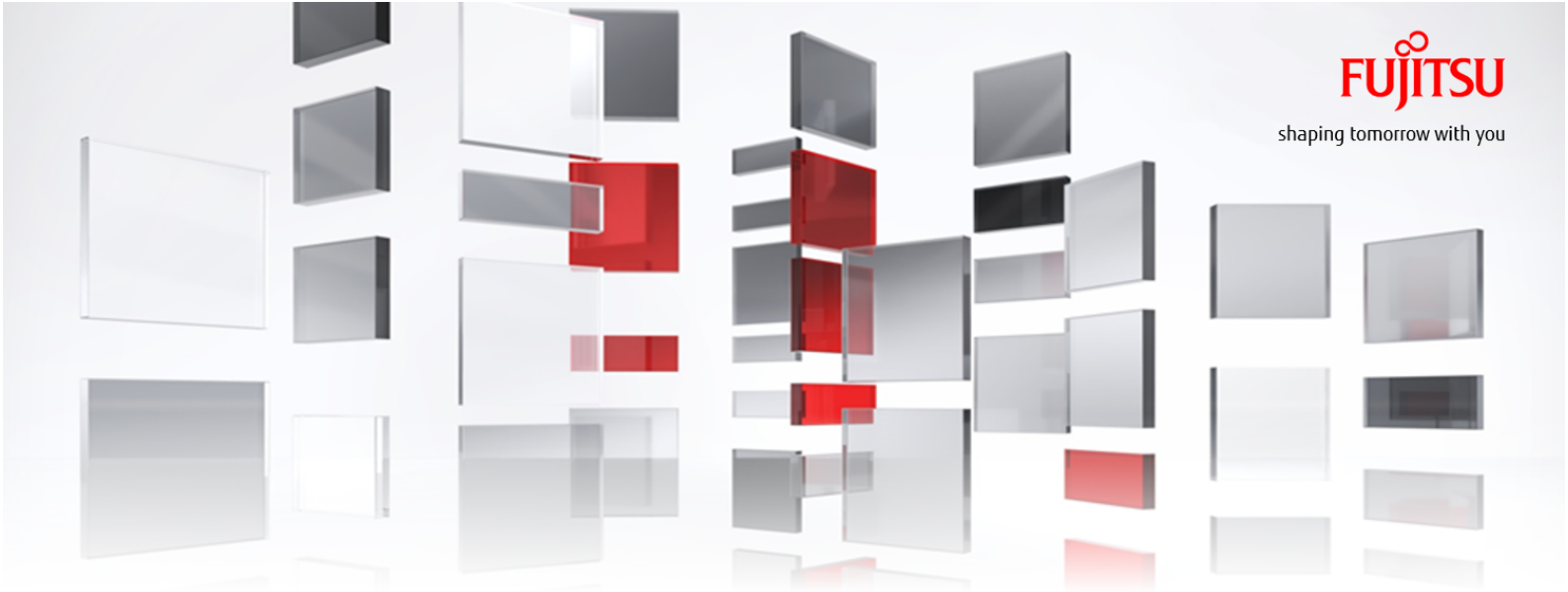


FUJITSU
shaping tomorrow with you

FUJITSU Cloud Service for OSS

IaaS Heat Template Specifications

Version 2.7
FUJITSU LIMITED

K5IA-DC-M-003-001E

Preface

Purpose of This Document

This document describes the specifications of and how to use the Heat template supplied with FUJITSU Cloud Service for OSS IaaS. Refer to this document when building a system on the FUJITSU Cloud Service for OSS IaaS.

Intended readers

This document is intended for users who plan or develop applications and services for use on FUJITSU Cloud Service for OSS IaaS. Readers of this document are assumed to have:

- Basic knowledge of virtualization technology (hypervisor, virtual server, virtual storage, virtual network).
- Basic knowledge of OpenStack
- Basic knowledge of the operating system that will be used
- Basic knowledge related to the Internet and intranets.
- Basic knowledge related to security.
- Basic knowledge of system operations, such as backups, monitoring, and redundancy

Target region of this document

The target regions of this document are East Japan Region 1, East Japan Region 2, West Japan Region 1, West Japan Region 2.

Structure of the manuals

Refer to the following related manuals according to your purpose and usage.

Manual	Purpose and usage
IaaS Features Handbook	Explains details on the features provided by this service.
IaaS API User Guide	Explains how to use the REST API, how to build the API runtime environment, and sample scripts according to usage sequences, etc.
IaaS API Reference	Refer to this document as a detailed reference when using the REST API.
IaaS Service Portal User Guide	Explains how to use the features provided by this service when using the service portal (Web GUI).

Trademarks

Export Administration Regulations

Exportation/release of this document may require following the necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Notices

- Information in this document may be subject to change without prior notice.
- No part of the content of this document may be reproduced without the written permission of Fujitsu Limited.
- Fujitsu assumes no responsibility for infringement of any patent rights or other rights of third parties arising from the use of information in this document.

Document history

Version	Date of Update	Edited places	Description
2.1	July 25, 2017	4.1.5.3 OS::Neutron::FirewallRule	Corrected a mistake
		4.1.5.11 OS::Neutron::Subnet	Corrected a mistake
2.3	December 22, 2017	All sections	Changed structure
		All sections	Corrected the description
		Preface	Corrected the description
		1.1 Purpose and scope of this document	Corrected the description
		2 What is Heat?	Description added
		3.1 Template Structure	Description added
		3.2 Parameters Section	Description added
		3.2.1 Parameter Constraints	Description added
		3.2.2 Pseudo Parameters	Description added
		3.3 Resources Section	Description added
		3.3.2 Resource Dependencies	Description added
		3.3.3 Referring to information	Description added
		3.4 Outputs Section	Description added
		6.7 Sample system configuration - Example Heat template	Description added
		Appendix B Creating stacks using APIs	Description added
		Appendix C Handling errors	Description added
		Appendix D Referring to outputs	Description added
		5.1.5.11.2 Note	Description added
		5.1.7.2.1.1 Basic parameters	Corrected the description

Version	Date of Update	Edited places	Description
		5.1.7.4.1.1 Basic parameters	Corrected the description
2.4	February 22, 2018	Absorbed the content of v1.7, and indicated in various sections of this document that some content varies depending on the region being used	Changed structure
2.5	March 22, 2018	Basic parameters	Changed structure
		Basic parameters	Changed structure
		Example Security Group creation Heat template	Description added
		Example Virtual Server creation with global IP assigned Heat template	Description added
		Example Database Virtual Server creation Heat template	Description added
2.6	April 19, 2018	Updating a stack	Description added
		Supported resource types	Description added
		Object Storage	Description added
		Example virtual server creation Heat template	Changed structure
		Example Heat template for changing the flavor of a virtual server using stack update	Description added
		Example virtual server creation (Windows OS) Heat template with computer name specification	Changed structure
		Database	Description added
		Object Storage	Description added
2.6.1	June 29, 2018	Preface	Target region names added
2.7	July 31, 2018	Notes	Note is added

Contents

Part 1: Preface

Topics:

1.1 Purpose and scope of this document

This document explains the Heat template of the FUJITSU Cloud Service for OSS IaaS, which is used to provide the orchestration function.

It should be noted that the content of this document is subject to change without prior notice.

Part 2: What is Heat?

Topics:

2.1 Overview of Heat

Using the IaaS, it is possible to automatically create and manage systems that involve virtual resources such as "virtual servers", "virtual networks", and "virtual volumes", using the provided orchestration function. Heat places requests for operations with the coordination function based on various definition templates (text-based).
For details on the virtual resources that can be used with Heat, refer to "Supported resource types".

Using Heat has the following merits:

1. Improved efficiency in system construction and management operations
2. Configuration of auto scaling

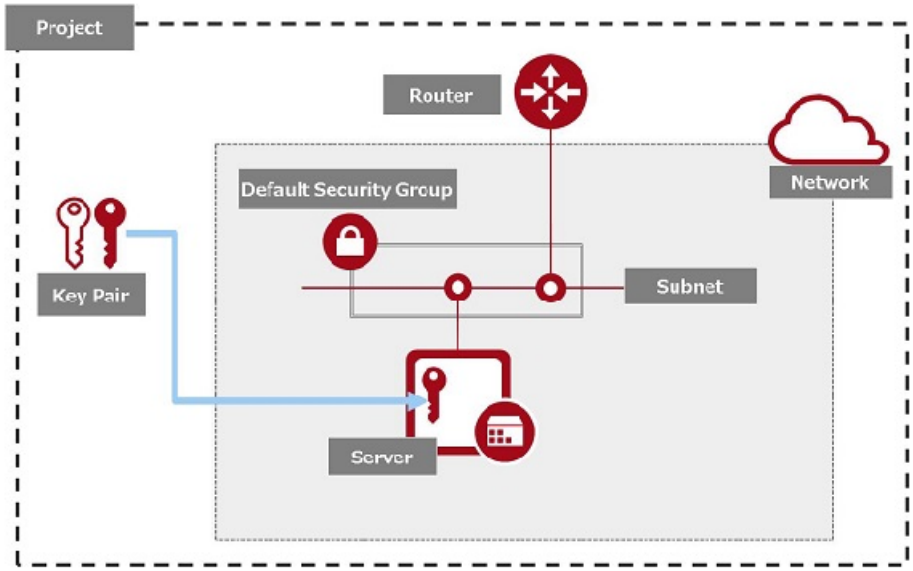
The details of each of these merits are explained below.

Improved efficiency in system construction and management operations

Using Heat it is possible to perform batch creation of virtual resources. In addition, reusing created Heat template definitions makes it possible to easily replicate systems and perform modification or deletion. This enables efficient system construction and management on IaaS, and reduces the workload of users.

An example, comparing the results of when construction is performed manually using the IaaS service portal and when the same operation (based on the number of screen operations on the IaaS service portal) is performed using Heat is given on the following page.

This comparison is based on construction of a simple system involving a single virtual server and one network as shown in the following figure.



SecurityGroup uses the default security group that is selected automatically.

Figure 1: System Configuration for Workload Comparison

The following figure shows the comparison results for each step of system construction.

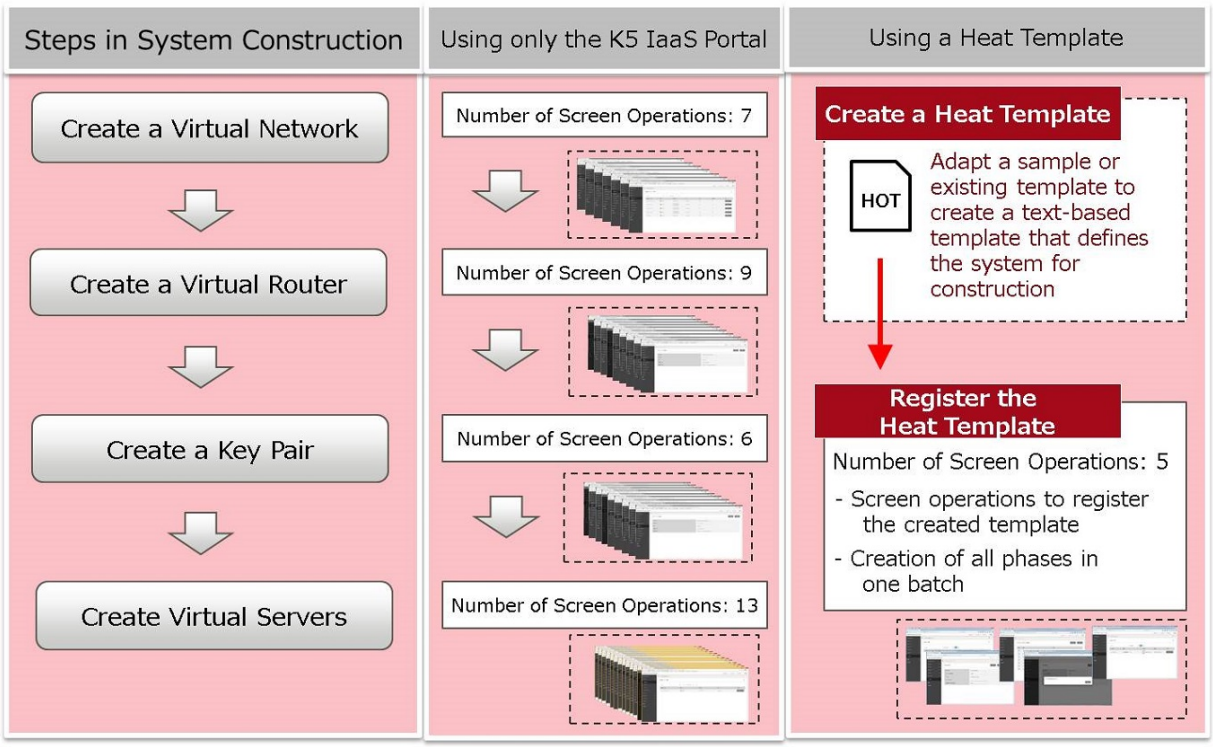


Figure 2: Comparison of System Construction work Based on the Number of Operations on the IaaS Service Portal Window

Configuration of auto scaling

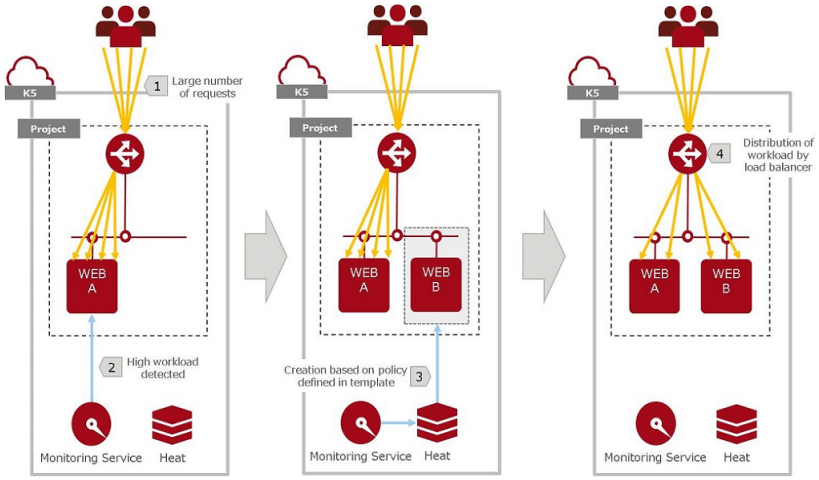
Auto scaling is a function for dynamically increasing or decreasing the number of virtual servers in a system, based on the CPU workload and other events.

Auto scaling enables handling of sporadic increases in the requests for services.

In addition, as it is possible to automatically reduce the number of virtual servers when the number of requests drops, unnecessary expenses are not incurred.

This auto scaling can only be configured using Heat.

The figure below gives an image of the operation of auto scaling (scale out) based on high CPU workload of a virtual server for which load balancing is being performed.



The figure below gives an image of the operation of auto scaling (scale in) when the workload of the CPU lowers.

Figure 3: Image of Auto Scaling Operations (Scale Out)

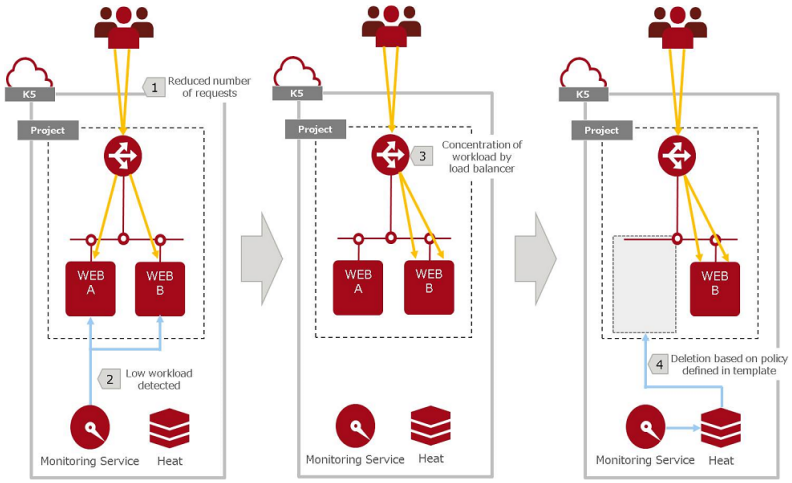


Figure 4: Image of Auto Scaling Operations (Scale In)

2.2 What are Heat templates?

In order to use Heat, text-based templates written in the YAML format are necessary. These templates are generally referred to as HOT (Heat Orchestration Templates), but in this document they are referred to as Heat templates.
YAML is a format for expressing structured data. Heat templates described in the YAML format are based on the format introduced in "HEAT Orchestration Template format". Each section is used to describe resources, parameters, etc., and a single template describes the configuration of an entire system.

