

NTT Communications

Cloudⁿ

Provisioning API Manual

Ver.1.0

Please refrain from redistributing (including distributing, copying, and providing) the content of this booklet.

Version	Date of Compilation	Revision
Ver.1.0	April 9, 2013	First edition created

1. Making an API Request	Page 4 –
1) Format of an API Request	
2) Creating a Request	
3) Checking a Response	
2 Using a Cloudⁿ Provisioning Service	Page 9 –
1) Setting Up a Stack	
2) Checking a Stack/Template	
3 Cloudⁿ Provisioning API Reference	Page 13 –
1) Provisioning API List	
2) Provisioning Common Information	
3) Provisioning API	

1-1) Format of an API Request

This service provides an API for adding and deleting settings of the Provisioning service.

When you use the API, you can directly handle the resources from your program. In addition, the API is compatible with the Amazon Web Service Cloud compatible API (2010-05-15). To use the service, you need to have an access to the API Server (endpoint) defined by the URL shown below.

API server(endpoint) URL :

<https://prov-api.jp-e1.cloudn-service.com/Provision/queryapi/>

[Format of an API Request]

You can send an API request in the Query API format as shown below:

```
https://prov-api.jp-e1.cloudn-  
service.com/Provision/queryapi/?Action=DescribeStacks&Version=2010-05-  
15&SignatureVersion=2&SignatureMethod=HmacSHA256&Timestamp=2013-02-  
01T05%3A54%3A53.578Z&AWSAccessKeyId=<APIKey>&Signature=<Signature>
```

In the example above, the request stated here is trying to acquire the metric information (monitoring/measuring items) supported by the Cloudⁿ Provisioning service.

The major components of an API request include a command type and its optional

- ar
1. <https://prov-api.jp-e1.cloudn-service.com/Provision/queryapi/>
 2. Action=DescribeStacks
 3. Version=2010-05-15
 4. SignatureVersion=2
 5. SignatureMethod=HmacSHA256
 6. Timestamp=2013-02-01T05%3A54%3A53.578Z
 7. AWSAccessKeyId=<APIKey>
 8. Signature=<Signature>

Line 1: API server/endpoint URL

Line 2: Command to Cloudⁿ Provisioning

Line 3: Command option and argument

Lines 4 - 8: Signature information

The following sections describe how you add a signature to the statement of a request.

1-2) Creating a Request

You should add a signature to an API request to secure the content of a request. To create a signature, you need to combine a user's secret key with the HMAC-SHA-256 hash algorithm on the basis of the statement of a request (created based on the elements in 1).

Public and secret keys necessary to use the service are distributed in advance. These key are, in our service, called APIKEY and SECRETKEY, respectively. You must use those APIKEY and SECRETKEY that are distributed in advance.

This section describes how you generate a signature and an HTTP request.

1

Define the command parameters of a request. Shown below is an example of a request to acquire the stack (a stack bundles different settings as one) information registered to the Cloudⁿ Provisioning service.

Command (Parameter) = key	Value (Sample) = value
Action	DescribeStacks
Version	2010-05-15
SignatureVersion	2
SignatureMethod	HmacSHA256
Timestamp	2013-01-30T18%3A09%3A45Z
AWSAccessKeyId	<APIKEY>



Upper case letters and lower case letters are distinguished in the keys you define.



The Timestamp key creates the time of request issuance in the iso8601 format.



For the details of the parameters, see section 3, Cloudⁿ Provisioning API Reference.

1-2) Creating a Request

After the previous step, create a signature.

2

Sort the command parameters you have created in 1 in the ASCII-based ascending order to URL-encode the definitions. This step is required to create a signature. The statements in the request are not sorted. (You do not need to sort the statements in the request.)

Command (Parameter) = key	Value (Sample) = value
AWSAccessKeyId	<APIKEY>
Action	DescribeStacks
SignatureMethod	HmacSHA256
SignatureVersion	2
Timestamp	2013-01-30T18%3A09%3A45Z
Version	2010-05-15



Note that the keys are sorted in the ASCII-based ascending order but not in the alphabetical order.

3

Use an equal sign (=) to connect a corresponding key and value in 2. Use an ampersand sign (&) to join adjacent character strings. Add the elements of the HTTP request to create a character string, to which you are going to add an **signature**. The character string shown below is called "data" from here on.

GET prov-api.jp-e1.cloudn-service.com

/Provision/queryapi/

AWSAccessKeyId=<APIKey>&Action=DescribeStacks&SignatureMethod=HmacSHA256&SignatureVersion=2&Timestamp=2013-01-30T18%3A09%3A45Z&Version=2010-05-15



Press the Return key to separate lines until the "AWSAccessKeyId=..." query begins (as shown with a triangle (◀) in the example above. Do not separate the line of query, but create it as one line.

1-2) Creating a Request

4

For character string "data" created in 3, create a signature from HMAC-SHA256 and SECRETKEY. Encode them in Base 64 to include them in an HTTP request.

HMAC-SHA256:

Use a function in the library such as OpenSSL.
(Example: "ruby-hmac(0.4.0)" in the gem library in the case of Ruby.)

SECRETKEY:

Use the SECRETKEY we provide you with.

An example of a signature based on HMAC:

5df60c66d6715d33c5b49af3428c0cbb84918a0baa96c29f3b32670a742bdc29

An example of a signature: (encoded in base64)

XfYmZtZxXTPFtJrzQowMu4SRiguqlsKfOzJnCnQr3Ck=



Use caution not to separate a signature by the Return key.

