aws-iso-b|aws-us-gov):tnb:([a-z]{2}(-(gov|isob|iso))?-(east|west|north|south|central){1,2}-[0-9]):\d{12}:(network-package/np-[a-f0-9]{17})`

Required: Yes

id

ID of the individual network package.

Type: String

Pattern: `np-[a-f0-9]{17}`

Required: Yes

metadata

The metadata of the network package.

Type: ListSolNetworkPackageMetadata object

Required: Yes

nsdOnboardingState

Onboarding state of the network service descriptor in the network package.

Type: String

Valid Values: `CREATED | ONBOARDED | ERROR`

Required: Yes

nsdOperationalState

Operational state of the network service descriptor in the network package.

Type: String

Valid Values: `ENABLED | DISABLED`

Required: Yes

nsdUsageState

Usage state of the network service descriptor in the network package.

Type: String

Valid Values: `IN_USE | NOT_IN_USE`

Required: Yes

nsdDesigner

Designer of the onboarded network service descriptor in the network package.

Type: String

Required: No

nsdId

ID of the network service descriptor on which the network package is based.

Type: String

Required: No

nsdInvariantId

Identifies a network service descriptor in a version independent manner.

Type: String

Required: No

nsdName

Name of the onboarded network service descriptor in the network package.

Type: String

Required: No

nsdVersion

Version of the onboarded network service descriptor in the network package.

Type: String

Required: No

vnfPkgIds

Identifies the function package for the function package descriptor referenced by the onboarded network package.

Type: Array of strings

Pattern: `fp-[a-f0-9]{17}`

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ListSolNetworkPackageMetadata

Metadata related to a network package.

A network package is a .zip file in CSAR (Cloud Service Archive) format defines the function packages you want to deploy and the AWS infrastructure you want to deploy them on.

Contents

createdAt

The date that the resource was created.

Type: Timestamp

Required: Yes

lastModified

The date that the resource was last modified.

Type: Timestamp

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ModifyVnfInfoMetadata

Metadata related to the configuration properties used during update of a specific network function in a network instance.

Contents

vnfConfigurableProperties

The configurable properties used during update of the network function instance.

Type: JSON value

Required: Yes

vnfInstanceId

The network function instance that was updated in the network instance.

Type: String

Pattern: `fi-[a-f0-9]{17}`

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

NetworkArtifactMeta

Metadata for network package artifacts.

Artifacts are the contents of the package descriptor file and the state of the package.

Contents

overrides

Lists network package overrides.

Type: Array of ToscaOverride objects

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ProblemDetails

Details related to problems with AWS TNB resources.

Contents

detail

A human-readable explanation specific to this occurrence of the problem.

Type: String

Required: Yes

title

A human-readable title of the problem type.

Type: String

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PutSolFunctionPackageContentMetadata

Update metadata in a function package.

A function package is a .zip file in CSAR (Cloud Service Archive) format that contains a network function (an ETSI standard telecommunication application) and function package descriptor that uses the TOSCA standard to describe how the network functions should run on your network.

Contents

vnfd

Metadata for function package artifacts.

Artifacts are the contents of the package descriptor file and the state of the package.

Type: FunctionArtifactMeta object

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PutSolNetworkPackageContentMetadata

Update metadata in a network package.

A network package is a .zip file in CSAR (Cloud Service Archive) format defines the function packages you want to deploy and the AWS infrastructure you want to deploy them on.

Contents

nsd

Metadata for network package artifacts.

Artifacts are the contents of the package descriptor file and the state of the package.

Type: NetworkArtifactMeta object

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ToscaOverride

Overrides of the TOSCA node.

Contents

defaultValue

Default value for the override.

Type: String

Required: No

name

Name of the TOSCA override.

Type: String

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

UpdateNsMetadata

Metadata related to the configuration properties used during update of a network instance.

Contents

nsdInfoId

The network service descriptor used for updating the network instance.

Type: String

Pattern: `np-[a-f0-9]{17}`

Required: Yes

additionalParamsForNs

The configurable properties used during update.

Type: JSON value

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

UpdateSolNetworkModify

Information parameters and/or the configurable properties for a network function.

A network function instance is a function in a function package .