Expressing these in highly readable YAML makes it easy to manage and modify system configurations.

Also, the fact that they are text based makes them easy to reuse.

Based on the content defined in the Heat template, the various resources are created on IaaS. This group of resources is referred to as a "stack".

Heat templates are text-based templates that are used to perform the definition, creation, and management of stacks.

2.3 Creating stacks using Heat templates

Here, as an example, we explain the procedure for creating a stack from a Heat template, using the IaaS service portal.
The Heat template used as the sample is the one in "Sample system configuration - Example Heat template" (Heat_template_sample.yaml).

The content of the stack created using this sample is as shown below.

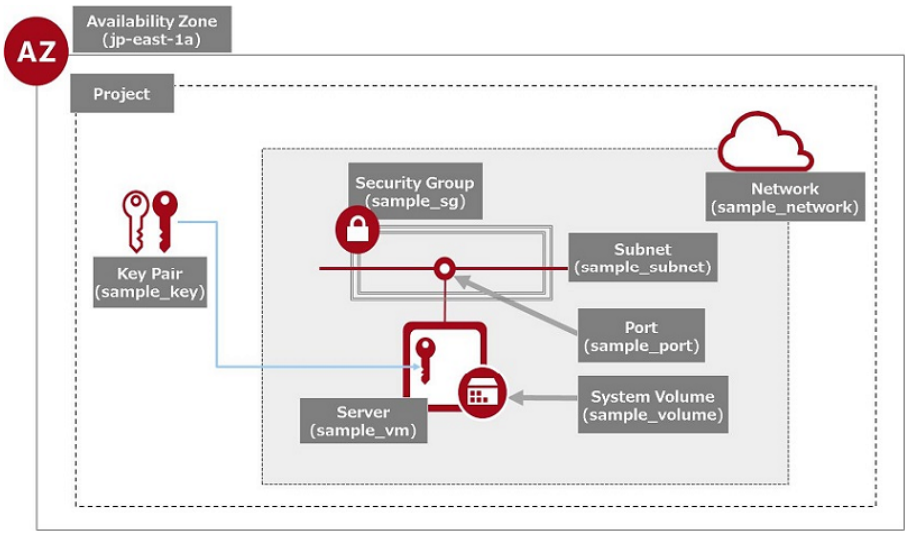


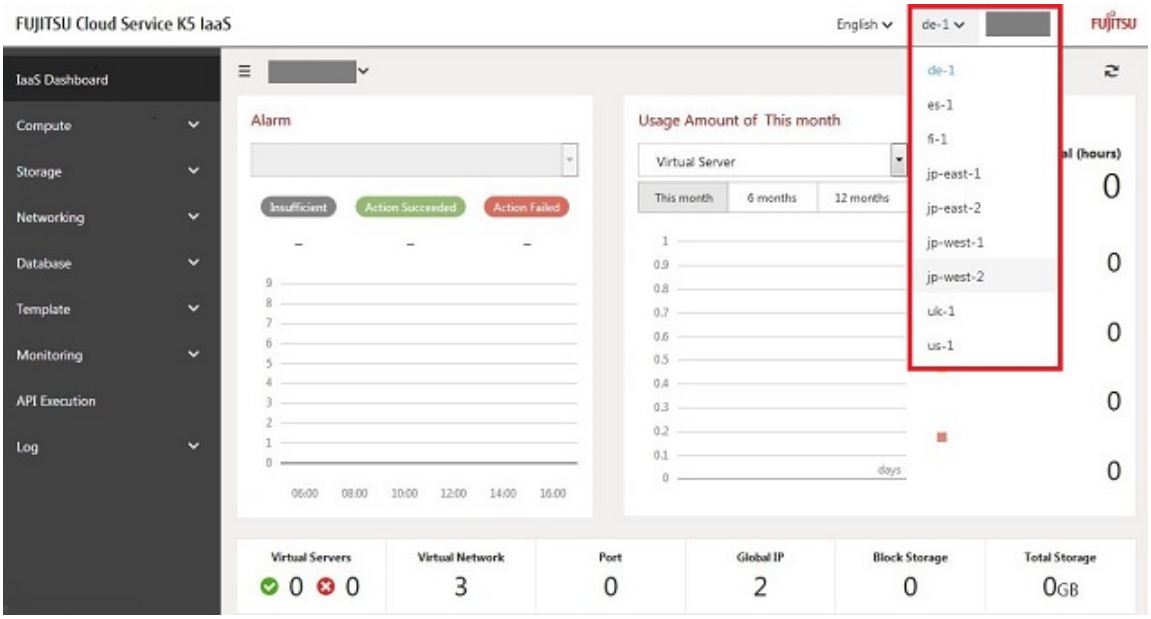
Figure 5: Sample System Configuration

For the method for creation using APIs, refer to "Creating stacks using APIs".
The following sections show the procedures for creating stacks and for checking stacks.

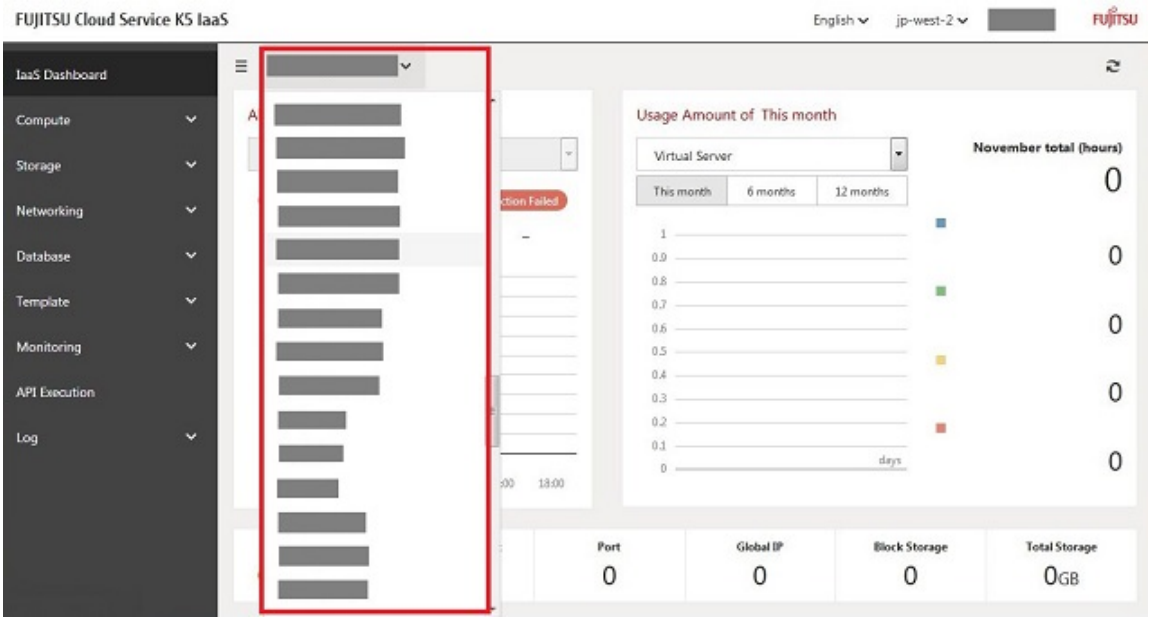
2.3.1 Creating a stack

Procedure

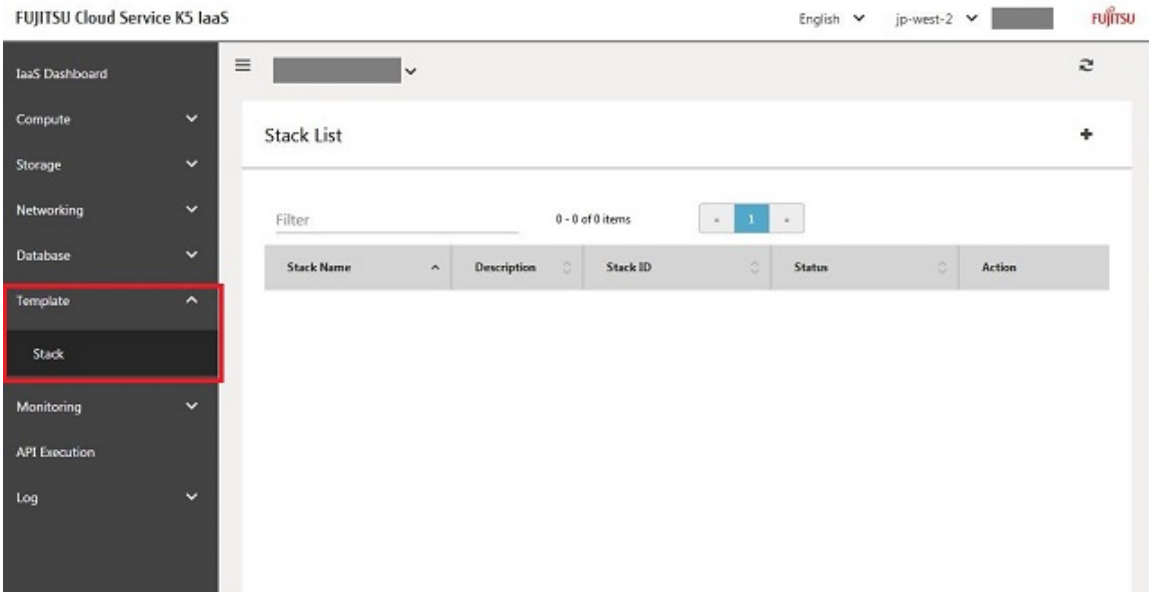
1. After logging in to the Portal, selecting "IaaS Portal" from the "Management" tab displays the "IaaS Dashboard" as below. Here, select the region in which to create the stack.



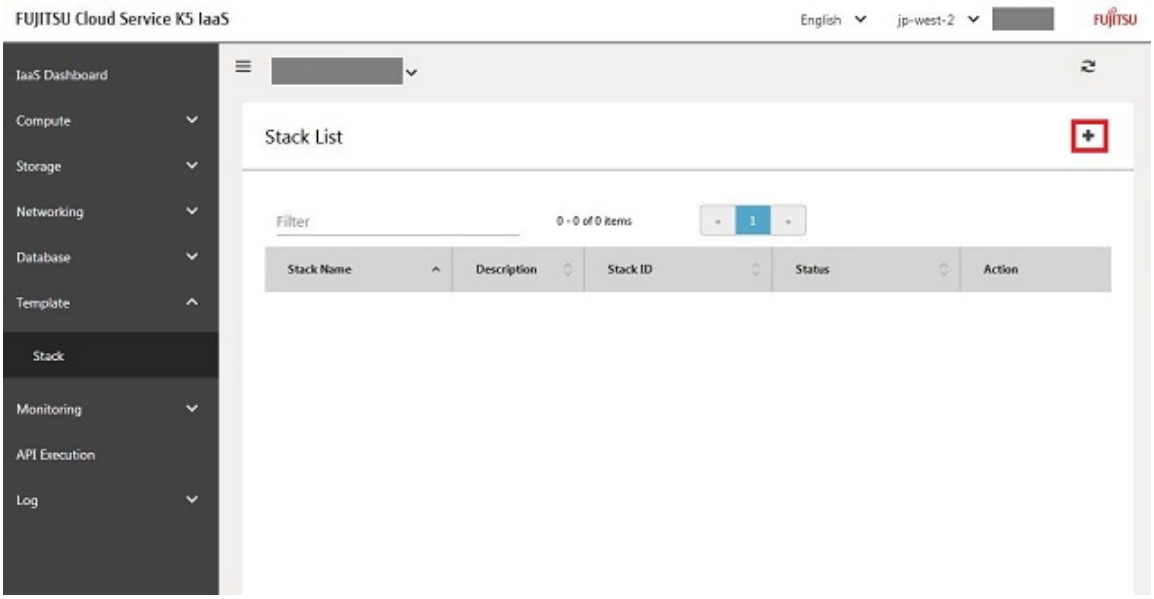
2. Select the project you want to create the stack in.



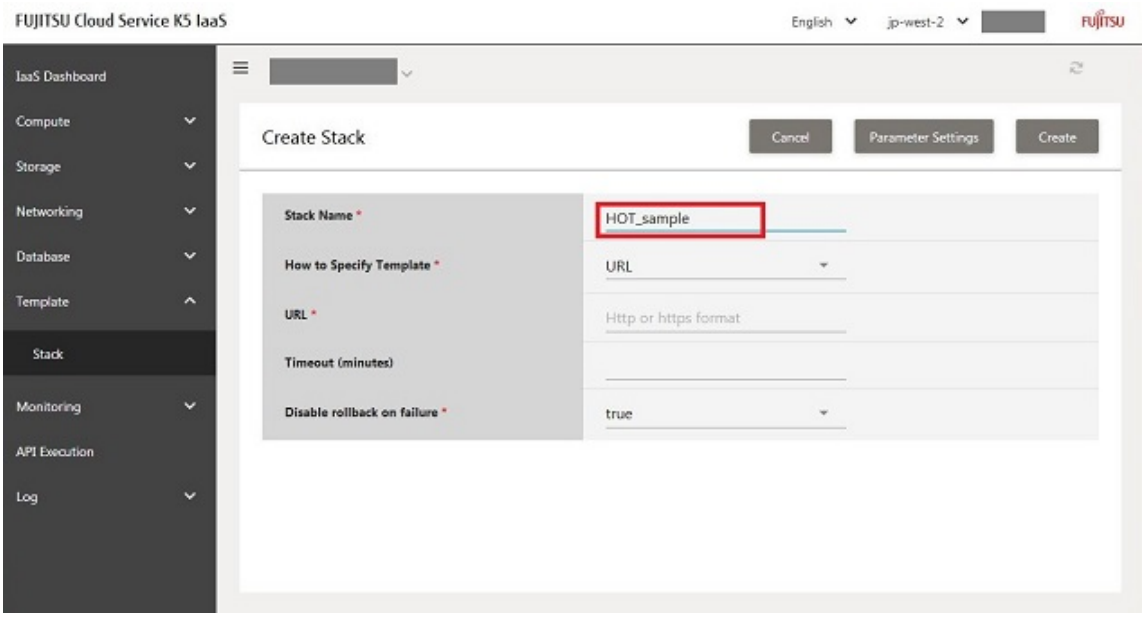
3. Clicking "Template" > "Stack" displays the "Stack list" window.



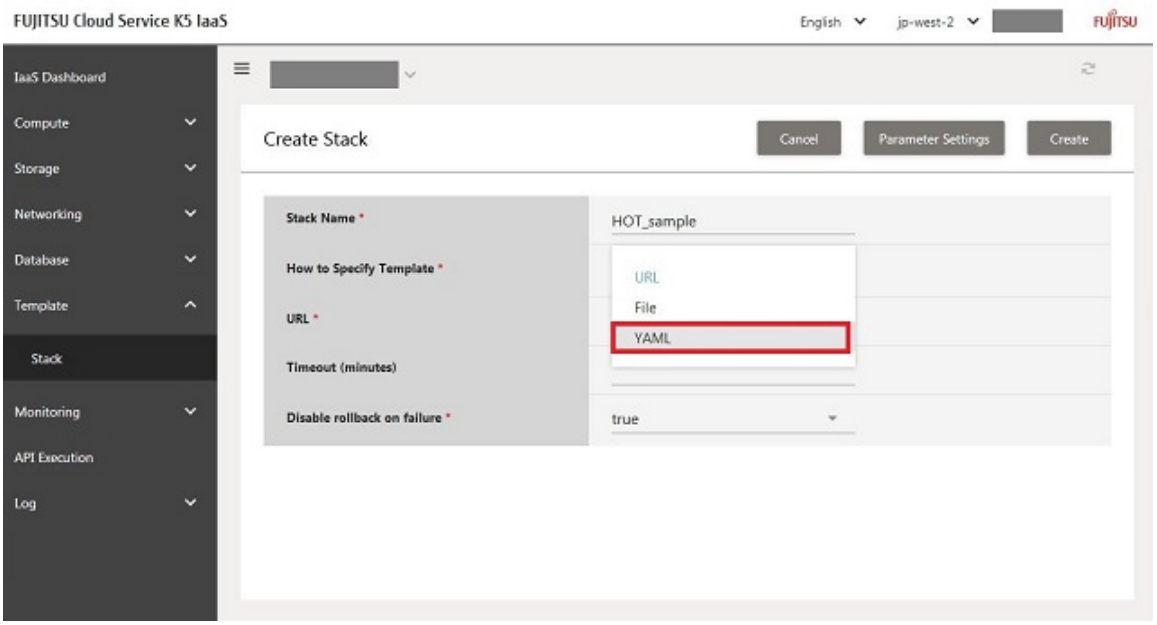
4. Clicking the "+" on the upper right displays the "Create Stack" window.