5

Add the signature to the statement of a request, and create a character string to form a request. Encode the arguments of the parameters in advance. Connect commands and parameters in the form of key=value (make sure to URL-encode "value" in advance), and use an ampersand sign (&) to join adjacent character strings. You do not need to sort the elements in an HTTP request.

```
Action=DescribeStacks&SignatureMethod=HmacSHA256&SignatureVersion=2&AWSAccessKeyId=<APIKEY>&Version=2010-05-15&Timestamp=2013-01-30T18%3A09%3A45Z&Signature=XfYmZtZxXTPFtJrzQowMu4SRiguqlsKfOzJnCnQr3Ck%3D
```

*Created without using a return code

6

Use HTTPS to issue a GET request on the basis of the statement of the request you have created. The endpoint of the Cloudⁿ Provisioning service is:

<https://prov-api.jp-e1.cloudn-service.com/Provision/queryapi/>

```
GET /?  
Action=DescribeStacks&SignatureMethod=HmacSHA256&SignatureVersion=2&AWSAccessKeyId=<APIKEY>&Version=2010-05-15&Timestamp=2013-01-30T18%3A09%3A45Z&Signature=XfYmZtZxXTPFtJrzQowMu4SRiguqlsKfOzJnCnQr3Ck%3D
```

*Created without using a return code

1-3) Checking a Response

1

If your request is successful, a response in the xml format returns as shown below:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<DescribeStacksResponse xmlns="http://cloudformation.amazonaws.com/doc/2010-05-15/">
  <DescribeStacksResult>
    <Stacks>
      <member>
        <StackId>7683a06d-8f09-4f0d-bb18-fb9e8f2ff3e_sg2</StackId>
        <StackStatus>CREATE_COMPLETE</StackStatus>
        <StackName>SampleStack</StackName>
        <StackStatusReason/>
        <Description>SampleStack</Description>
        <CreationTime>2013-04-05T16:56:53Z</CreationTime>
        <DisableRollback>>false</DisableRollback>
      </member>
      <member>
        <StackId>d8649d7c-4a1c-4ceb-9129-cfa7fcc554f_Compute-Ref</StackId>
        <StackStatus>CREATE_COMPLETE</StackStatus>
        <StackName>Compute-Ref</StackName>
        <StackStatusReason/>
        <Description>Sample Stack2</Description>
        <CreationTime>2013-04-05T16:50:41Z</CreationTime>
        <DisableRollback>>false</DisableRollback>
        <Outputs>
          <member>
            <OutputValue>jp-e1a</OutputValue>
            <Description>Availability Zone of the newly created EC2 instance</Description>
            <OutputKey>AZ</OutputKey>
          </member>
          <member>
            <OutputValue>1b165066-f35a-4fb5-b56f-1aaecb6b89b5</OutputValue>
            <Description>InstanceId of the newly created EC2 instance</Description>
            <OutputKey>InstanceId</OutputKey>
          </member>
        </Outputs>
      </member>
    </Stacks>
  </DescribeStacksResult>
  <ResponseMetadata>
    <RequestId>bd2b1662-1831-4c5d-8bab-f95f74356912</RequestId>
  </ResponseMetadata>
</DescribeStacksResponse>
```


2-1) Setting Up a Stack

This section shows a specific example to describe how you set up a stack in Cloudⁿ Provisioning.

1

In this section, a stack in Cloudⁿ Provisioning is set up by using an API. First of all, prepare a template in advance that will be included in the stack. The template shown below is used to create a new security group in an environment of Cloud Compute. This template is used in this example.

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "Provisioning Sample Template.",
  "Resources" : {
    "SecurityGroupSSH" : {
      "Type" : "Cloudn::Compute::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable SSH access via port 22 from 192.0.2.1/32",
        "SecurityGroupIngress" : [ {
          "IpProtocol" : "tcp",
          "FromPort" : "22",
          "ToPort" : "22",
          "CidrIp" : "192.0.2.1/32"
        } ]
      }
    }
  }
}
```

To register the template mentioned above as a template, you have two options: ① upload the above-mentioned template on a Web site of your choice and define the URL, or ② follow RFC3986 to execute URL-encoding and use a query string for the definition. In this section, assume that we use option ① and that the above-mentioned template is uploaded to "http://www.example.com/sample.template".

Specify the following parameters, and issue the request.

Command (Parameter) = key	Value (Sample) = value
Action	CreateStack
StackName	SecuritySSH
TemplateURL	http%3A%2F%2Fwww.example.com%2Fsample.template
SignatureMethod	HmacSHA256
SignatureVersion	2
Version	2010-05-15
Timestamp	2013-01-30T18%3A09%3A45Z
AWSAccessKeyId	<APIKEY>
Signature	<Signature>



For the details of the parameters, see section 3, "Provisioning API Reference".

2-1) Setting Up a Stack

2

If the request is successful, a response returns in the xml format as shown below:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<CreateStackResponse xmlns="http://cloudformation.amazonaws.com/doc/2010-05-15/">
  <CreateStackResult>
    <StackId>f21bbb5c-e15b-433b-8377-758d2825dc38_SecuritySSH</StackId>
  </CreateStackResult>
  <ResponseMetadata>
    <RequestId>c497561f-09f0-483d-b152-cfafad023484</RequestId>
  </ResponseMetadata>
</CreateStackResponse>
```

3

If you call for the DescribeStacks API, you can confirm that the stack is registered as shown below:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<DescribeStacksResponse xmlns="http://cloudformation.amazonaws.com/doc/2010-05-15/">
  <DescribeStacksResult>
    <Stacks>
      <member>
        <StackId>f21bbb5c-e15b-433b-8377-758d2825dc38_SecuritySSH</StackId>
        <StackStatus>CREATE_COMPLETE</StackStatus>
        <StackName>SecuritySSH</StackName>
        <StackStatusReason/>
        <Description>Provisioning Sample Template.</Description>
        <CreationTime>2013-01-30T18:09:45Z</CreationTime>
        <DisableRollback>>false</DisableRollback>
      </member>
    </Stacks>
  </DescribeStacksResult>
  <ResponseMetadata>
    <RequestId>9e6660d7-c72a-4936-add8-c83ac3ee3712</RequestId>
  </ResponseMetadata>
</DescribeStacksResponse>
```



For the information about calling for DescribeStacks, see section 3, "Provisioning API Reference".