Contents

vnfConfigurableProperties

Provides values for the configurable properties declared in the function package descriptor.

Type: JSON value

Required: Yes

vnfInstanceId

ID of the network function instance.

A network function instance is a function in a function package .

Type: String

Pattern: `fi-[a-f0-9]{17}`

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

UpdateSolNetworkServiceData

Information parameters and/or the configurable properties for a network descriptor used for update.

Contents

nsdInfoId

ID of the network service descriptor.

Type: String

Pattern: `np-[a-f0-9]{17}`

Required: Yes

additionalParamsForNs

Values for the configurable properties declared in the network service descriptor.

Type: JSON value

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ValidateSolFunctionPackageContentMetadata

Validates function package content metadata.

A function package is a .zip file in CSAR (Cloud Service Archive) format that contains a network function (an ETSI standard telecommunication application) and function package descriptor that uses the TOSCA standard to describe how the network functions should run on your network.

Contents

vnfd

Metadata for function package artifacts.

Artifacts are the contents of the package descriptor file and the state of the package.

Type: FunctionArtifactMeta object

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ValidateSolNetworkPackageContentMetadata

Validates network package content metadata.

A network package is a .zip file in CSAR (Cloud Service Archive) format defines the function packages you want to deploy and the AWS infrastructure you want to deploy them on.

Contents

nsd

Metadata for network package artifacts.

Artifacts are the contents of the package descriptor file and the state of the package.

Type: NetworkArtifactMeta object

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Common Parameters

The following list contains the parameters that all actions use for signing Signature Version 4 requests with a query string. Any action-specific parameters are listed in the topic for that action. For more information about Signature Version 4, see Signing AWS API requests in the *IAM User Guide*.

Action

The action to be performed.

Type: string

Required: Yes

Version

The API version that the request is written for, expressed in the format YYYY-MM-DD.

Type: string

Required: Yes

X-Amz-Algorithm

The hash algorithm that you used to create the request signature.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Valid Values: `AWS4-HMAC-SHA256`

Required: Conditional

X-Amz-Credential

The credential scope value, which is a string that includes your access key, the date, the region you are targeting, the service you are requesting, and a termination string ("aws4_request"). The value is expressed in the following format: *access_key*/*YYYYMMDD*/*region*/*service*/aws4_request.

For more information, see Create a signed AWS API request in the *IAM User Guide*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-Date

The date that is used to create the signature. The format must be ISO 8601 basic format (YYYYMMDD'T'HHMMSS'Z'). For example, the following date time is a valid X-Amz-Date value: `20120325T120000Z`.

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see Elements of an AWS API request signature in the *IAM User Guide*.

Type: string

Required: Conditional

X-Amz-Security-Token

The temporary security token that was obtained through a call to AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS STS, see AWS services that work with IAM in the *IAM User Guide*.

Condition: If you're using temporary security credentials from AWS STS, you must include the security token.

Type: string

Required: Conditional

X-Amz-Signature

Specifies the hex-encoded signature that was calculated from the string to sign and the derived signing key.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-SignedHeaders

Specifies all the HTTP headers that were included as part of the canonical request. For more information about specifying signed headers, see Create a signed AWS API request in the *IAM User Guide*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 403

ExpiredTokenException

The security token included in the request is expired

HTTP Status Code: 403

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 403

InternalFailure

The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

MalformedHttpRequestException

Problems with the request at the HTTP level, e.g. we can't decompress the body according to the decompression algorithm specified by the content-encoding.

HTTP Status Code: 400

NotAuthorized

You do not have permission to perform this action.

HTTP Status Code: 401

OptInRequired

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

RequestAbortedException

Convenient exception that can be used when a request is aborted before a reply is sent back (e.g. client closed connection).

HTTP Status Code: 400

RequestEntityTooLargeException

Problems with the request at the HTTP level. The request entity is too large.

HTTP Status Code: 413

RequestExpired

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

RequestTimeoutException

Problems with the request at the HTTP level. Reading the Request timed out.

HTTP Status Code: 408

ServiceUnavailable

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 400

UnrecognizedClientException

The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

UnknownOperationException

The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 404

ValidationError

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400