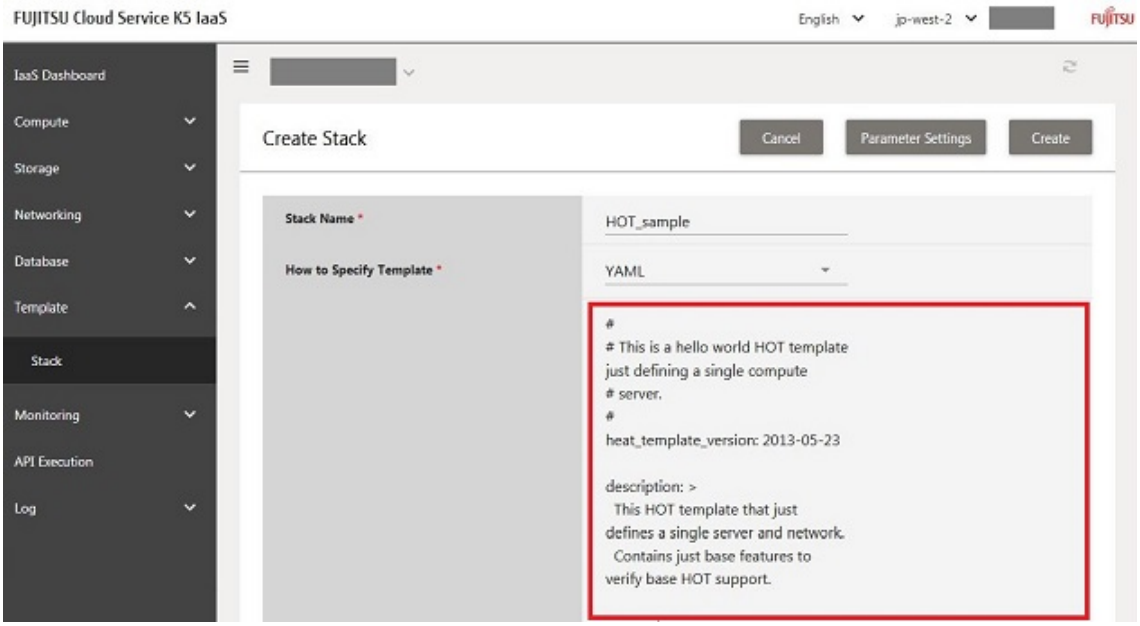
5. Input the desired name for the stack.



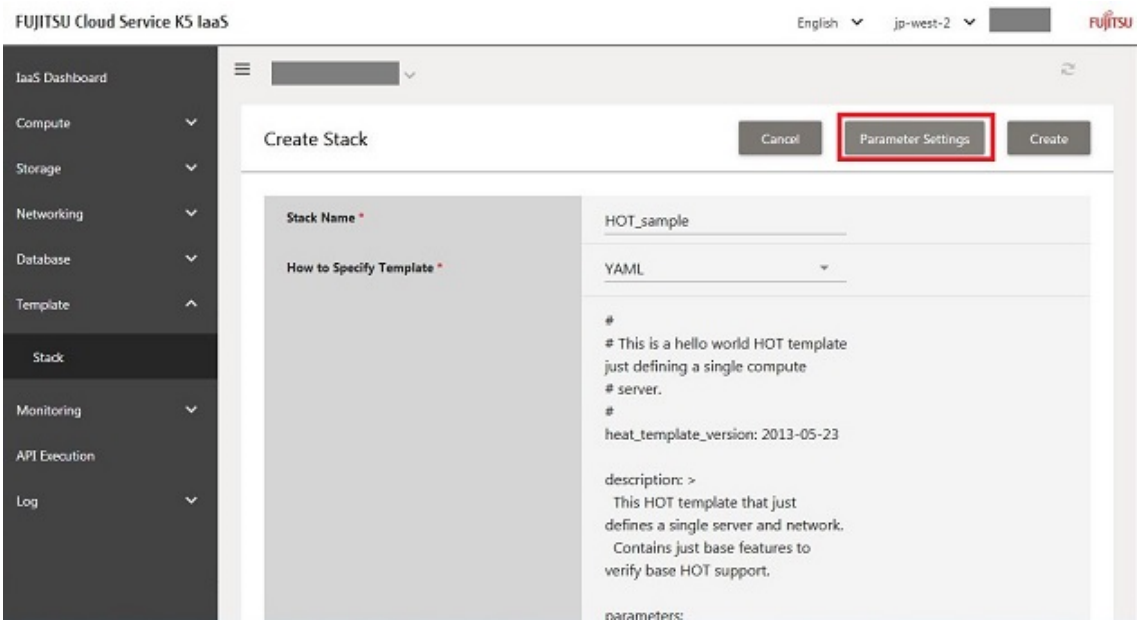
6. Selecting "YAML" from "How to Specify Template" displays the "YAML Editor" field.



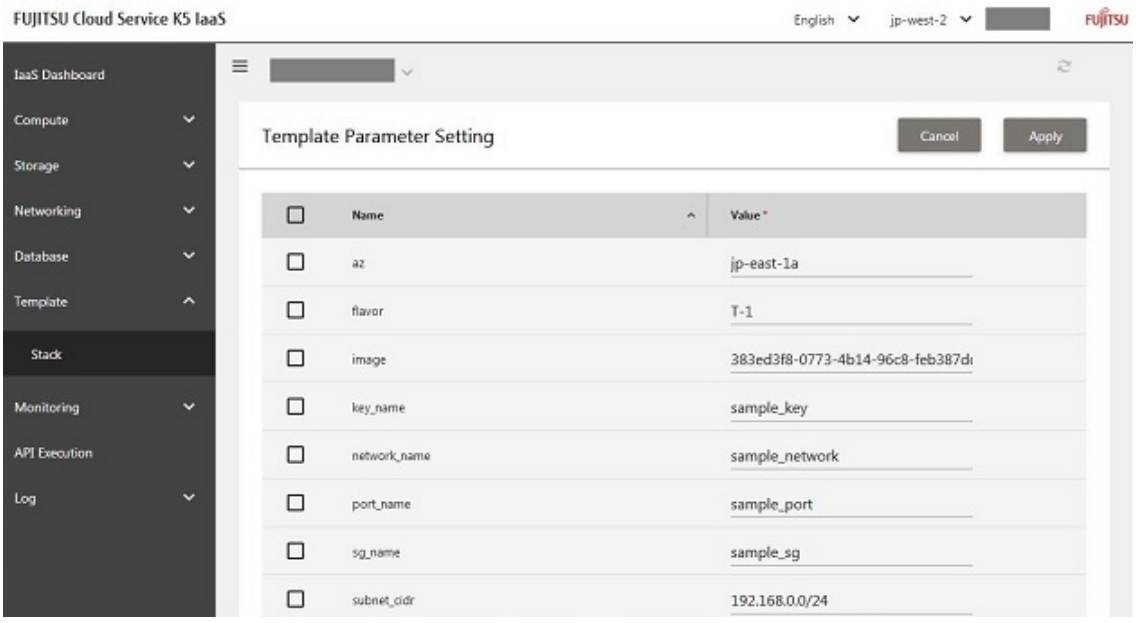
7. Copy the content of the sample Heat template (Heat_template_sample.yaml), and paste it into the "YAML Editor" field.



8. Clicking "Parameter Settings" on the upper right displays the "Template Parameter Setting" window.

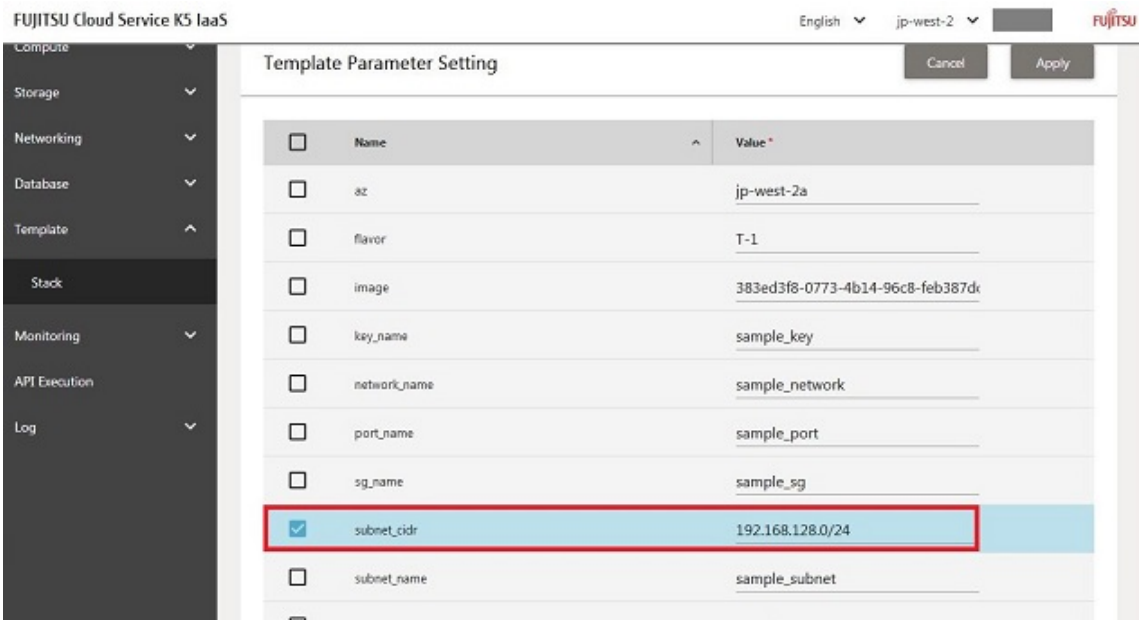


Template Parameter Setting Window - Display Example



The values defined in the parameters section of the sample are displayed for each parameter.
For details of the parameters section, refer to "Parameters Section".

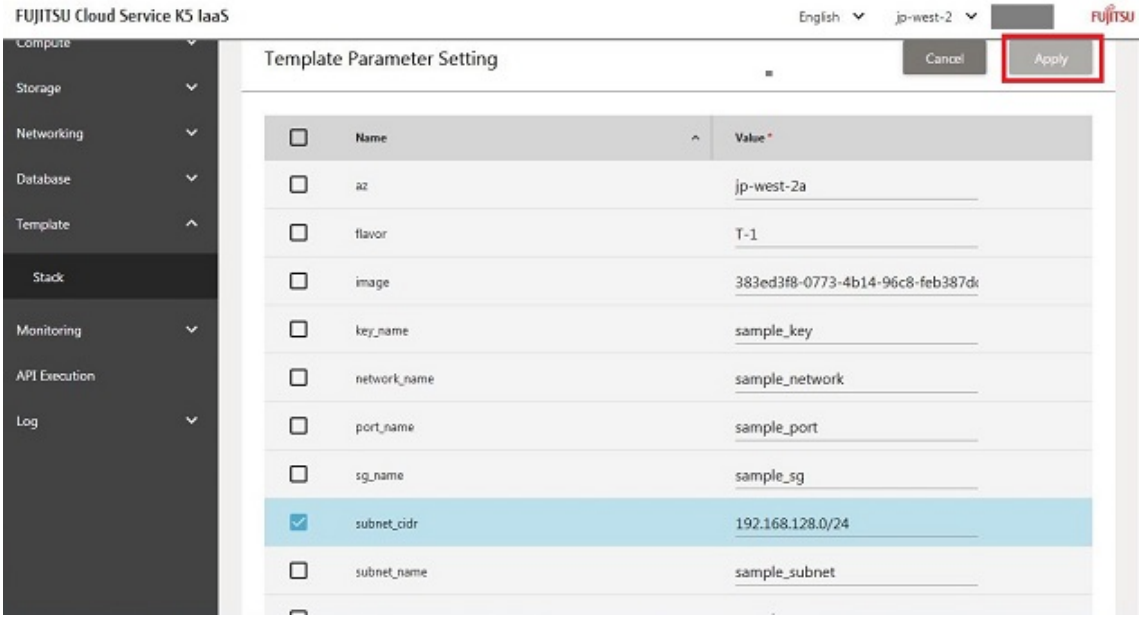
9. If there are any parameters you want to change, select their checkbox and input the new value.



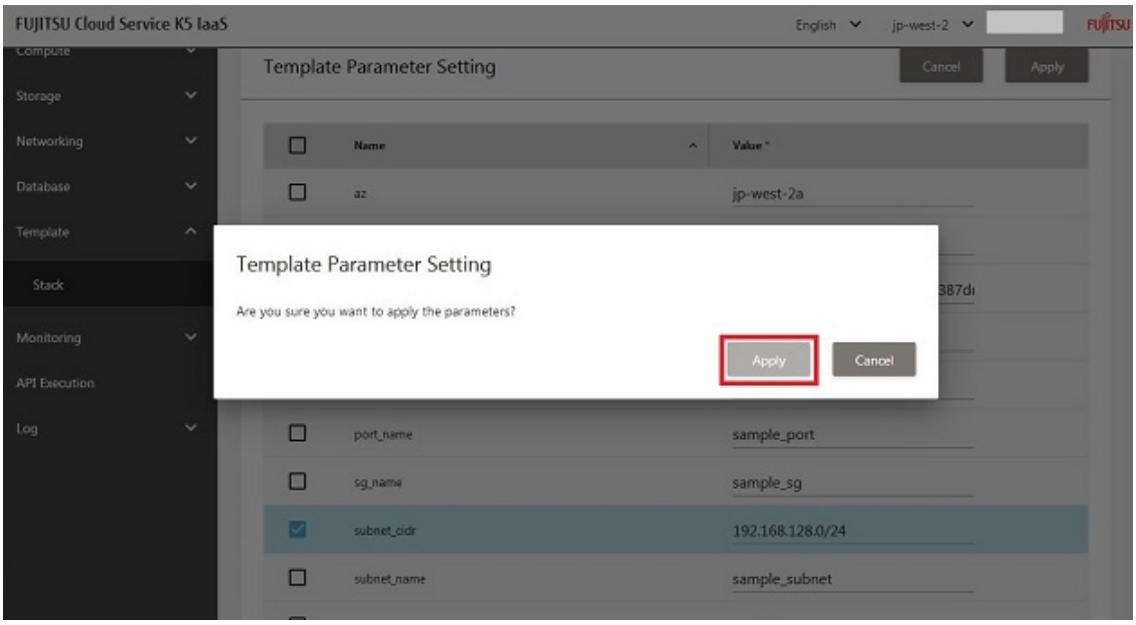
CAUTION

- If the value of a parameter is changed without checking the checkbox, the change will not be reflected and the original value will be used.
- In the "Sample system configuration - Example Heat template", the az (availability zone) is "jp-east-1a". Change the az based on the region you have selected. If there are any other parameters that require changing, please change them.

10. Clicking "Apply" on the upper right displays the "Template Parameter Setting" window.



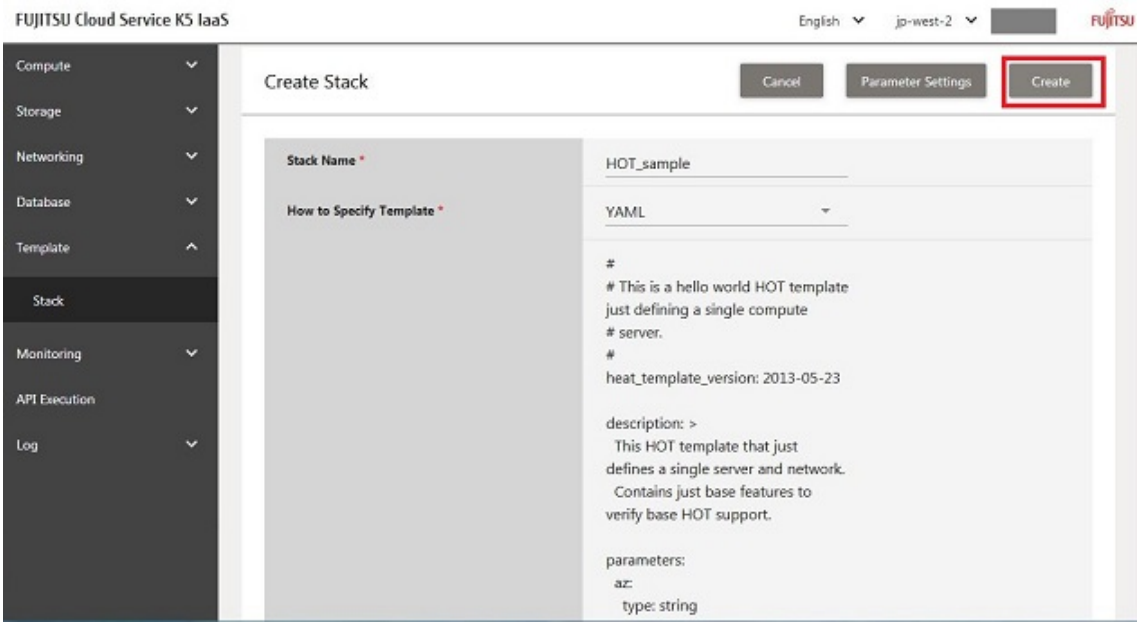
11.Clicking "Apply" returns you to the "Create Stack" window.