2-2) Checking a Stack/Template

1

If you call for the DescribeStacks API, you can confirm that the stack is registered as shown below:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<DescribeStacksResponse xmlns="http://cloudformation.amazonaws.com/doc/2010-05-15/">
  <DescribeStacksResult>
    <Stacks>
      <member>
        <StackId>f21bbb5c-e15b-433b-8377-758d2825dc38_SecuritySSH</StackId>
        <StackStatus>CREATE_COMPLETE</StackStatus>
        <StackName>SecuritySSH</StackName>
        <StackStatusReason/>
        <Description>Provisioning Sample Template.</Description>
        <CreationTime>2013-01-30T18:09:45Z</CreationTime>
        <DisableRollback>>false</DisableRollback>
      </member>
    </Stacks>
  </DescribeStacksResult>
  <ResponseMetadata>
    <RequestId>9e6660d7-c72a-4936-add8-c83ac3ee3712</RequestId>
  </ResponseMetadata>
</DescribeStacksResponse>
```



For the information about calling for DescribeStacks, see section 3, "Provisioning API Reference".

2-2) Checking a Stack/Template

2

If you call for the GetTemplate API, you can confirm the template in the stack as shown below:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<GetTemplateResponse xmlns="http://cloudformation.amazonaws.com/doc/2010-05-15/">
  <GetTemplateResult>
    <TemplateBody>
      {
        "AWSTemplateFormatVersion" : "2010-09-09",
        "Description" : "Provisioning Sample Template.",
        "Resources" : {
          "SecurityGroupSSH" : {
            "Type" : "Cloudn::Compute::SecurityGroup",
            "Properties" : {
              "GroupDescription" : "Enable SSH access via port 22 from 192.0.2.1/32",
              "SecurityGroupIngress" : [ {
                "IpProtocol" : "tcp",
                "FromPort" : "22",
                "ToPort" : "22",
                "CidrIp" : "192.0.2.1/32"
              } ]
            }
          }
        }
      }
    </TemplateBody>
  </GetTemplateResult>
  <ResponseMetadata>
    <RequestId>be0efc21-148d-4510-8158-e04368938889</RequestId>
  </ResponseMetadata>
</GetTemplateResponse>
```



For the information about calling for GetTemplate, see section 3, Provisioning API Reference".

3-1) Provisioning API List (Action)

This service provides the following Actions.

	Command	Description
Action	CreateStack	Creates a stack as specified in the template.
	DeleteStack	Deletes a specified stack.
	DescribeStackEvents	Returns all stack related events for a specified stack.
	DescribeStackResource	Returns a description of the specified resource in the specified stack.
	DescribeStackResources	Returns Cloud ⁿ resource descriptions for running and deleted stacks
	DescribeStacks	Returns the description for the specified stack.
	GetTemplate	Returns the template body for a specified stack name.
	ListStackResources	Returns descriptions of all resources of the specified stack.
	ListStacks	Returns the summary information for stacks whose status matches the specified StackStatusFilter.
	UpdateStack	Updates a stack as specified in the template.
	ValidateTemplate	Validates a specified template.



The following APIs are not supported in this service.

CancelUpdateStack
EstimateTemplateCost

3-2) Provisioning API Common Information

API Server (End Point) for this service is as follows,

<https://prov-api.jp-e1.cloudn-service.com/Provision/queryapi/>

The following table describes parameters that are used for Cloudⁿ Provisioning Query API Request.

Common Parameters		
Parameter	Description	Required
Action	The action to be performed. Default : None Type : String	Yes
AuthParams	Not Supported	No
AWSAccessKeyId	The access key ID that corresponds to the secret access key that you used to sign the request. Default : None Type : String	Yes
Expires	Not Supported	No
SecurityToken	Not Supported	No
Signature	The digital signature that you created for the request. Default : None Type : String	Yes
SignatureMethod	The hash algorithm that you used to create the request signature. Default : None Valid Values : HmacSHA256 HmacSHA1 Type : String	Yes
SignatureVersion	The signature version you use to sign the request. Set this to the value that is recommended for your service. Default : None Valid Values : 2 Type : String	Yes
Timestamp	The date and time when the request was signed, expressed in the format YYYY-MM-DDThh:mm:ssZ, as specified in the ISO 8601 standard. Default : None Type : String	Yes
Version	The API version that the request is written for, expressed in the format YYYY-MM-DD. Default : None Valid Values : 2010-05-15 Type : String	Yes

3-3) Provisioning API

CreateStack

Description	Creates a stack as specified in the template.
-------------	---

Request	
Request Method	GET
Path,Parameters, etc	Parameters described in "Common Parameters" and "Request Parameters" below are used. "Create Stack" is used as Action.
Header	None
Body	None

Request Parameters		
Parameter	Description	Required
StackName	The name associated with the stack.	Yes
TemplateBody	Structure containing the template body. Minimum length of 1. Maximum length of 51200.	Yes (*1)
TemplateURL	Location of file containing the template body	Yes (*1)
DisableRollback	Set to true to disable rollback of the stack if stack creation failed.You can specify either DisableRollback or OnFailure, but not both. (Default: false)	No
OnFailure	Determines what action will be taken if stack creation fails. This must be one of: DO_NOTHING, ROLLBACK, or DELETE.You can specify either OnFailure or DisableRollback, but not both. (Default : ROLLBACK)	No
TimeoutInMinutes	The amount of time that can pass before the stack status becomes CREATE_FAILED.	No
Parameters.member.N	A list of Parameter structures that specify input parameters for the stack. Parameter consists of Name-Value Pair, so you need to specify Parameters.member.N.ParameterKey and Parameters.member.N.ParameterValue. [e.g.] Parameters.member.1.ParameterKey=example Parameters.member.1.ParameterValue=29831-beef-cafe-2929229	No
MailAddress.member.N (*2)	E-mail notification to the specified address when stack create completed.	No



(*1) Required either TemplateBody or TemplateURL. TemplateBody has a priority in case both parameters specified.

- (*2) Original function of Cloudⁿ Provisioning which does not exist in AWS, and alternative implementation of "NotificationARNs.member.N" in AWS.

Response				
Response (Body)	Format	XML		
	XML	Node Name	Description	Child Node
		<i>CreateStackResponse</i>	Container of Response	ResponseMetadata, CreateStackResult
		<i>ResponseMetadata</i>	Container of metadata	RequestId
		<i>RequestId</i>	Request ID	
		<i>CreateStackResult</i>	Container which stores result	StackID
		<i>StackId</i>	Unique ID which shows stack instance	

3-3) Provisioning API

DeleteStack

Description	Deletes a specified stack.
-------------	----------------------------

Request

Request Method	GET
Path,Parameters, etc	Parameters described in "Common Parameters" and "Request Parameters" below are used. "Delete Stack" is used as Action.
Header	None
Body	None

Request Parameters

Parameter	Description	Required
StackName	The name associated with the stack.	Yes

Response

Response (Body)	Format	XML		
	XML	Node Name	Description	Child Node
		<i>DeleteStackResponse</i>	Container of Response	ResponseMetadata, DeleteStackResult
		<i>ResponseMetadata</i>	Container of metadata	RequestId
		<i>RequestId</i>	Request ID	
		<i>DeleteStackResult</i>	Container which stores result	

3-3) Provisioning API

DescribeStackEvents

Description Returns all stack related events for a specified stack.