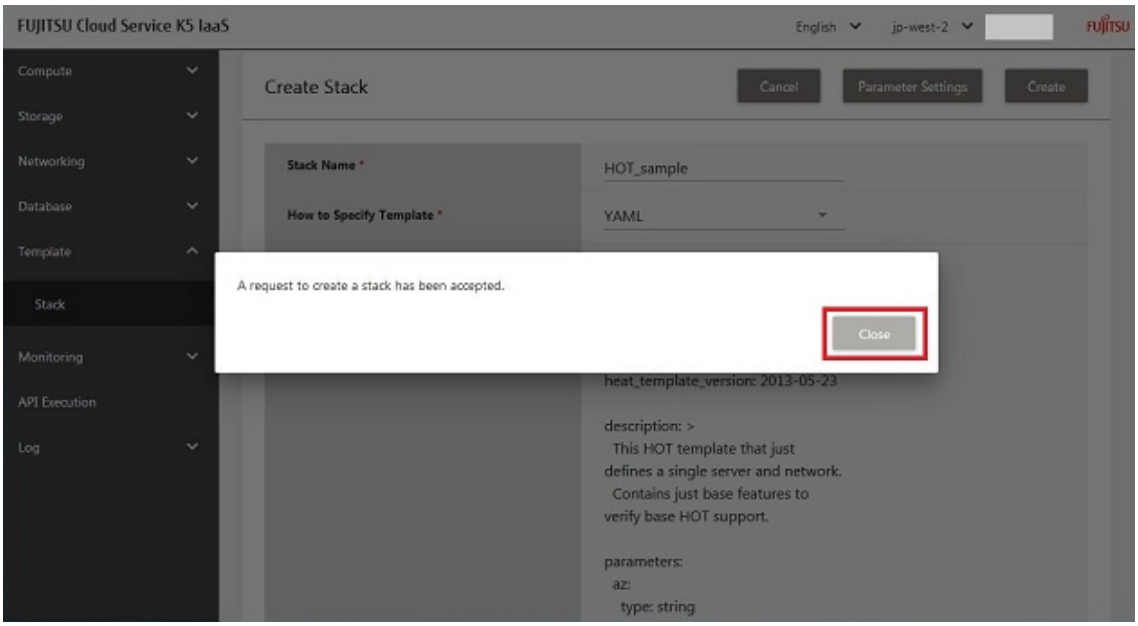
12.Clicking "Create" displays the request acceptance dialog.

Notice

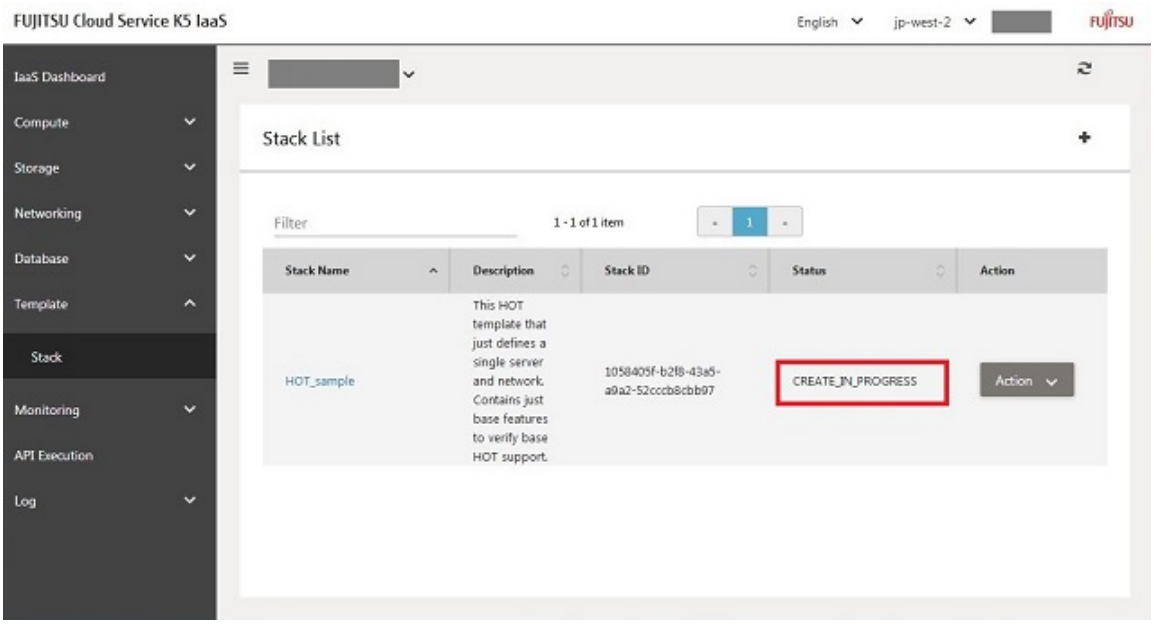
If there are any mistakes in the format, etc. of the template, an error will occur here.



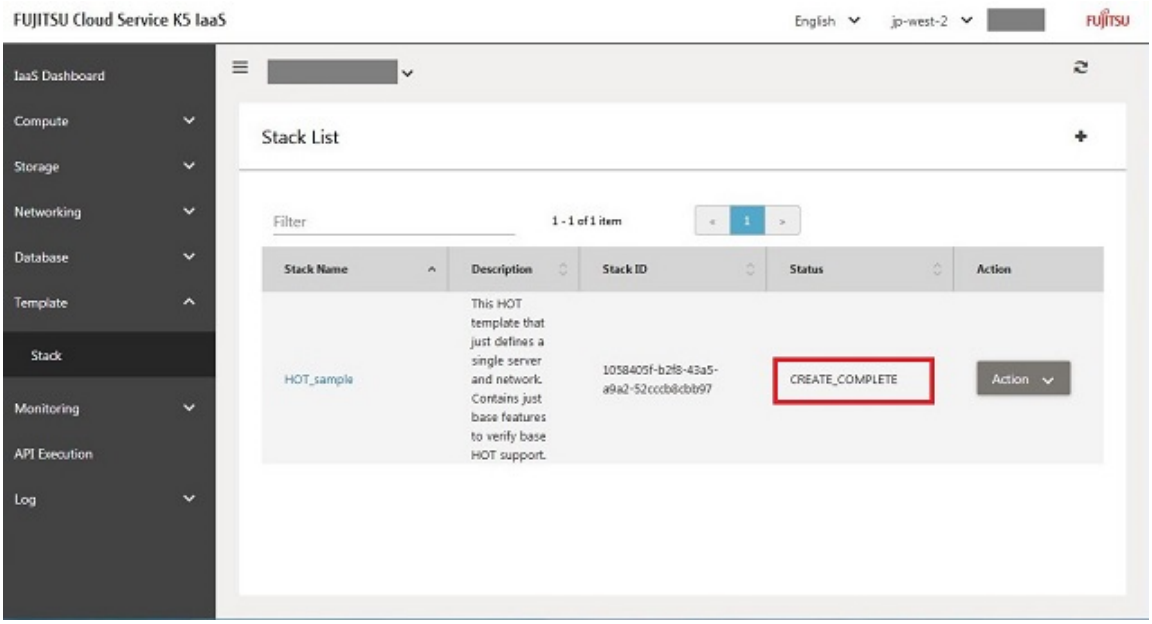
13.Clicking "Close" will return you to the "Stack list" window.



14.The status of the stack for which the creation request was accepted will be "CREATE_IN_PROGRESS".



15. After several minutes (*1), refresh the window, and if the status of the stack becomes "CREATE_COMPLETE" (*2), creation of the resources described in the Heat template is complete.



CAUTION

- *1: The time taken for creation varies depending on the state of the IaaS service and the content of the stack being created.
- *2: If the status becomes "CREATE_FAILED", it is possible to check the error content of stack creation using an API. Refer to "Handling errors".

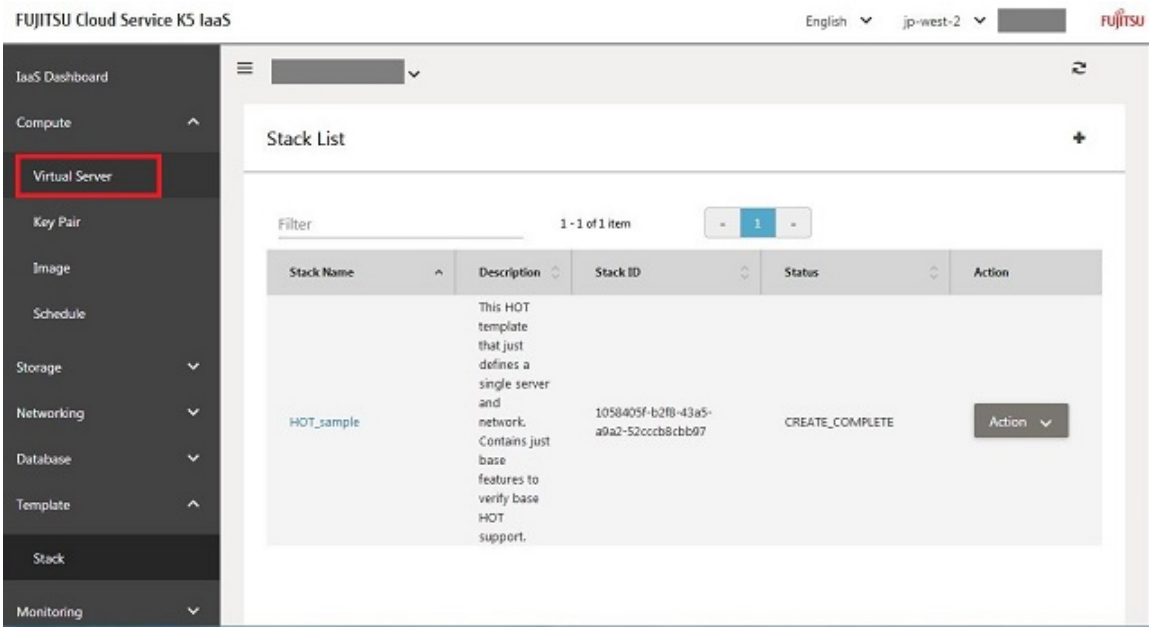
This completes creation.

2.3.2 Checking a stack

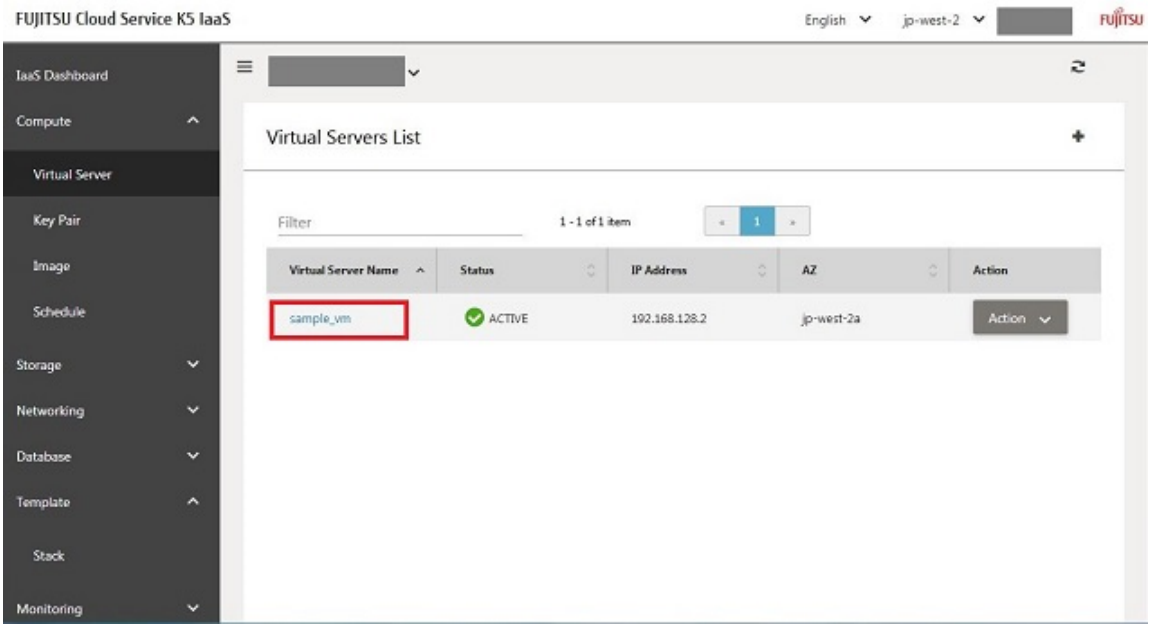
It is possible to confirm that the resources contained in the stack have been created from the IaaS Service Portal.

Procedure

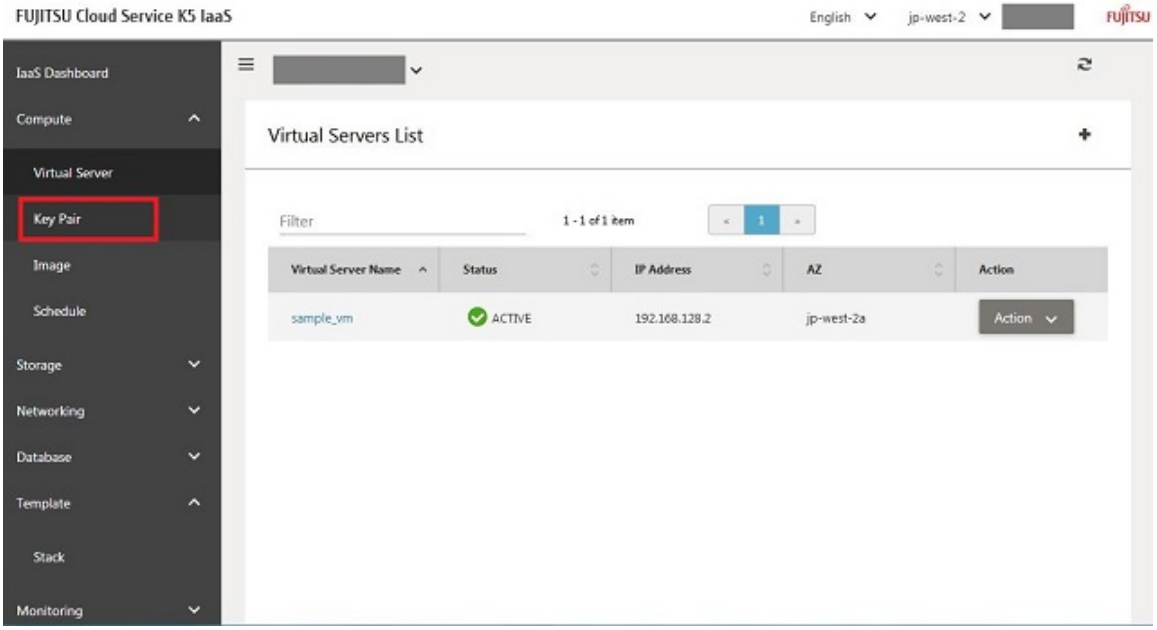
1. Click "Compute" > "Virtual Server".



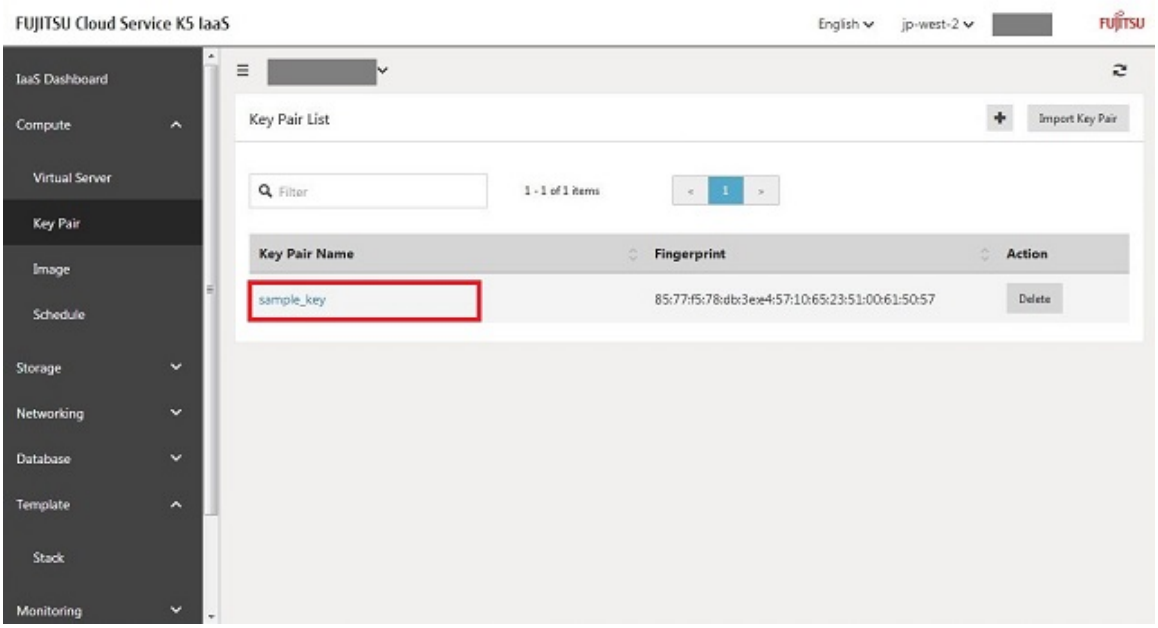
2. Confirm that the server (sample_vm) has been created.