Request

Request Method	GET
Path,Parameters, etc	Parameters described in "Common Parameters" and "Request Parameters" below are used. "DescribeStack Event" is used as Action.
Header	None
Body	None

Request Parameters

Parameter	Description	Required
StackName	The name or the unique identifier associated with the stack.	No
NextToken	String that identifies the start of the next list of events, if there is one.	No

Response

Response (Body)	Format	XML																																							
	XML	<table border="1"> <thead> <tr> <th>Node Name</th> <th>Description</th> <th>Child Node</th> </tr> </thead> <tbody> <tr> <td><i>DescribeStackEventsResponse</i></td> <td>Container of Response</td> <td>ResponseMetadata, DescribeStackEventsResult, NextToken</td> </tr> <tr> <td><i>ResponseMetadata</i></td> <td>Container of metadata</td> <td>RequestId</td> </tr> <tr> <td><i>RequestId</i></td> <td>Request ID</td> <td></td> </tr> <tr> <td><i>DescribeStackEventsResult</i></td> <td>All stack related events for a specified stack.</td> <td>StackEvent</td> </tr> <tr> <td><i>NextToken</i></td> <td>String that identifies the start of the next list of events, if there is one.</td> <td></td> </tr> <tr> <td><i>StackEvent</i></td> <td>The StackEvent data type.</td> <td>member</td> </tr> <tr> <td><i>member</i></td> <td>Container which stores result</td> <td>EventId, LogicalResourceId, PhysicalResourceId, ResourceProperties, ResourceStatus, ResourceStatusReason, ResourceType, StackId, StackName, Timestamp</td> </tr> <tr> <td><i>EventId</i></td> <td>Unique ID which identifies event</td> <td></td> </tr> <tr> <td><i>LogicalResourceId</i></td> <td>The logical name of the resource as specified in the template</td> <td></td> </tr> <tr> <td><i>PhysicalResourceId</i></td> <td>The name or unique identifier that corresponds to a physical instance ID of a resource.</td> <td></td> </tr> <tr> <td><i>ResourceProperties</i></td> <td>BLOB of the properties used to create the resource.</td> <td></td> </tr> <tr> <td><i>ResourceStatus</i></td> <td>Current status of the resource. CREATE_IN_PROGRESS CREATE_FAILED CREATE_COMPLETE DELETE_IN_PROGRESS DELETE_FAILED DELETE_COMPLETE UPDATE_IN_PROGRESS UPDATE_FAILED UPDATE_COMPLETE</td> <td></td> </tr> </tbody> </table>	Node Name	Description	Child Node	<i>DescribeStackEventsResponse</i>	Container of Response	ResponseMetadata, DescribeStackEventsResult, NextToken	<i>ResponseMetadata</i>	Container of metadata	RequestId	<i>RequestId</i>	Request ID		<i>DescribeStackEventsResult</i>	All stack related events for a specified stack.	StackEvent	<i>NextToken</i>	String that identifies the start of the next list of events, if there is one.		<i>StackEvent</i>	The StackEvent data type.	member	<i>member</i>	Container which stores result	EventId, LogicalResourceId, PhysicalResourceId, ResourceProperties, ResourceStatus, ResourceStatusReason, ResourceType, StackId, StackName, Timestamp	<i>EventId</i>	Unique ID which identifies event		<i>LogicalResourceId</i>	The logical name of the resource as specified in the template		<i>PhysicalResourceId</i>	The name or unique identifier that corresponds to a physical instance ID of a resource.		<i>ResourceProperties</i>	BLOB of the properties used to create the resource.		<i>ResourceStatus</i>	Current status of the resource. CREATE_IN_PROGRESS CREATE_FAILED CREATE_COMPLETE DELETE_IN_PROGRESS DELETE_FAILED DELETE_COMPLETE UPDATE_IN_PROGRESS UPDATE_FAILED UPDATE_COMPLETE	
		Node Name	Description	Child Node																																					
		<i>DescribeStackEventsResponse</i>	Container of Response	ResponseMetadata, DescribeStackEventsResult, NextToken																																					
		<i>ResponseMetadata</i>	Container of metadata	RequestId																																					
		<i>RequestId</i>	Request ID																																						
		<i>DescribeStackEventsResult</i>	All stack related events for a specified stack.	StackEvent																																					
		<i>NextToken</i>	String that identifies the start of the next list of events, if there is one.																																						
		<i>StackEvent</i>	The StackEvent data type.	member																																					
		<i>member</i>	Container which stores result	EventId, LogicalResourceId, PhysicalResourceId, ResourceProperties, ResourceStatus, ResourceStatusReason, ResourceType, StackId, StackName, Timestamp																																					
		<i>EventId</i>	Unique ID which identifies event																																						
		<i>LogicalResourceId</i>	The logical name of the resource as specified in the template																																						
		<i>PhysicalResourceId</i>	The name or unique identifier that corresponds to a physical instance ID of a resource.																																						
		<i>ResourceProperties</i>	BLOB of the properties used to create the resource.																																						
		<i>ResourceStatus</i>	Current status of the resource. CREATE_IN_PROGRESS CREATE_FAILED CREATE_COMPLETE DELETE_IN_PROGRESS DELETE_FAILED DELETE_COMPLETE UPDATE_IN_PROGRESS UPDATE_FAILED UPDATE_COMPLETE																																						

3-3) Provisioning API

DescribeStackEvents

Response				
Response (Body)	Format	XML		
	XML	Node Name	Description	Child Node
		<i>ResourceStatusReason</i>	Success/failure message associated with the resource	
		<i>ResourceType</i>	Type of Resource	
		<i>StackId</i>	Unique ID which shows stack instance	
		<i>StackName</i>	The name associated with the stack.	
		<i>Timestamp</i>	The date and time when the request was signed	

3-3) Provisioning API

DescribeStackResource

Description Returns a description of the specified resource in the specified stack.