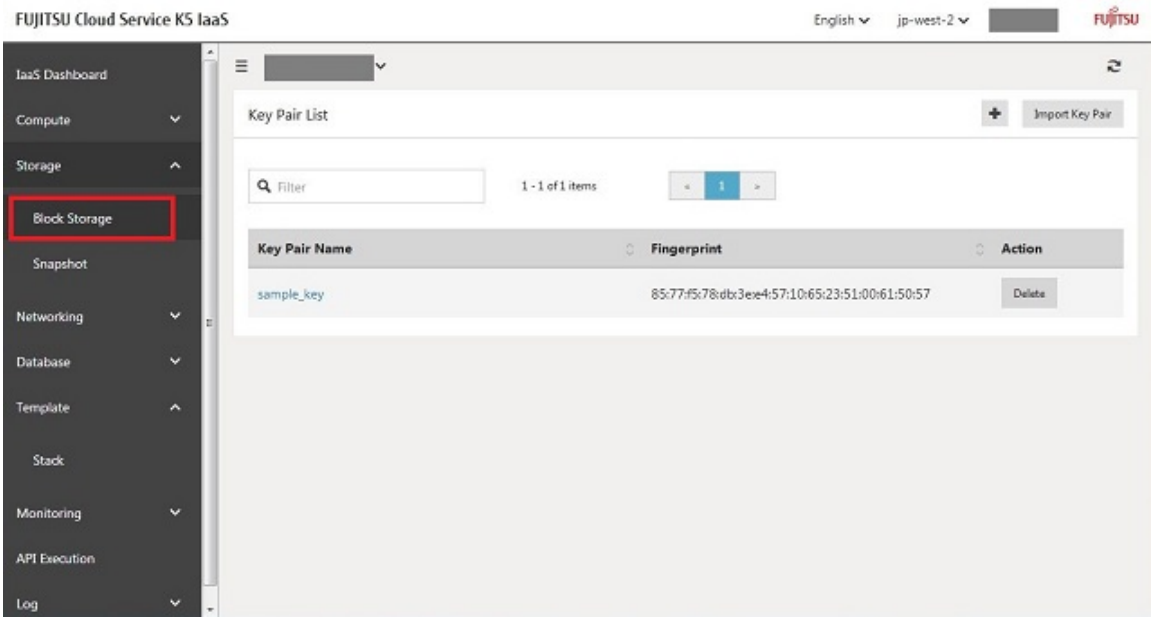
3. Click "Compute" > "Key Pair".



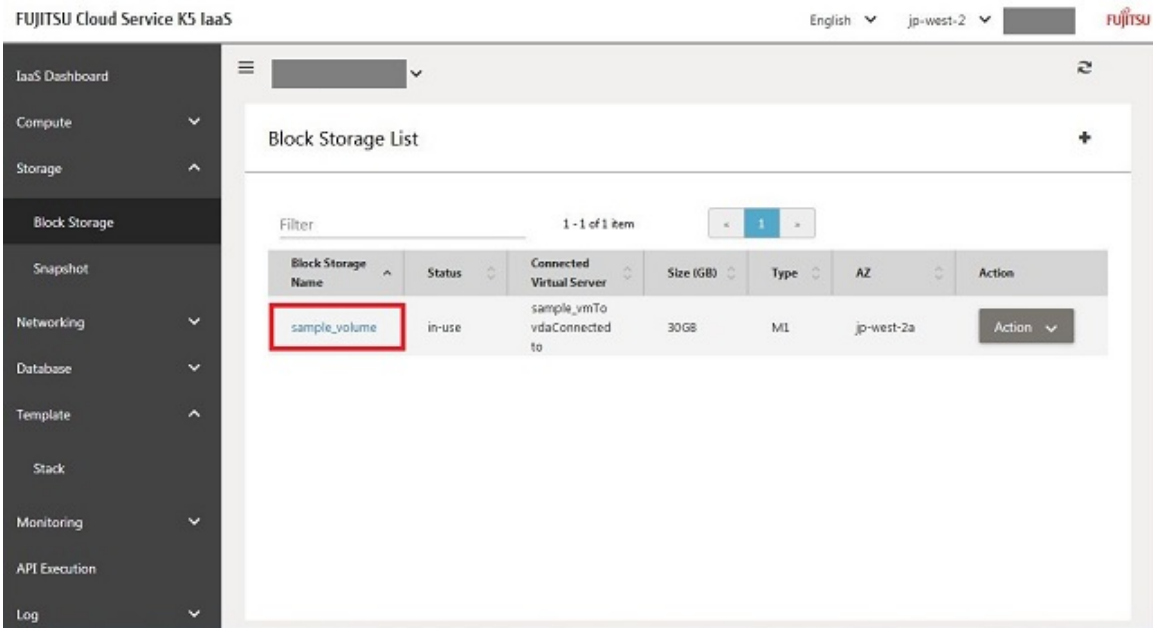
4. Confirm that the key pair (sample_key) has been created.



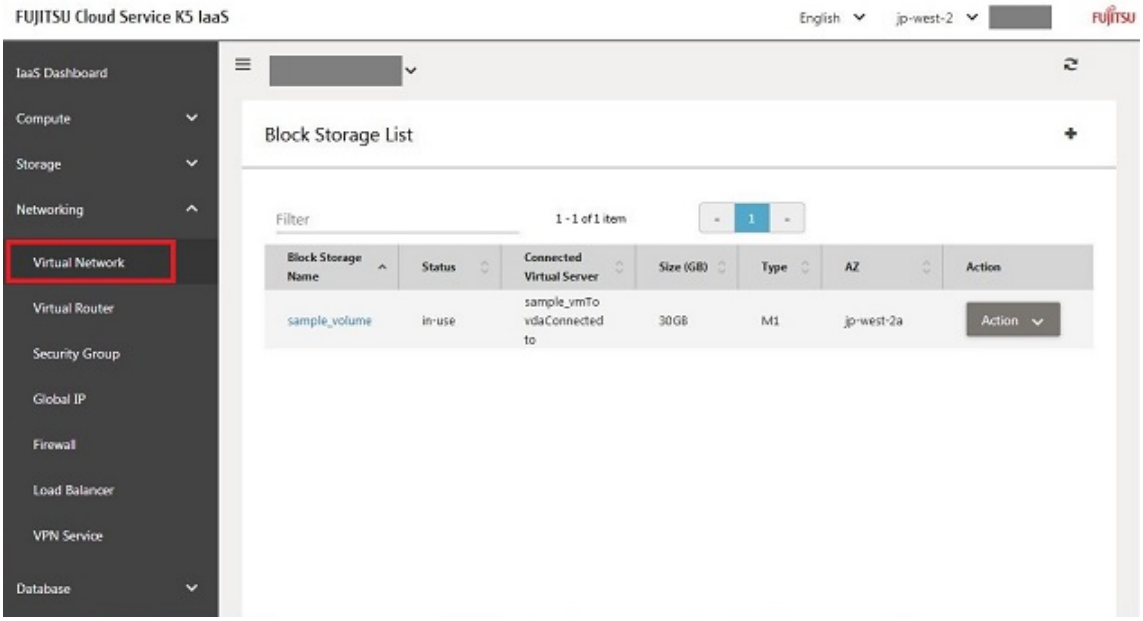
5. Click "Storage" > "Block Storage".



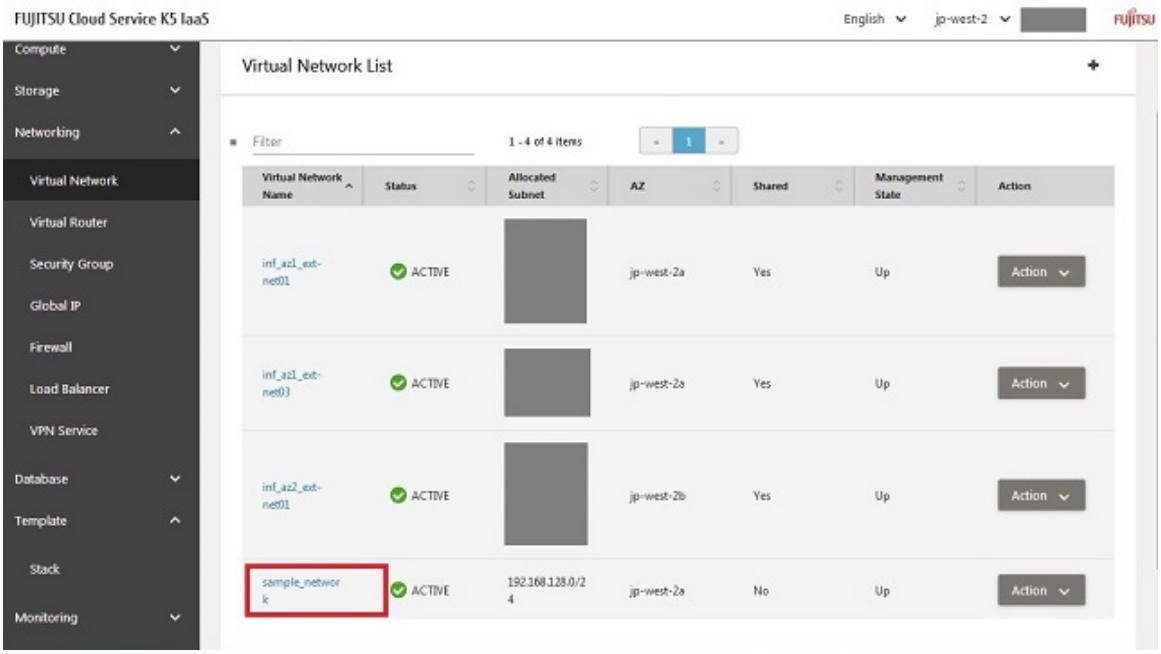
6. Confirm that the block storage (sample_volume) has been created.



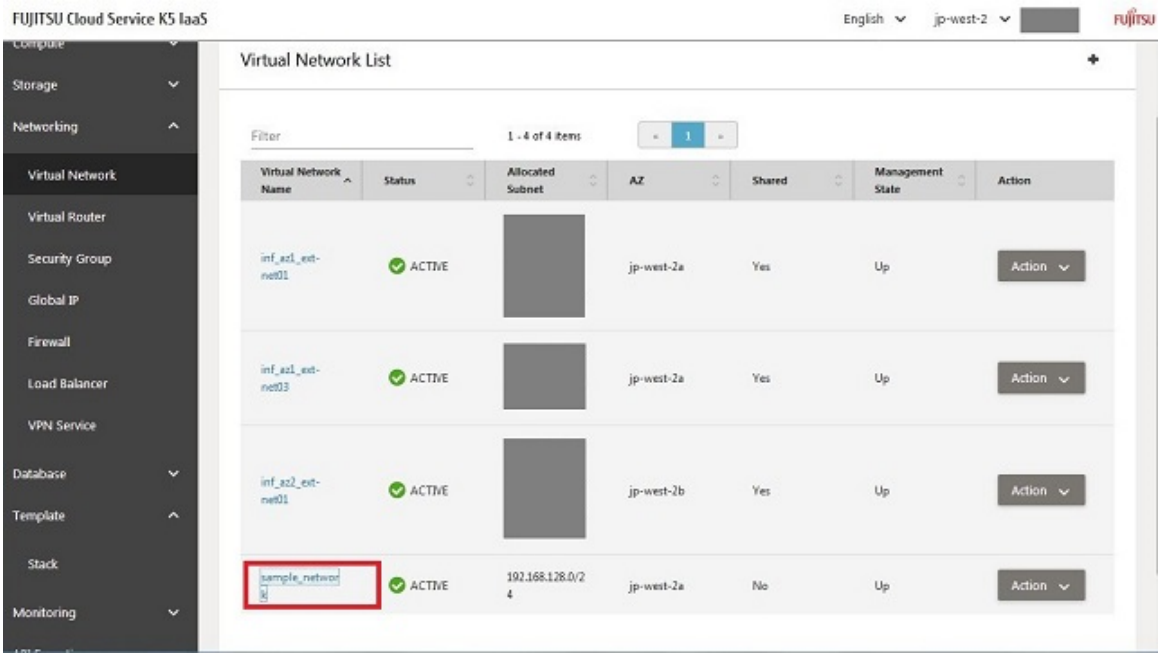
7. Click "Networking" > "Virtual Network".



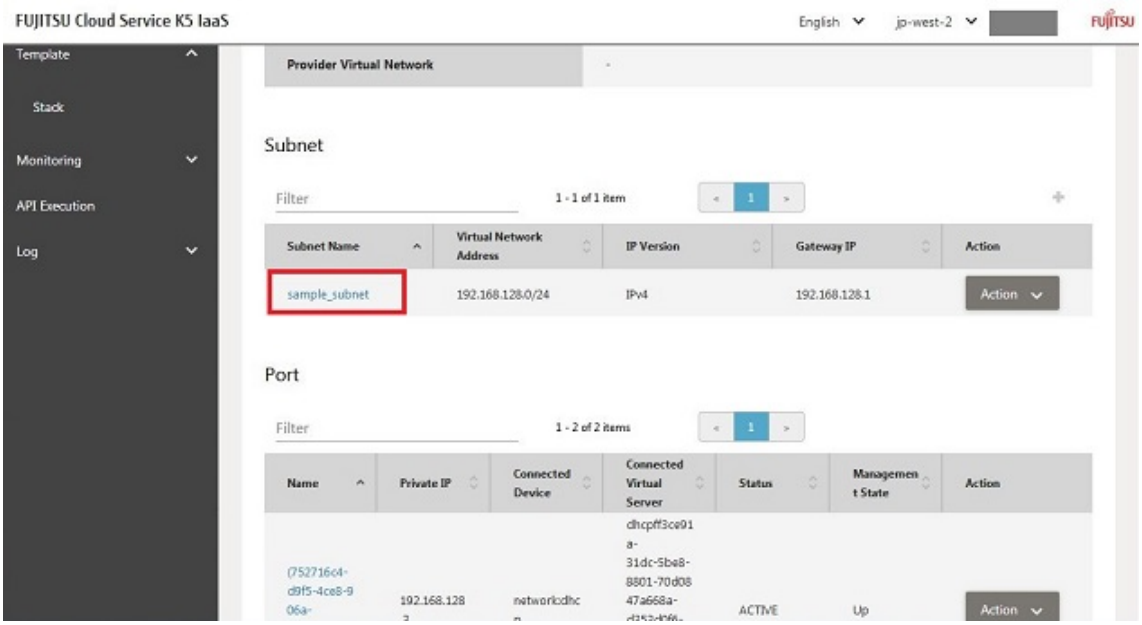
8. Confirm that the network (sample_network) has been created.