Request

Request Method	GET
Path,Parameters, etc	Parameters described in "Common Parameters" and "Request Parameters" below are used. "DescribeStackResource" is used as Action.
Header	None
Body	None

Request Parameters

Parameter	Description	Required
LogicalResourceId	The logical name of the resource specified in the template.	Yes
StackName	The name associated with the stack.	Yes

Response

Response (Body)	Format	XML																																																
	XML	<table border="1"> <thead> <tr> <th>Node Name</th> <th>Description</th> <th>Child Node</th> </tr> </thead> <tbody> <tr> <td><i>DescribeStackResourceResponse</i></td> <td>Container of Response</td> <td>ResponseMetadata, DescribeStackResourceResult,</td> </tr> <tr> <td><i>ResponseMetadata</i></td> <td>Container of metadata</td> <td>RequestId</td> </tr> <tr> <td><i>RequestId</i></td> <td>Request ID</td> <td></td> </tr> <tr> <td><i>DescribeStackResourceResult</i></td> <td>Container which stores result</td> <td>StackResourceDetail</td> </tr> <tr> <td><i>StackResourceDetail</i></td> <td>Containing the description of the specified resource in the specified stack.</td> <td>Description, Metadata, LastUpdatedTimestamp, LogicalResourceId, PhysicalResourceId, StackName, ResourceStatus, ResourceStatusReason, StackId</td> </tr> <tr> <td><i>Description</i></td> <td>User defined description associated with the resource.</td> <td></td> </tr> <tr> <td><i>LastUpdatedTimestamp</i></td> <td>Time the status was updated.</td> <td></td> </tr> <tr> <td><i>LogicalResourceId</i></td> <td>The logical name of the resource specified in the template.</td> <td></td> </tr> <tr> <td><i>Metadata</i></td> <td>The JSON format content of the Metadata attribute declared for the resource.</td> <td></td> </tr> <tr> <td><i>PhysicalResourceId</i></td> <td>The name or unique identifier that corresponds to a physical instance ID of a resource.</td> <td></td> </tr> <tr> <td><i>ResourceStatus</i></td> <td>Current status of the resource. CREATE_IN_PROGRESS CREATE_FAILED CREATE_COMPLETE DELETE_IN_PROGRESS DELETE_FAILED DELETE_COMPLETE UPDATE_IN_PROGRESS UPDATE_FAILED UPDATE_COMPLETE</td> <td></td> </tr> <tr> <td><i>ResourceStatusReason</i></td> <td>Success/failure message associated with the resource.</td> <td></td> </tr> <tr> <td><i>ResourceType</i></td> <td>Type of the resource.</td> <td></td> </tr> <tr> <td><i>StackId</i></td> <td>Unique ID which shows stack instance</td> <td></td> </tr> <tr> <td><i>StackName</i></td> <td>The name associated with the stack.</td> <td></td> </tr> </tbody> </table>	Node Name	Description	Child Node	<i>DescribeStackResourceResponse</i>	Container of Response	ResponseMetadata, DescribeStackResourceResult,	<i>ResponseMetadata</i>	Container of metadata	RequestId	<i>RequestId</i>	Request ID		<i>DescribeStackResourceResult</i>	Container which stores result	StackResourceDetail	<i>StackResourceDetail</i>	Containing the description of the specified resource in the specified stack.	Description, Metadata, LastUpdatedTimestamp, LogicalResourceId, PhysicalResourceId, StackName, ResourceStatus, ResourceStatusReason, StackId	<i>Description</i>	User defined description associated with the resource.		<i>LastUpdatedTimestamp</i>	Time the status was updated.		<i>LogicalResourceId</i>	The logical name of the resource specified in the template.		<i>Metadata</i>	The JSON format content of the Metadata attribute declared for the resource.		<i>PhysicalResourceId</i>	The name or unique identifier that corresponds to a physical instance ID of a resource.		<i>ResourceStatus</i>	Current status of the resource. CREATE_IN_PROGRESS CREATE_FAILED CREATE_COMPLETE DELETE_IN_PROGRESS DELETE_FAILED DELETE_COMPLETE UPDATE_IN_PROGRESS UPDATE_FAILED UPDATE_COMPLETE		<i>ResourceStatusReason</i>	Success/failure message associated with the resource.		<i>ResourceType</i>	Type of the resource.		<i>StackId</i>	Unique ID which shows stack instance		<i>StackName</i>	The name associated with the stack.	
		Node Name	Description	Child Node																																														
		<i>DescribeStackResourceResponse</i>	Container of Response	ResponseMetadata, DescribeStackResourceResult,																																														
		<i>ResponseMetadata</i>	Container of metadata	RequestId																																														
		<i>RequestId</i>	Request ID																																															
		<i>DescribeStackResourceResult</i>	Container which stores result	StackResourceDetail																																														
		<i>StackResourceDetail</i>	Containing the description of the specified resource in the specified stack.	Description, Metadata, LastUpdatedTimestamp, LogicalResourceId, PhysicalResourceId, StackName, ResourceStatus, ResourceStatusReason, StackId																																														
		<i>Description</i>	User defined description associated with the resource.																																															
		<i>LastUpdatedTimestamp</i>	Time the status was updated.																																															
		<i>LogicalResourceId</i>	The logical name of the resource specified in the template.																																															
		<i>Metadata</i>	The JSON format content of the Metadata attribute declared for the resource.																																															
		<i>PhysicalResourceId</i>	The name or unique identifier that corresponds to a physical instance ID of a resource.																																															
		<i>ResourceStatus</i>	Current status of the resource. CREATE_IN_PROGRESS CREATE_FAILED CREATE_COMPLETE DELETE_IN_PROGRESS DELETE_FAILED DELETE_COMPLETE UPDATE_IN_PROGRESS UPDATE_FAILED UPDATE_COMPLETE																																															
		<i>ResourceStatusReason</i>	Success/failure message associated with the resource.																																															
		<i>ResourceType</i>	Type of the resource.																																															
<i>StackId</i>	Unique ID which shows stack instance																																																	
<i>StackName</i>	The name associated with the stack.																																																	

3-3) Provisioning API

DescribeStackResources

Description Returns Cloudⁿ resource descriptions for running and deleted stacks

Request

Request Method	GET
Path,Parameters, etc	Parameters described in "Common Parameters" and "Request Parameters" below are used. "DescribeStackResources" is used as Action.
Header	None
Body	None

Request Parameters

Parameter	Description	Required
LogicalResourceId	The logical name of the resource specified in the template.	No
StackName	The name associated with the stack.	No
PhysicalResourceId	The name or unique identifier that corresponds to a physical instance ID of a resource.	

Response

Response (Body)	Format	XML		
	XML	Node Name	Description	Child Node
		<i>DescribeStackResourceResponse</i>	Container of Response	ResponseMetadata, DescribeStackResourcesResult,
		<i>ResponseMetadata</i>	Container of metadata	RequestId
		<i>RequestId</i>	Request ID	
		<i>DescribeStackResourcesResult</i>	Container which stores result	StackResources
		<i>StackResources</i>	The StackResource data type.	Description, LastUpdatedTimestamp, LogicalResourceId, PhysicalResourceId, StackName, ResourceStatus, ResourceStatusReason, StackId
		<i>Description</i>	User defined description associated with the resource.	
		<i>Timestamp</i>	The date and time when the request was signed	
		<i>LogicalResourceId</i>	The logical name of the resource specified in the template.	
		<i>PhysicalResourceId</i>	The name or unique identifier that corresponds to a physical instance ID of a resource.	
		<i>ResourceStatus</i>	Current status of the resource. CREATE_IN_PROGRESS CREATE_FAILED CREATE_COMPLETE DELETE_IN_PROGRESS DELETE_FAILED DELETE_COMPLETE UPDATE_IN_PROGRESS UPDATE_FAILED UPDATE_COMPLETE	
		<i>ResourceStatusReason</i>	Success/failure message associated with the resource.	
		<i>ResourceType</i>	Type of the resource.	
		<i>StackId</i>	Unique ID which shows stack instance	
		<i>StackName</i>	The name associated with the stack.	

3-3) Provisioning API

DescribeStacks

Description Returns the description for the specified stack.

Request

Request Method	GET
Path,Parameters, etc	Parameters described in "Common Parameters" and "Request Parameters" below are used. "DescribeStacks" is used as Action.
Header	None
Body	None

Request Parameters

Parameter	Description	Required
StackName	The name or the unique identifier associated with the stack.	No
NextToken	String that identifies the start of the next list of events, if there is one.	No

Response

Response (Body)	Format	XML																																																										
	XML	<table border="1"> <thead> <tr> <th>Node Name</th> <th>Description</th> <th>Child Node</th> </tr> </thead> <tbody> <tr> <td><i>DescribeStacksResponse</i></td> <td>Container of Response</td> <td>ResponseMetadata, DescribeStacksResult, NextToken</td> </tr> <tr> <td><i>ResponseMetadata</i></td> <td>Container of metadata</td> <td>RequestId</td> </tr> <tr> <td><i>RequestId</i></td> <td>Request ID</td> <td></td> </tr> <tr> <td><i>DescribeStacksResult</i></td> <td>All stack related events for a specified stack.</td> <td>Stacks</td> </tr> <tr> <td><i>NextToken</i></td> <td>String that identifies the start of the next list of events, if there is one.</td> <td></td> </tr> <tr> <td><i>Stacks</i></td> <td>The StackEvent data type.</td> <td>Member</td> </tr> <tr> <td><i>Member</i></td> <td>Container which stores result</td> <td><i>Capabilities, CreationTime, Description, DisableRollback, LastUpdatedTime, outputs, Parameters, StackId, StackName, StackStatus, StackStatusReason, TimeoutInMinutes</i></td> </tr> <tr> <td><i>Capabilities</i></td> <td>The capabilities allowed in the stack.</td> <td></td> </tr> <tr> <td><i>CreationTime</i></td> <td>Time at which the stack was created.</td> <td></td> </tr> <tr> <td><i>Description</i></td> <td>User defined description associated with the stack.</td> <td></td> </tr> <tr> <td><i>DisableRollback</i></td> <td>Boolean to enable or disable rollback on stack creation failures.</td> <td></td> </tr> <tr> <td><i>LastUpdatedTime</i></td> <td>The time the stack was last updated. This field will only be returned if the stack has been updated at least once.</td> <td></td> </tr> <tr> <td><i>Outputs</i></td> <td>A list of output structures.</td> <td>Member</td> </tr> <tr> <td></td> <td><i>member</i></td> <td>Contents of each Output container</td> <td>Description, OutputKey, OutputValue</td> </tr> <tr> <td></td> <td><i>Description</i></td> <td>User defined description associated with the output.</td> <td></td> </tr> <tr> <td></td> <td><i>OutputKey</i></td> <td>The key associated with the output.</td> <td></td> </tr> <tr> <td></td> <td><i>OutputValue</i></td> <td>The value associated with the output.</td> <td></td> </tr> </tbody> </table>	Node Name	Description	Child Node	<i>DescribeStacksResponse</i>	Container of Response	ResponseMetadata, DescribeStacksResult, NextToken	<i>ResponseMetadata</i>	Container of metadata	RequestId	<i>RequestId</i>	Request ID		<i>DescribeStacksResult</i>	All stack related events for a specified stack.	Stacks	<i>NextToken</i>	String that identifies the start of the next list of events, if there is one.		<i>Stacks</i>	The StackEvent data type.	Member	<i>Member</i>	Container which stores result	<i>Capabilities, CreationTime, Description, DisableRollback, LastUpdatedTime, outputs, Parameters, StackId, StackName, StackStatus, StackStatusReason, TimeoutInMinutes</i>	<i>Capabilities</i>	The capabilities allowed in the stack.		<i>CreationTime</i>	Time at which the stack was created.		<i>Description</i>	User defined description associated with the stack.		<i>DisableRollback</i>	Boolean to enable or disable rollback on stack creation failures.		<i>LastUpdatedTime</i>	The time the stack was last updated. This field will only be returned if the stack has been updated at least once.		<i>Outputs</i>	A list of output structures.	Member		<i>member</i>	Contents of each Output container	Description, OutputKey, OutputValue		<i>Description</i>	User defined description associated with the output.			<i>OutputKey</i>	The key associated with the output.			<i>OutputValue</i>	The value associated with the output.	
		Node Name	Description	Child Node																																																								
		<i>DescribeStacksResponse</i>	Container of Response	ResponseMetadata, DescribeStacksResult, NextToken																																																								
		<i>ResponseMetadata</i>	Container of metadata	RequestId																																																								
		<i>RequestId</i>	Request ID																																																									
		<i>DescribeStacksResult</i>	All stack related events for a specified stack.	Stacks																																																								
		<i>NextToken</i>	String that identifies the start of the next list of events, if there is one.																																																									
		<i>Stacks</i>	The StackEvent data type.	Member																																																								
		<i>Member</i>	Container which stores result	<i>Capabilities, CreationTime, Description, DisableRollback, LastUpdatedTime, outputs, Parameters, StackId, StackName, StackStatus, StackStatusReason, TimeoutInMinutes</i>																																																								
		<i>Capabilities</i>	The capabilities allowed in the stack.																																																									
		<i>CreationTime</i>	Time at which the stack was created.																																																									
		<i>Description</i>	User defined description associated with the stack.																																																									
		<i>DisableRollback</i>	Boolean to enable or disable rollback on stack creation failures.																																																									
		<i>LastUpdatedTime</i>	The time the stack was last updated. This field will only be returned if the stack has been updated at least once.																																																									
		<i>Outputs</i>	A list of output structures.	Member																																																								
	<i>member</i>	Contents of each Output container	Description, OutputKey, OutputValue																																																									
	<i>Description</i>	User defined description associated with the output.																																																										
	<i>OutputKey</i>	The key associated with the output.																																																										
	<i>OutputValue</i>	The value associated with the output.																																																										

3-3) Provisioning API

DescribeStacks

Response (Cont.)