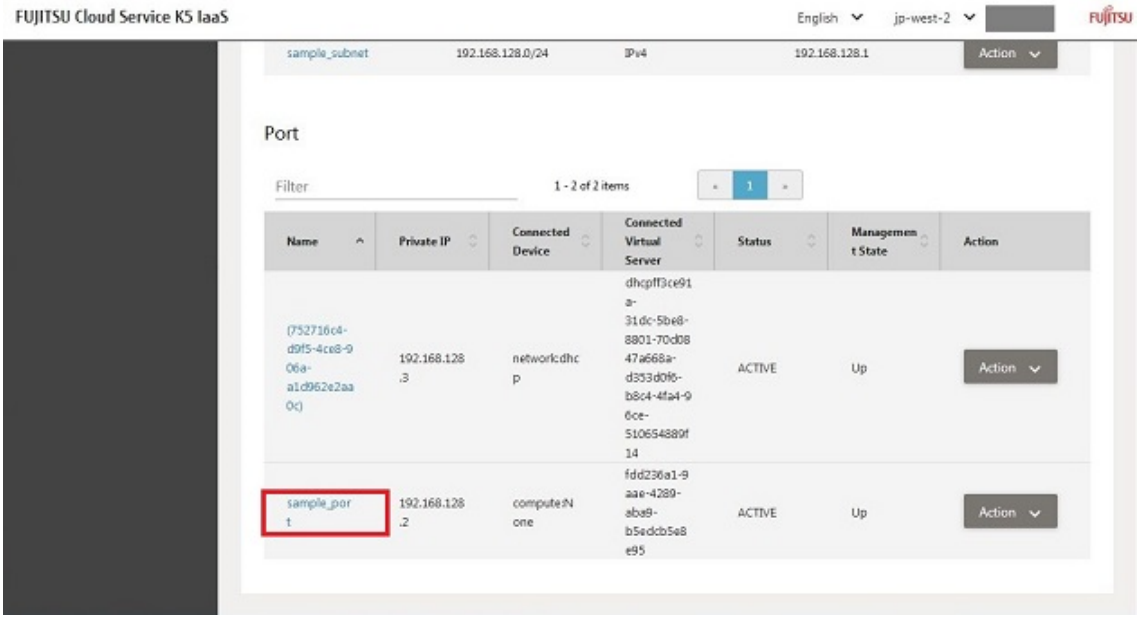
9. Click the created network.



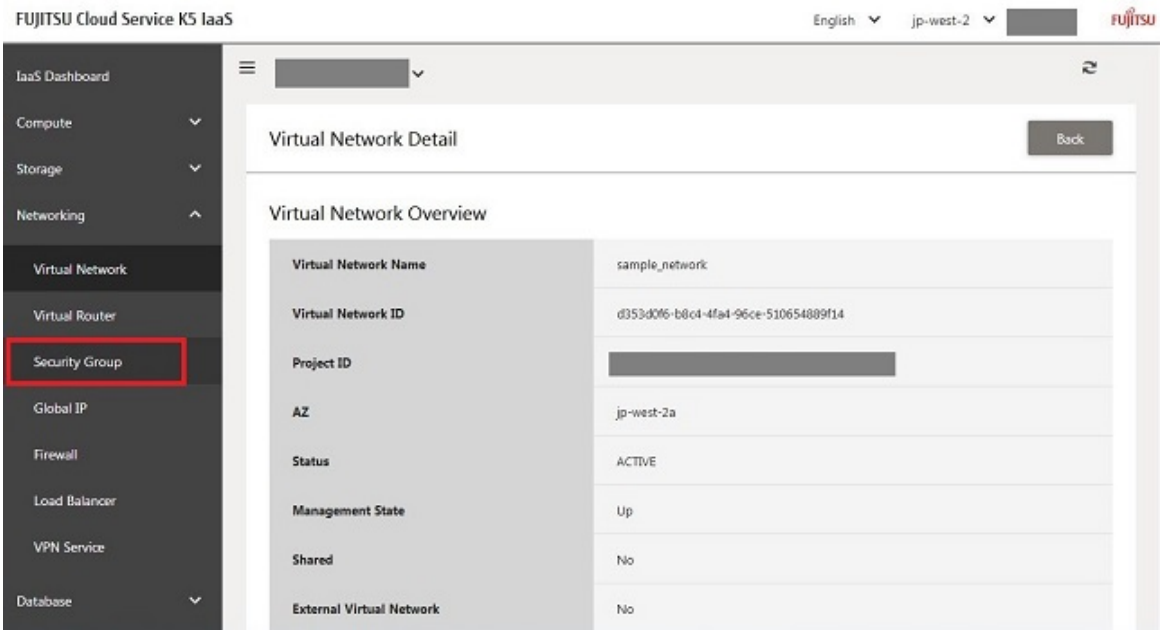
10.Confirm in the Subnet area that the subnet (sample_subnet) has been created.



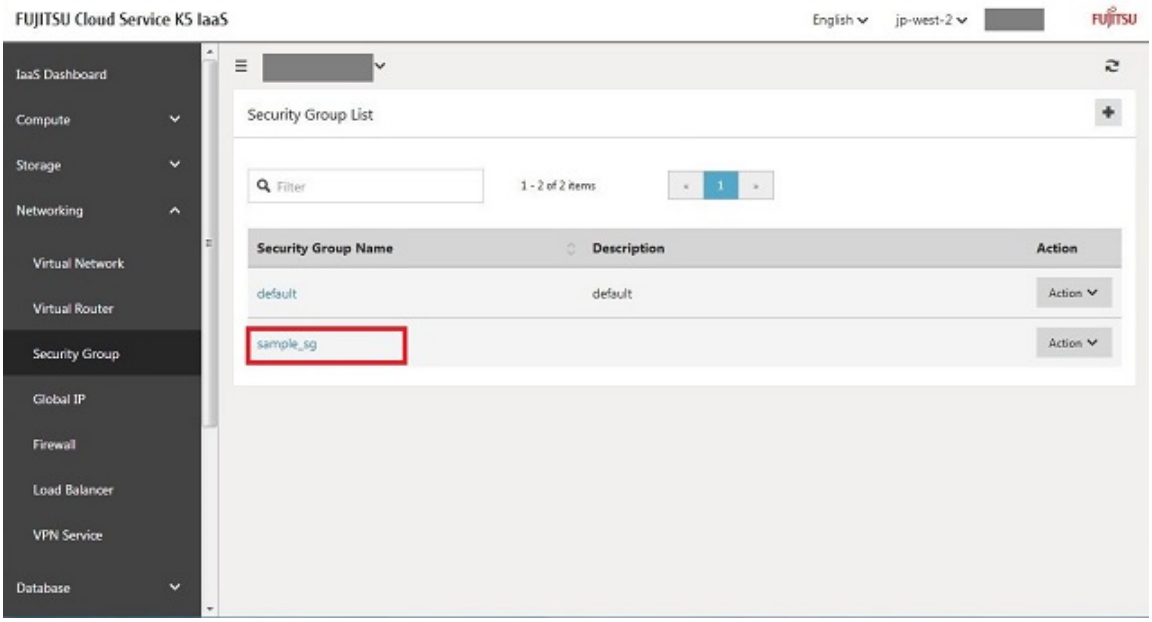
11.Confirm in the Port area that the port (sample_port) has been created.



12.Click "Networking" > "Security Group".



13.Confirm that the security group (sample_sg) has been created.



This completes the check.

2.3.3 Updating a stack

Here, as an example, we explain the procedure for updating a stack using a Heat template, using the IaaS service portal.

Procedure

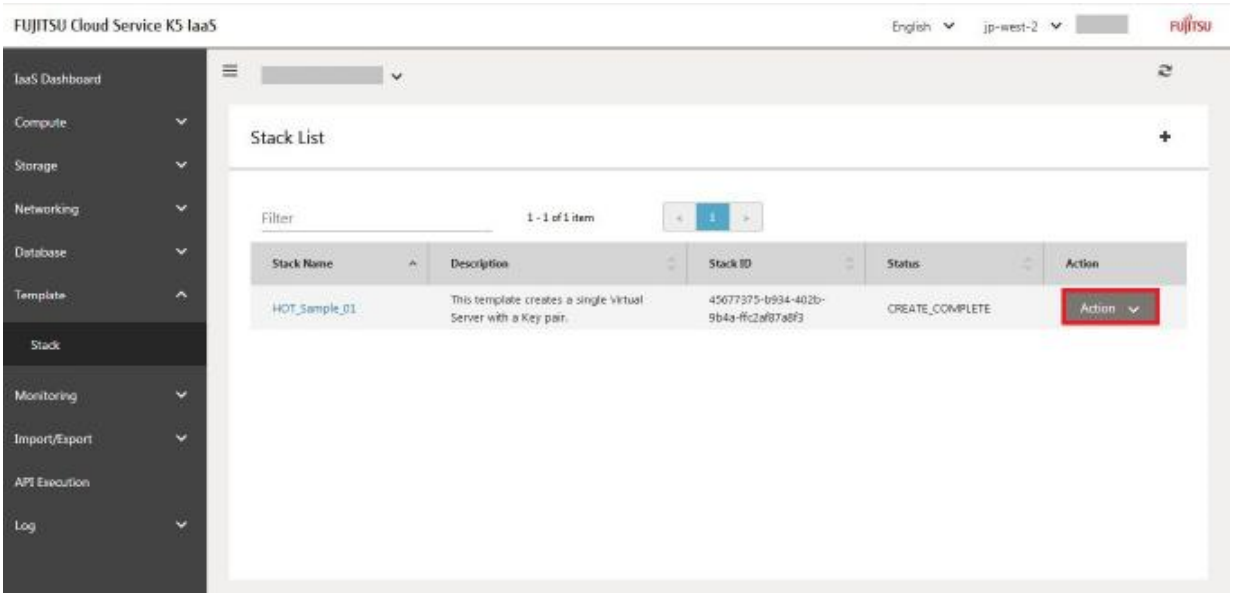
This performs update of a stack that has already been created in advance.

Of the example Heat templates used in this document, first, use "Example virtual server creation Heat template" (Creating_a_virtual_server.yaml) to create a resource.

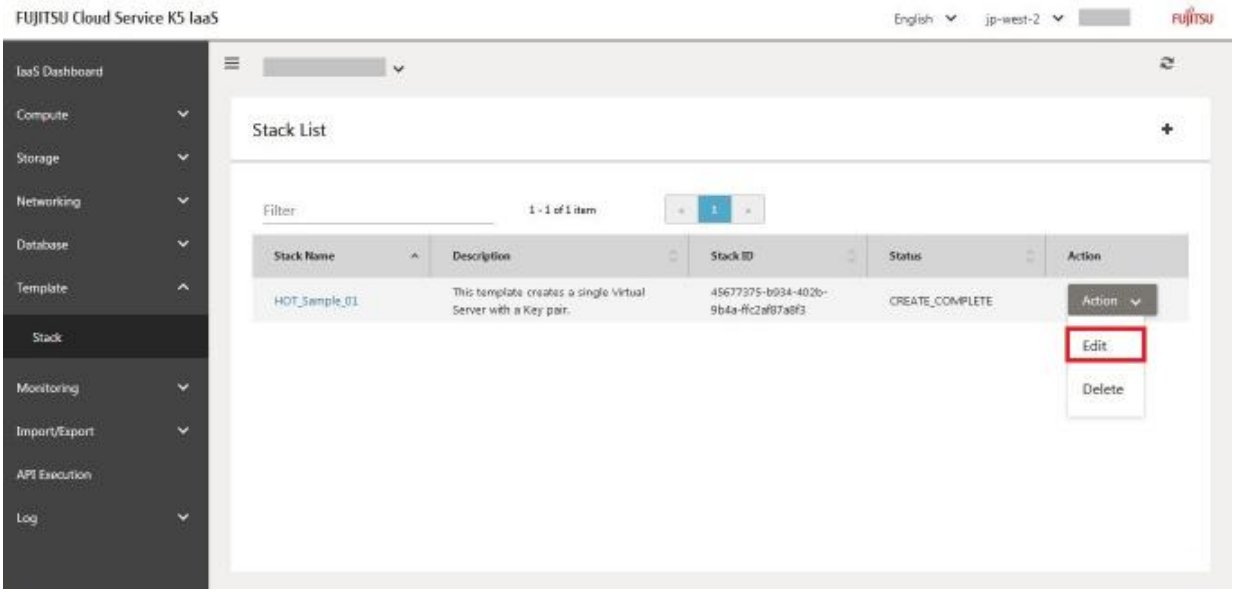
For the created stack, perform update using the example Heat template "Example Heat template for changing the flavor of a virtual server using stack update" (Updating_a_virtual_server.yaml) in this document.

The procedure for updating a stack is given below.

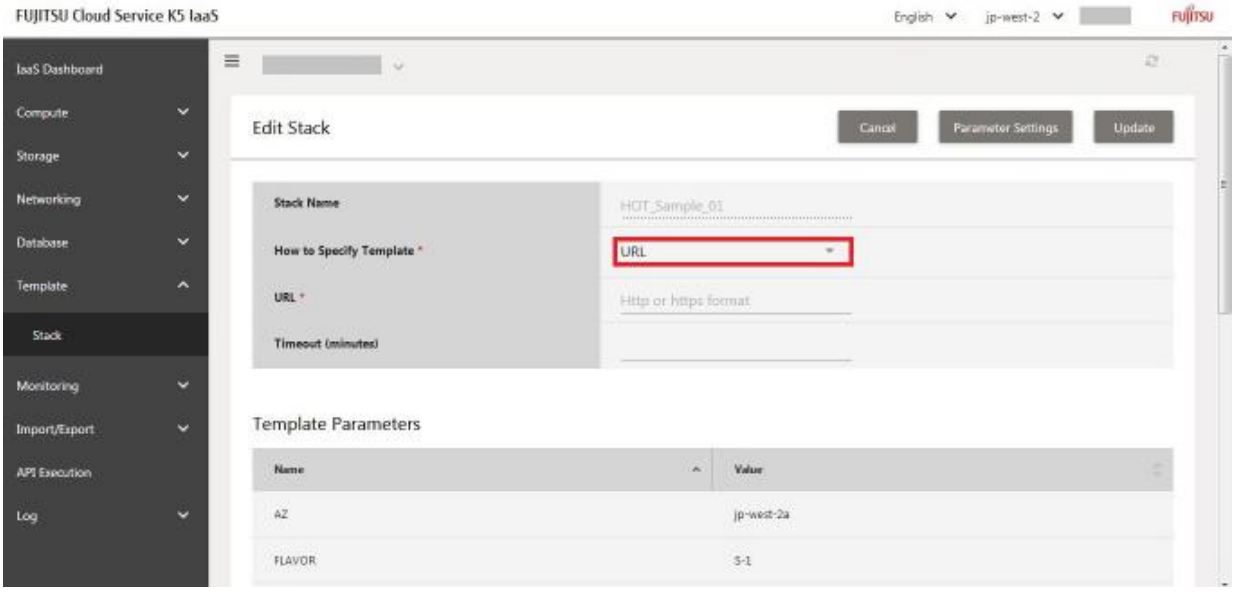
1. Display "Template" > "Stack List", and click "Action" of the stack to update.



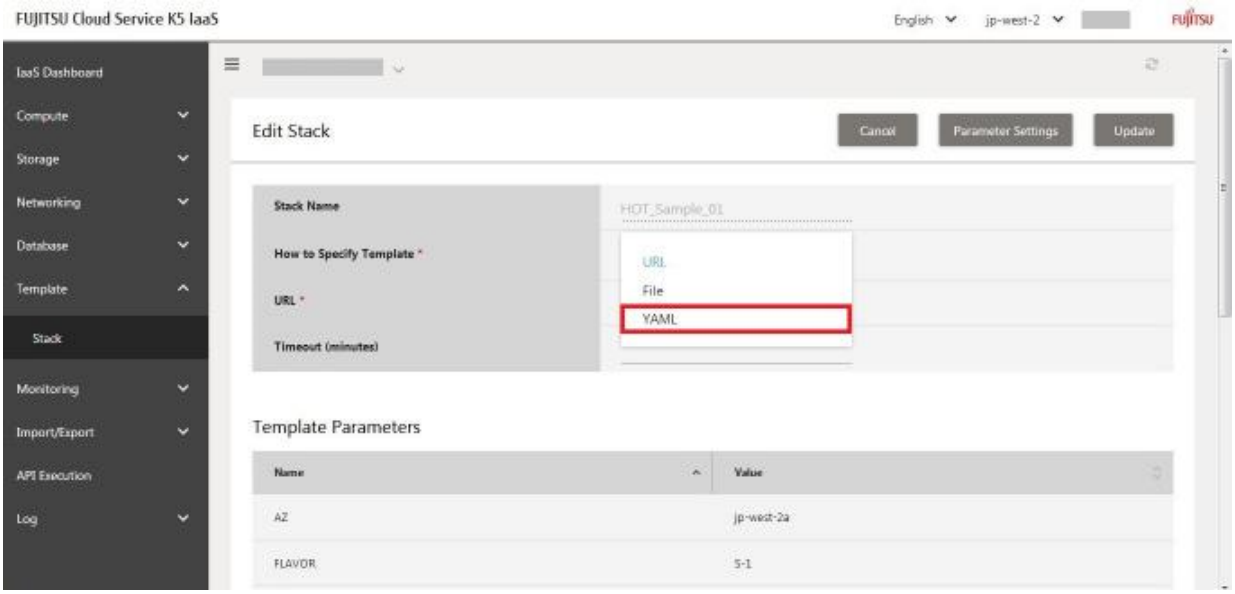
2. Click "Edit".