Response (Body)	Format	XML				
	XML	Node Name		Description	Child Node	
		<i>Parameters</i>		Container for parameters	<i>Member</i>	
		<i>Member</i>	<i>Member</i>		Contents of each Output container	ParameterKey, ParameterValue
				<i>ParameterKey</i>	The key associated with the parameter.	
				<i>ParameterValue</i>	The value associated with the parameter.	
		<i>StackId</i>		Unique ID which shows stack instance		
		<i>StackName</i>		The name associated with the stack.		
		<i>StackStatus</i>		Current status of the stack. CREATE_IN_PROGRESS CREATE_FAILED CREATE_COMPLETE ROLLBACK_IN_PROGRESS ROLLBACK_FAILED ROLLBACK_COMPLETE DELETE_IN_PROGRESS DELETE_FAILED DELETE_COMPLETE UPDATE_IN_PROGRESS UPDATE_COMPLETE_CLEANUP_IN_PROGRESS UPDATE_COMPLETE UPDATE_ROLLBACK_IN_PROGRESS UPDATE_ROLLBACK_FAILED UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS UPDATE_ROLLBACK_COMPLETE		
		<i>StackStatusReason</i>		Success/failure message associated with the stack status.		
<i>TimeoutInMinutes</i>		The amount of time within which stack creation should complete.				

3-3) Provisioning API

GetTemplate

Description	Returns the template body for a specified stack name.
-------------	---

Request

Request Method	GET
Path,Parameters, etc	Parameters described in "Common Parameters" and "Request Parameters" below are used. "GetTemplate" is used as Action.
Header	None
Body	None

Request Parameters

Parameter	Description	Required
StackName	The name associated with the stack.	Yes

Response

Response (Body)	Format	XML		
	XML	Node Name	Description	Child Node
		<i>GetTemplateResponse</i>	Container of Response	ResponseMetadata, GetTemplateResult,
		<i>ResponseMetadata</i>	Container of metadata	RequestId
		<i>RequestId</i>	Request ID	
		<i>GetTemplateResult</i>	Container which stores result	TemplateBody
		<i>TemplateBody</i>	Structure containing the template body (JSON Format).	

3-3) Provisioning API (Action)

ListStackResources

Description	Returns descriptions of all resources of the specified stack.
-------------	---

Request

Request Method	GET
Path,Parameters, etc	Parameters described in "Common Parameters" and "Request Parameters" below are used. "ListStackResources" is used as Action.
Header	None
Body	None

Request Parameters

Parameter	Description	Required
StackName	The name or the unique identifier associated with the stack.	No
NextToken	String that identifies the start of the next list of events, if there is one.	No

Response

Response (Body)	Format	XML		
	XML	Node Name	Description	Child Node
		<i>ListStackResourcesResponse</i>	Container of Response	ResponseMetadata, ListStackResourcesResult, NextToken
		<i>ResponseMetadata</i>	Container of metadata	RequestId
		<i>RequestId</i>	Request ID	
		<i>ListStackResourcesResult</i>	Container which stores result	StackResourceSummaries,
		<i>NextToken</i>	String that identifies the start of the next list of events, if there is one.	
		<i>StackResourceSummaries</i>	The StackEvent data type.	Member
		<i>Member</i>	Container which stores result	LastUpdatedTimestamp, LogicalResourceId, PhysicalResourceId, ResourceStatus, ResourceStatusReason, ResourceType,
		<i>LastUpdatedTimestamp</i>	Time the status was updated.	
		<i>LogicalResourceId</i>	The logical name of the resource specified in the template.	
		<i>PhysicalResourceId</i>	The name or unique identifier that corresponds to a physical instance ID of a resource	
		<i>ResourceStatus</i>	Current status of the resource. CREATE_IN_PROGRESS CREATE_FAILED CREATE_COMPLETE DELETE_IN_PROGRESS DELETE_FAILED DELETE_COMPLETE UPDATE_IN_PROGRESS UPDATE_FAILED UPDATE_COMPLETE	
		<i>ResourceStatusReason</i>	Success/failure message associated with the resource.	
		<i>ResourceType</i>	Type of the resource.	

3-3) Provisioning API (Action)

ListStacks

Description	Returns the summary information for stacks whose status matches the specified StackStatusFilter.
Request	
Request Method	GET
Path,Parameters, etc	Parameters described in "Common Parameters" and "Request Parameters" below are used. "ListStacks" is used as Action.
Header	None
Body	None

Request Parameters		
Parameter	Description	Required
NextToken	String that identifies the start of the next list of events, if there is one.	No
StackStatusFilter.member.N	Stack status to use as a filter. Specify one or more stack status codes to list only stacks with the specified status codes. Valid Values are as follows, CREATE_IN_PROGRESS CREATE_FAILED CREATE_COMPLETE ROLLBACK_IN_PROGRESS ROLLBACK_FAILED ROLLBACK_COMPLETE DELETE_IN_PROGRESS DELETE_FAILED DELETE_COMPLETE UPDATE_IN_PROGRESS UPDATE_COMPLETE_CLEANUP_IN_PROGRESS UPDATE_COMPLETE UPDATE_ROLLBACK_IN_PROGRESS UPDATE_ROLLBACK_FAILED UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS UPDATE_ROLLBACK_COMPLETE [e.g.] StackStatusFilter.member.1=CREATE_COMPLETE	No

Response				
Response (Body)	Format	XML		
		XML	Node Name	Description
		<i>ListStacksResponse</i>	Container of Response	ResponseMetadata, ListStacksResult, NextToken
		<i>ResponseMetadata</i>	Container of metadata	RequestId
		<i>RequestId</i>	Request ID	
		<i>ListStacksResult</i>	The output for ListStacks	StackResourceSummaries,
		<i>NextToken</i>	String that identifies the start of the next list of events, if there is one.	
		<i>StackResourceSummaries</i>	A list of StackResourceSummary structures.	Member
		<i>Member</i>	Container which stores result.	LastUpdatedTimestamp, LogicalResourceId, PhysicalResourceId, ResourceStatus, ResourceStatusReason, ResourceType,
		<i>LastUpdatedTimestamp</i>	Time the status was updated.	
		<i>LogicalResourceId</i>	The logical name of the resource specified in the template.	
		<i>PhysicalResourceId</i>	The name or unique identifier that corresponds to a physical instance ID of a resource.	