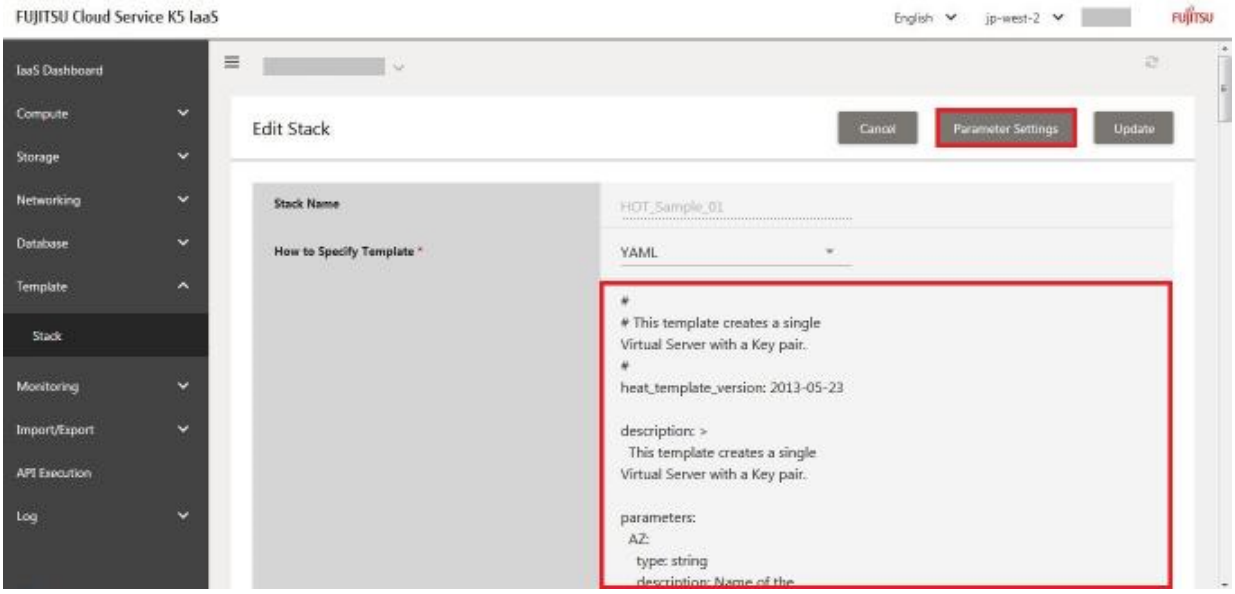
3. Click the "URL" of "Edit Stack" > "How to Specify Template".



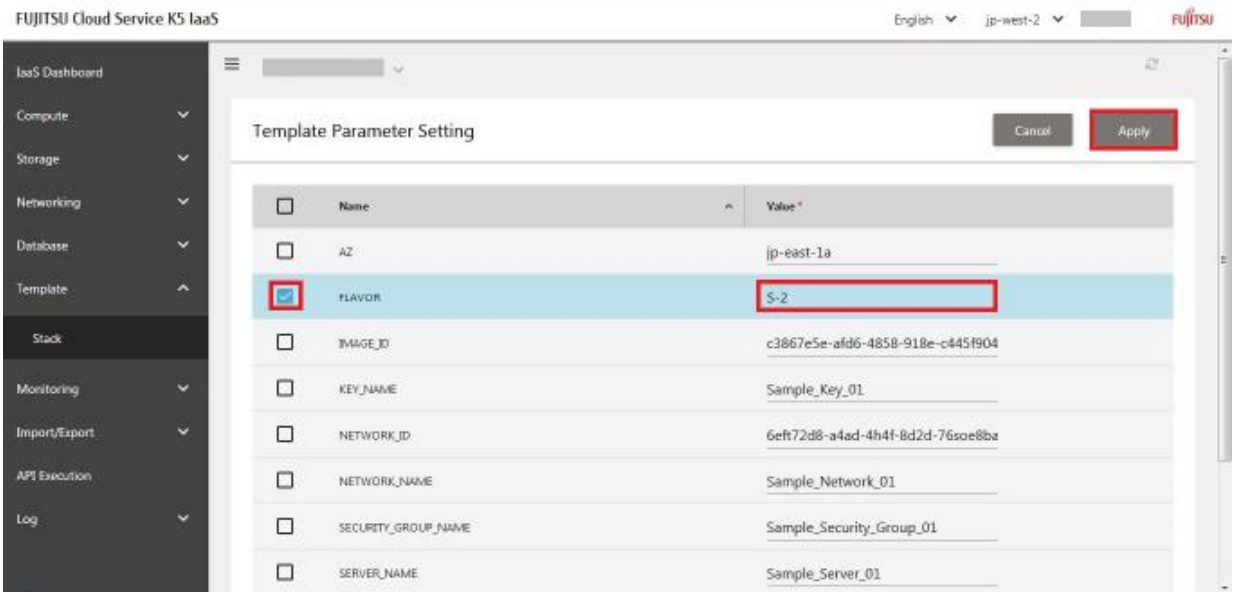
4. Select "YAML".



5. Enter the details of the YAML file from "Edit Stack" > "YAML Editor", and then click "Parameter Settings".



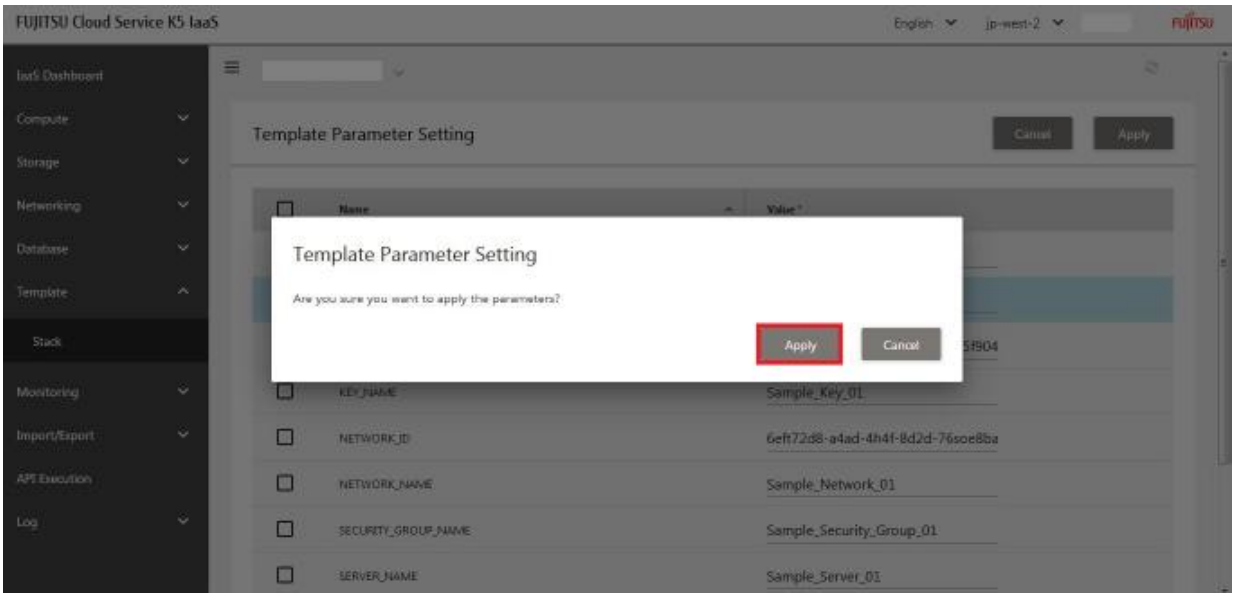
6. "Template Parameters Setting" is displayed.
   Place a check in the checkbox of the parameter to update, enter the updated value in "Value", and then click "Apply".



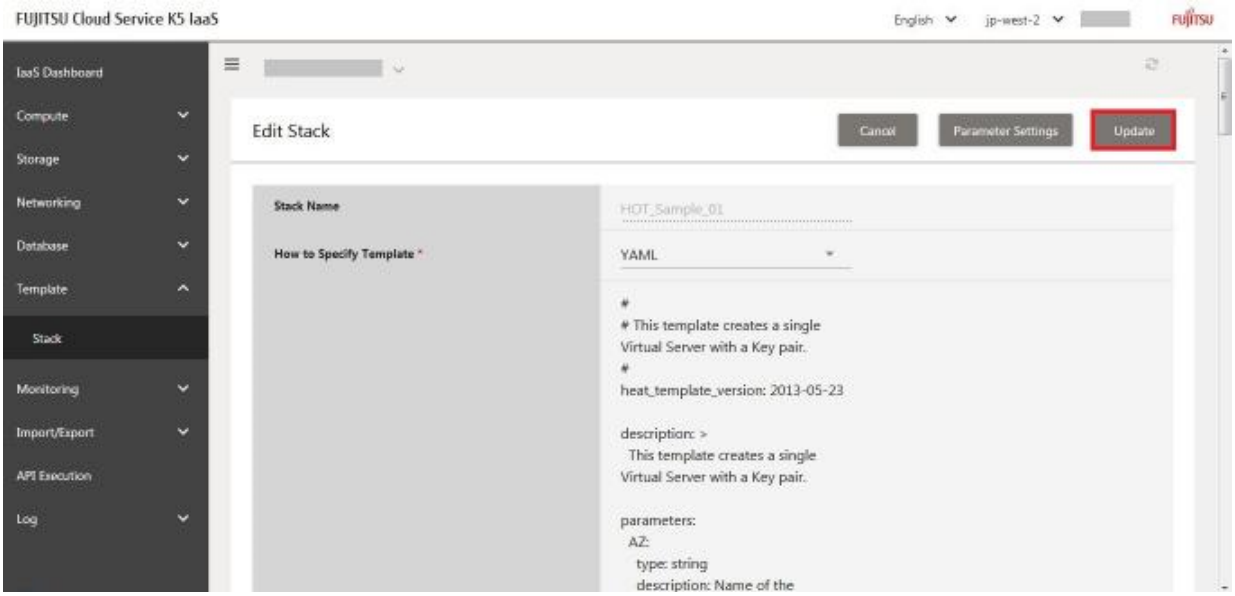
CAUTION

- If the value of a parameter is changed without checking the checkbox, the change will not be reflected and the original value will be used.

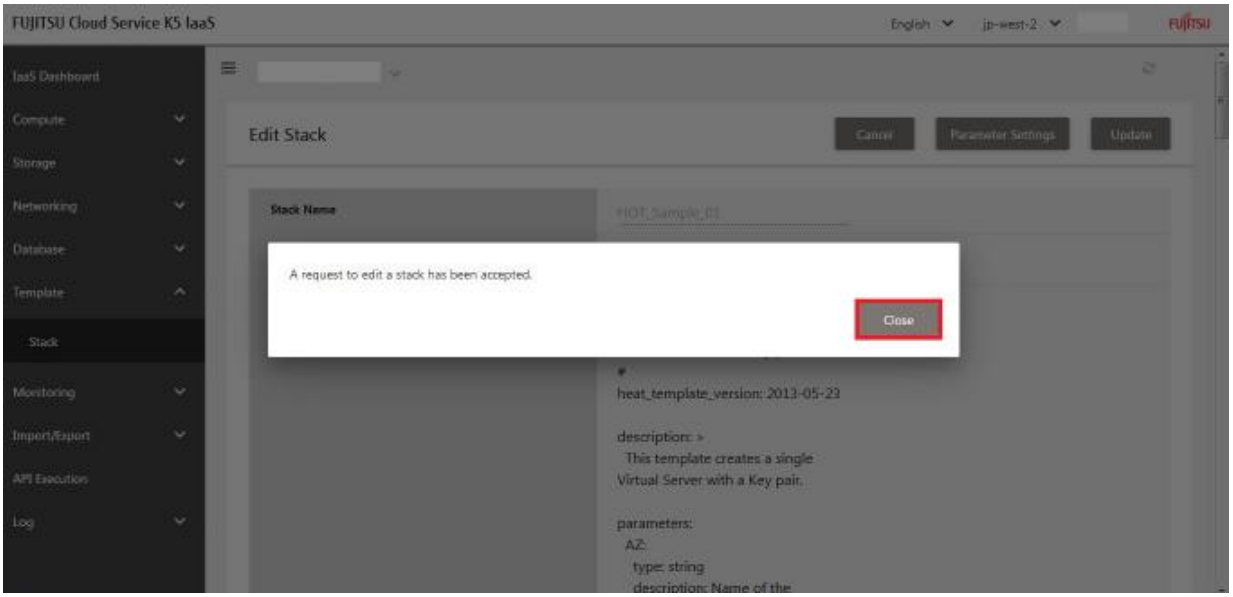
7. When "Are you sure you want to apply the parameters?" is displayed, click "Apply".



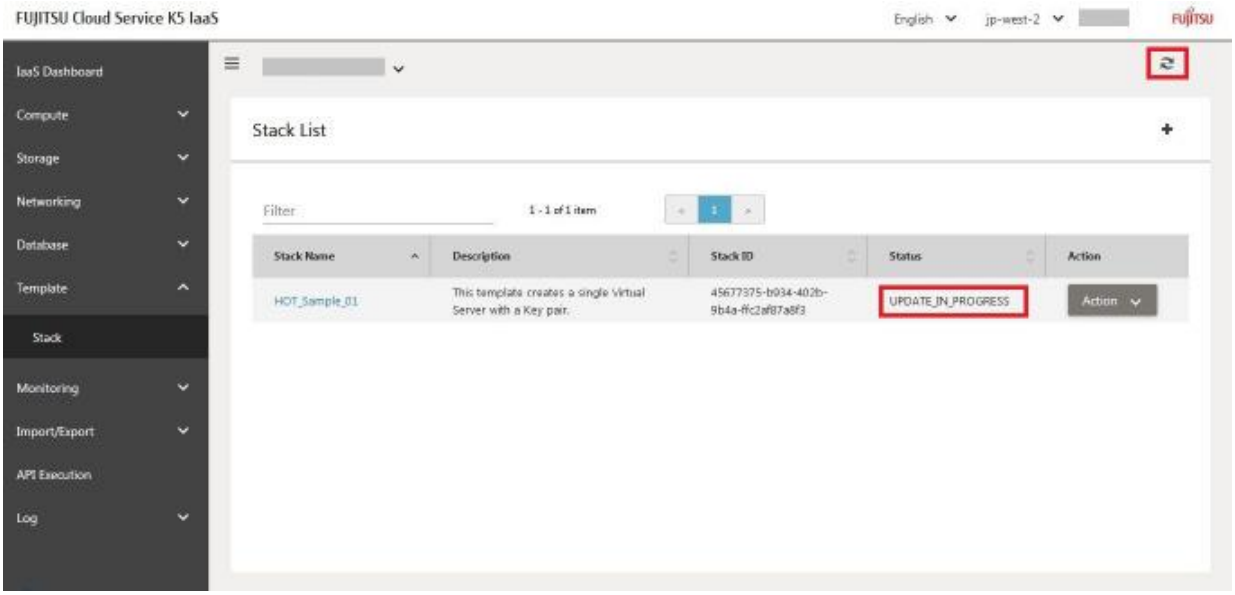
8. You are returned to the "Edit Stack" window.
   Click "Update".



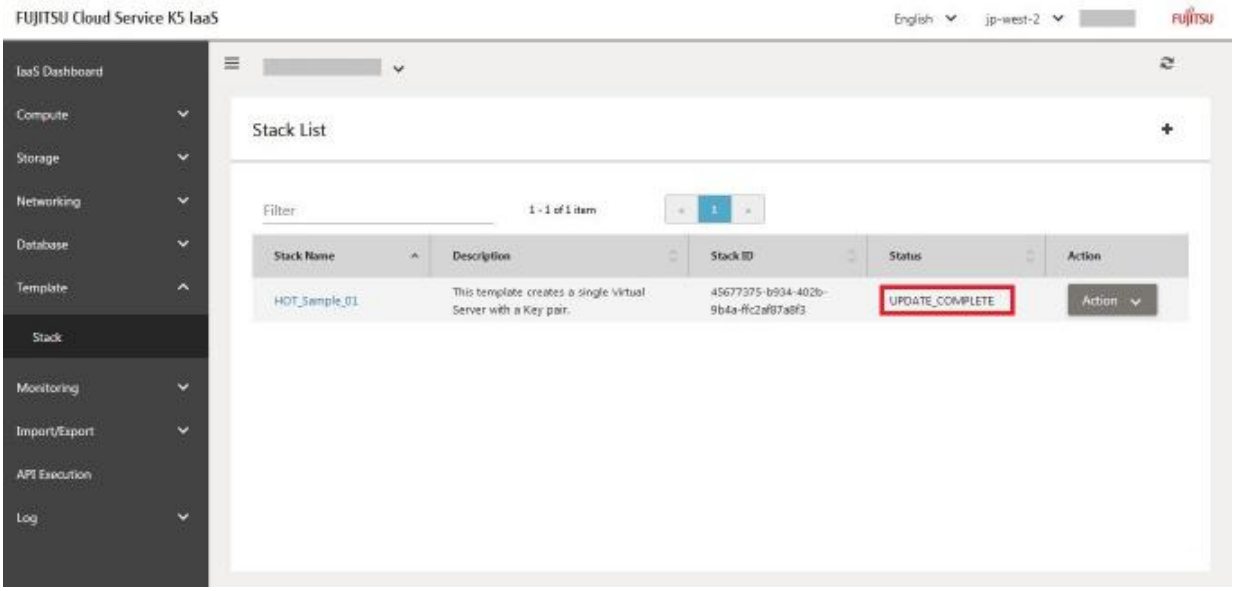
9. When "A request to edit a stack has been accepted." is displayed, click "Close".