3-3) Provisioning API

ListStacks

Response (Cont.)				
Response (Body)	Format	XML		
	XML	Node Name	Description	Child Node
		<i>ResourceStatus</i>	Current status of the resource. CREATE_IN_PROGRESS CREATE_FAILED CREATE_COMPLETE DELETE_IN_PROGRESS DELETE_FAILED DELETE_COMPLETE UPDATE_IN_PROGRESS UPDATE_FAILED UPDATE_COMPLETE	
		<i>ResourceStatusReason</i>	Success/failure message associated with the resource.	
<i>ResourceType</i>	Type of the resource.			

3-3) Provisioning API

UpdateStack

Description	Updates a stack as specified in the template.
-------------	---

Request	
Request Method	GET
Path,Parameters, etc	Parameters described in "Common Parameters" and "Request Parameters" below are used. "UpdateStack" is used as Action.
Header	None
Body	None

Request Parameters1		
Parameter	Description	Required
StackName	The name associated with the stack.	Yes
TemplateBody	Structure containing the template body. Minimum length of 1. Maximum length of 51200.	Yes (*1)
TemplateURL	Location of file containing the template body	Yes (*1)
Parameters.member.N	A list of Parameter structures that specify input parameters for the stack. Parameter consists of Name-Value Pair, so you need to specify Parameters.member.N.ParameterKey and Parameters.member.N.ParameterValue. [e.g.] Parameters.member.1.ParameterKey=example Parameters.member.1.ParameterValue=29831-beef-cafe-2929229	No



(*1) Required either TemplateBody or TemplateURL. TemplateBody has a priority in case both parameters specified.

Response				
Response (Body)	Format	XML		
	XML	Node Name	Description	Child Node
		<i>UpdateStackResponse</i>	Container of Response	ResponseMetadata, UpdateStackResult
		<i>ResponseMetadata</i>	Container of metadata	RequestId
		<i>RequestId</i>	Request ID	
		<i>UpdateStackResult</i>	The output for a UpdateStack action.	StackID
		<i>StackId</i>	Unique ID which shows stack instance	

3-3) Provisioning API

ValidateTemplate

Description | Validates a specified template.

Request

Request Method	GET
Path,Parameters, etc	Parameters described in "Common Parameters" and "Request Parameters" below are used. "ValidateTemplate" is used as Action.
Header	None
Body	None

Request Parameters1

Parameter	Description	Required
TemplateBody	Structure containing the template body. Minimum length of 1. Maximum length of 51200.	Yes (*1)
TemplateURL	Location of file containing the template body	Yes (*1)



(*1) Required either TemplateBody or TemplateURL. TemplateBody has a priority in case both parameters specified.

Response

Response (Body)	Format	XML																																				
	XML	<table border="1"> <thead> <tr> <th>Node Name</th> <th>Description</th> <th>Child Node</th> </tr> </thead> <tbody> <tr> <td><i>ValidateTemplateResponse</i></td> <td>Container of Response</td> <td>ResponseMetadata, UpdateStackResult</td> </tr> <tr> <td><i>ResponseMetadata</i></td> <td>Container of metadata</td> <td>RequestId</td> </tr> <tr> <td><i>RequestId</i></td> <td>Request ID</td> <td></td> </tr> <tr> <td><i>ValidateTemplateResult</i></td> <td>Container which stores result</td> <td>Description, Parameters</td> </tr> <tr> <td><i>Description</i></td> <td>Unique ID which shows stack</td> <td></td> </tr> <tr> <td><i>Parameters</i></td> <td>Container for parameters</td> <td>Member</td> </tr> <tr> <td><i>Member</i></td> <td>Contents of each Output container</td> <td>DefaultValue, Description, NoEcho, ParameterKey</td> </tr> <tr> <td><i>DefaultValue</i></td> <td>The default value associated with the parameter.</td> <td></td> </tr> <tr> <td><i>Description</i></td> <td>User defined description associated with the parameter.</td> <td></td> </tr> <tr> <td><i>NoEcho</i></td> <td>Flag indicating whether the parameter should be displayed as plain text in logs and UIs.</td> <td></td> </tr> <tr> <td><i>ParameterKey</i></td> <td>The key associated with the parameter.</td> <td></td> </tr> </tbody> </table>	Node Name	Description	Child Node	<i>ValidateTemplateResponse</i>	Container of Response	ResponseMetadata, UpdateStackResult	<i>ResponseMetadata</i>	Container of metadata	RequestId	<i>RequestId</i>	Request ID		<i>ValidateTemplateResult</i>	Container which stores result	Description, Parameters	<i>Description</i>	Unique ID which shows stack		<i>Parameters</i>	Container for parameters	Member	<i>Member</i>	Contents of each Output container	DefaultValue, Description, NoEcho, ParameterKey	<i>DefaultValue</i>	The default value associated with the parameter.		<i>Description</i>	User defined description associated with the parameter.		<i>NoEcho</i>	Flag indicating whether the parameter should be displayed as plain text in logs and UIs.		<i>ParameterKey</i>	The key associated with the parameter.	
		Node Name	Description	Child Node																																		
		<i>ValidateTemplateResponse</i>	Container of Response	ResponseMetadata, UpdateStackResult																																		
		<i>ResponseMetadata</i>	Container of metadata	RequestId																																		
		<i>RequestId</i>	Request ID																																			
		<i>ValidateTemplateResult</i>	Container which stores result	Description, Parameters																																		
		<i>Description</i>	Unique ID which shows stack																																			
		<i>Parameters</i>	Container for parameters	Member																																		
		<i>Member</i>	Contents of each Output container	DefaultValue, Description, NoEcho, ParameterKey																																		
		<i>DefaultValue</i>	The default value associated with the parameter.																																			
		<i>Description</i>	User defined description associated with the parameter.																																			
		<i>NoEcho</i>	Flag indicating whether the parameter should be displayed as plain text in logs and UIs.																																			
		<i>ParameterKey</i>	The key associated with the parameter.																																			