10.You are returned to the "Stack List" window.
"UPDATE_IN_PROGRESS" is displayed for the "Status" of the updated stack.
Click the update button on the upper right, and confirm that the status is updated.
*It may take several minutes until the status is updated.



11.Confirm that "UPDATE_COMPLETE" (*1) is displayed for the status of the updated stack, and check the details of the resource to see that value of the parameter has been updated.



CAUTION

- *1: If the status becomes "UPDATE_FAILED", it is possible to use the API to check the details of the stack creation error. Refer to "Handling errors".

This ends the update operation for a stack.

Part 3: Heat Orchestration Template (HOT) format

Topics:

3.1 Template Structure

The Heat Orchestration Template (HOT) is defined in YAML format.
Below is an outline of the template structure.

```
heat_template_version: 2013-05-23

description: <description>

parameters:
  <parameters>

resources:
  <resources>

outputs:
  <outputs>
```

Section	Required	Description
heat_template_version	Yes	The HOT version. Specified as 2013-05-23.
description	No	Defines a description of the template.
parameters	No	Defines the input parameters. This is used when instantiating the template.
resources	No	Defines the resources.
outputs	No	Defines the output parameters. This can be used by users after instantiation of the template has been completed.

Notice

Heat templates use indenting to represent hierarchical structures. However, tab characters cannot be used for these indents.

parameters, resources, and outputs have the following relationships.

parameters	Defines the input parameters used when using a Heat template.
resources	Refers to parameters and can be used in resource creation
outputs	Refers to parameters and resources, and can be used as output information

```
parameters:
  Defines the input parameters used when using a Heat template
resources:
  Refers to parameters and can be used in resource creation
outputs:
  Refers to parameters and resources, and can be used as output information
```

3.2 Parameters Section

The parameters section defines the input parameters used when the template is instantiated.
For example, parameters specified by the user, such as name, password, and image ID.

As it is possible to specify the parameters when creating a stack, this means it is not necessary to define information such as passwords in the template. In the same way, by specifying the availability zone when creating a stack, it is possible to create stacks with the same configurations in multiple regions or availability zones using a single Heat template.

The type or default value is defined for each parameter, with the parameter name as the key.

```
parameters:
  <param name>:
    type: <string | number | json | comma_delimited_list>
    label: <human-readable name of the parameter>
    description: <description of the parameter>
    default: <default value for parameter>
    hidden: <true | false>
    constraints:
      <parameter constraints>
```

Element	Required	Description
<param name>	Yes	Defines the name of an input parameter.
type	Yes	Defines the data type of an input parameter. • string • number • json • comma_delimited_list
label	No	Defines the human-readable label.
description	No	Defines the human-readable description.
default	No	Defines the default value used when input of the parameter is omitted.
hidden	No	Defines whether to hide the parameter when a user requests information about a stack created from the template. This attribute can be used for passwords. If omitted, false will be used. • true • false
constraints	No	Defines the constraints of the parameter. Constraints are specified in the format described in the section below.

The defined parameters can be used in the "resources section" or the "outputs section".

(For the method for using parameters, refer to the "get_param" function that is explained later)

The following is an example of specifying the parameter (flavor) for the type (flavor) of the virtual server.

```
parameters:
...
  flavor:
    type: string
    description: Flavor for the server to be created
    default: T-1
...
resources:
  server:
    type: OS::Nova::Server
    properties:
      flavor: { get_param: flavor }
...
```

In the above example, T-1 is specified as the default value for the type (flavor) of the virtual server. If the flavor parameter is not specified when a stack is created, the flavor of created virtual servers will be T-1.

3.2.1 Parameter Constraints

This section explains the format of constraints in the parameters section.
Using constraints it is possible to define restrictions for input parameters.

As an example, the following type of usage becomes possible.

・Setting of limits on the length of user name strings and the characters that can be used.

```
constraints:
  - <constraint type>: <constraint definition>
    description: <constraint description>
```

Element	Required	Description
<constraint type>	Yes	Specifies the type of constraint. ・length ・range ・allowed_values ・allowed_pattern
<constraint definition>	Yes	Specifies the definition of a constraint in the format corresponding to its type.
description	No	Defines the message that is displayed for the user when a constraint is violated. If omitted, the default message will be used.

<constraint type> and <constraint definition> are described in the section below.

3.2.1.1 length

Specifies constraints for a string parameter. The minimum and maximum number of characters can be defined.

```
length: { min: <lower limit>, max: <upper limit> }
```

Element	Required	Description
min	No	Specifies the minimum number of characters. Specify min, max, or both.
max	No	Specifies the maximum number of characters. Specify min, max, or both.

3.2.1.2 range

Specifies constraints for a number parameter. The minimum and maximum value can be defined.

```
range: { min: <lower limit>, max: <upper limit> }
```

Element	Required	Description
min	No	Specifies the minimum value. Specify min, max, or both.

Element	Required	Description
max	No	Specifies the maximum value. Specify min, max, or both.

3.2.1.3 allowed_values

Specifies constraints for a string or number parameter. It specifies a regular expression that specified values must match.

```
allowed_values: [ <value>, <value>, ... ]
```

```
allowed_values:
  - <value>
  - <value>
  - ...
```

3.2.1.4 allowed_pattern

Specifies constraints for a string parameter. It specifies a regular expression that specified values must match.

```
allowed_pattern: <regular expression>
```

3.2.1.5 Example of specified constraints

```
parameters:
  user_name:
    type: string
    label: User Name
    description: User name to be configured for the application
    constraints:
      - length: { min: 6, max: 8 }
        description: User name must be between 6 and 8 characters
      - allowed_pattern: "[A-Z]+[a-zA-Z0-9]*"
        description: User name must start with an uppercase character
  instance_type:
    type: string
    label: Instance Type
    description: Instance type for compute instances
    constraints:
      - allowed_values:
        - S-1
        - S-2
        - S-4
```

3.2.2 Pseudo parameters

There are parameters provided by Heat separately to the input parameters. These parameters can be referenced using the embedded function get_param, in the same manner as for the parameters defined using the template.

Parameter name	Description
OS::stack_name	The name of the stack.
OS::stack_id	The ID used to identify stacks.

The following is an example of a template that makes the name of the virtual server name the same as the name of the stack.

```
resources:
  server:
    type: OS::Nova::Server
    properties:
      name: { get_param: "OS::stack_name" }
```

```
...
```

Notice If the stack name is "sample_stack" then the name of the created server will also be "sample_stack".

3.3 Resources Section

The resources section defines the resources that make up a stack deployed from the template (for example, compute instances, networks, storage volumes).

```
resources:
  <resource ID>:
    type: <resource type>
    properties:
      <property name>: <property value>
    metadata:
      <resource specific metadata>
    depends_on: <resource ID or list of ID>
    deletion_policy: <deletion policy>
```

*Depending on the type specified in <resource type>, the definable elements vary. For the <property name> and <property value> that can be used for each <resource type>, refer to "Resource type details".

Element	Required	Description
<resource ID>	Yes	Defines a unique resource ID in the template.
type	Yes	Specifies the resource type. For example, OS::Nova::Server, etc. For the specifiable resource types, refer to "Supported resource types".
properties	No	Specifies a list of resource properties.
<property name>	No	Specifies a property name. The name of the properties defined for the resource type can be used. For details, refer to Resource type details.
<property value>	No	Specifies a value using the data type corresponding to a property. This can be specified directly, or via Intrinsic Functions. For details, refer to Resource type details.
metadata	No	Specifies the metadata of a resource.
depends_on	No	Defines the dependency relationship with other resources. For details, refer to Resource dependencies.
deletion_policy	No	Specifies the deletion policy of resources. When *Delete* is specified, the resource entity will be removed on deletion. When *Retain* is specified, the resource entity will not be removed on deletion. If omitted, *Delete* will be used. • *Delete* • *Retain*

3.3.1 Example definition of resources

```
resources:
  my_instance:
    type: OS::Nova::Server
    properties:
      flavor: S-2
      image: F18-x86_64-cfntools
```

3.3.2 Resource dependencies

depends_on can be used to define resources with one or more dependencies.
Specify the resource ID for the value.

Example definition of a single dependency:

```
resources:
  server1:
    type: OS::Nova::Server
    depends_on: server2

  server2:
    type: OS::Nova::Server
```

Example definition of multiple dependencies:

```
resources:
  server1:
    type: OS::Nova::Server
    depends_on: [ server2, server3 ]

  server2:
    type: OS::Nova::Server

  server3:
    type: OS::Nova::Server
```


Notice

- When depends_on is specified, creation of the dependent resource will wait for the specified resource to be created first.
- In "Example definition of multiple dependencies", creation of server1 will start after server2 and server3 have been created.

3.3.3 Referring to information

Usually systems are configured with many resources being coordinated with other resources.

For example, virtual servers need system volumes, and virtual servers are connected to the network.

In Heat templates it is possible to directly describe the resource information (ID, etc.) of dependencies, however, when creating resources simultaneously as in "Sample system configuration - Example Heat template", it is not possible to describe that information.

In this type of situation, using functions enables defined resource information and parameters to be reflected on the "resources" of created resources.

* For details on functions, refer to "Intrinsic Functions".

The following is an example of how to refer to information, using the get_resource function to set the connection of the virtual server to the network, when creating the virtual server and the network at the same time.

```
resources:
  network:
```

```
    type: OS::Neutron::Net

  subnet:
    type: OS::Neutron::Subnet
    properties:
      network_id: { get_resource: network }

  port:
    type: OS::Neutron::Port
    properties:
      network_id: { get_resource: network }
      fixed_ips:
        - subnet_id: { get_resource: subnet }

  server:
    type: OS::Nova::Server
    properties:
      networks: ["port": {get_resource: port} ]
```

3.4 outputs Section

The outputs section defines output parameters that should be available to users.
The parameters defined here can be referenced by users as stack information. For example, output parameters are used for the IP address of deployed instances, or, the URL of a web application deployed to a stack.

```
outputs:
  <parameter name>:
    description: <description>
    value: <parameter value>
```

For details of the method to use "outputs" for viewing, refer to "Referring to outputs".

Element	Required	Description
<parameter name>	Yes	Defines the name of a unique output parameter in the outputs section.
description	No	Defines the description of an output parameter.
value	No	Defines the value of an output parameter. get_ attr can be used to obtain resource information. If omitted, a null character is used as the default value of an output parameter.

The example below defines the output parameter instance_ip used to obtain the IP address of a compute resource.

```
resources:
  my_instance:
    type: OS::Nova::Server
outputs:
  instance_ip:
    description: IP address of the deployed compute instance
    value: { get_attr: [my_instance, first_address] }
```

3.5 Intrinsic Functions

The embedded functions described below can be used in the HOT template.

3.5.1 get_param

The get_param function obtains the value of input parameters defined in the parameters section.

```
get_param: <parameter name>

or

get_param:
  - <parameter name>
  - <key/index 1>
  - <key/index 2>
  - ...
```

Element	Required	Description
<parameter name>	Yes	Specifies the name of an input parameter defined in the parameters section.
<key/index 1,2,..>	No	Specifies the key or index for obtaining data, when the input parameter defined in the parameters section is a list or hash.

Example specification of the get_param function:

```
parameters:
  instance_type:
    type: string
    label: Instance Type
    description: Instance type to be used.
  server_data:
    type: json

resources:
  my_instance:
    type: OS::Nova::Server
    properties:
      flavor: { get_param: instance_type }
      metadata: { get_param: [ server_data, metadata ] }
      key_name: { get_param: [ server_data, keys, 0 ] }
```

Example input parameter values:

```
{"instance_type": "S-1",
{"server_data": {"metadata": {"foo": "bar"},
                 "keys": ["a_key","other_key"]}}}
```

properties	Value obtained using get_param
flavor	"S-1"
metadata	{"foo": "bar"}
key_name	"a_key"

3.5.2 get_attr

The get_attr function references the attribute value of an instantiated resource.
The name of an attribute defined for the resource type must be specified.

```
get_attr:
  - <resource ID>
  - <attribute name>
```

```
  - <key/index 1>
  - <key/index 2>
  - ...
```

Element	Required	Description
<resource ID>	Yes	Resources SectionSpecifies a resource ID defined in the Resources Section.
<attribute name>	Yes	Specifies the name of an attribute for which you want to reference the value. The name of the attribute defined for the resource type can be used.
<key/index 1,2,..>	No	Specifies the key or index for obtaining data, when the attribute is a list or hash.

Example specification of the get_attr function:

```
resources:
  my_instance:
    type: OS::Nova::Server

outputs:
  instance_ip:
    description: IP address of the deployed compute instance
    value: { get_attr: [my_instance, first_address] }
  instance_private_ip:
    description: Private IP address of the deployed compute instance
    value: { get_attr: [my_instance, networks, private, 0] }
```

In this example, the networks attribute holds data such as the following.

```
{"public": ["2001:0db8:0000:0000:0000:ff00:0042:8329", "1.2.3.4"],
"private": ["10.0.0.1"]}
```

In the above example, the "instance_private_ip" value in the outputs section is "10.0.0.1".

3.5.3 get_resource

The get_resource function references another resource defined in the same template. The return value is the resource ID defined for each resource type.
For example, an IP address is returned for floating IP resources.

```
get_resource: <resource ID>
```

Element	Required	Description
<resource ID>	Yes	Resources SectionSpecifies a resource ID defined in the Resources Section.

3.5.4 str_replace

The str_replace function replaces strings.

```
str_replace:
  template: <template string>
  params: <parameter mappings>
```

Element	Required	Description
template	Yes	Specifies the string that is the replacement source.

Element	Required	Description
params	Yes	Specifies the mapping of the string for replacement. Other functions such as get_attr can be used.

Example specification of the str_replace function (1):

```
resources:
  my_instance:
    type: OS::Nova::Server

outputs:
  Login_URL:
    description: The URL to log into the deployed application
    value:
      str_replace:
        template: http://host/MyApplication
        params:
          host: { get_attr: [ my_instance, first_address ] }
```

In the above example, assuming the value returned by get_attr: [my_instance, first_address] is "10.0.0.1", the value of the output parameter "Login_URL" will be "http://10.0.0.1/MyApplication".

Example specification of the str_replace function (2):

```
parameters:
  DBRootPassword:
    type: string
    label: Database Password
    description: Root password for MySQL
    hidden: true

resources:
  my_instance:
    type: OS::Nova::Server
    properties:
      # general properties ...
      user_data:
        str_replace:
          template: |
            #!/bin/bash
            echo "Hello world"
            echo "Setting MySQL root password"
            mysqladmin -u root password $db_rootpassword
            # do more things ...
          params:
            $db_rootpassword: { get_param: DBRootPassword }
```

In the above example, in relation to user_data input for the compute resource, the str_replace function is used to replace the string "$db_rootpassword" with the value of the input parameter "DBRootPassword" of the template.

3.5.5 get_file

The get_file function references string data.
For example, use this function when referencing content of scripts or configuration files in a non-Heat format.

```
get_file: <content key>
```

Element	Required	Description
<content key>	Yes	Specifies a key for referencing string data. When executing from the REST API of Heat, the string data mapped in files of the request parameters is referenced.

```
resources:
  my_instance:
    type: OS::Nova::Server
    properties:
      # general properties ...
      user_data:
        get_file: my_instance_user_data.sh
  my_other_instance:
    type: OS::Nova::Server
    properties:
      # general properties ...
      user_data:
        get_file: http://example.com/my_other_instance_user_data.sh
```

In the above example, when executing from the REST API, the string data mapped in files of the request parameters is referenced.

```
"files" : {
  "my_instance_user_data.sh" : "<my_instance_user_data.sh file content(*)>",
  "http://example.com/my_other_instance_user_data.sh" : "<my_other_instance_user_data.sh
 file content(*)>"
}
```

(*1) To specify escaped characters, use '\'. For example: return -> \n, " ->\"

3.5.6 resource_facade

The resource_facade function is used by the resource template.
In the resource template, properties values can be retrieved. Use this function to retrieve other values.

```
resource_facade: <data type>
```

Element	Required	Description
<data type>	Yes	Specifies the data type for retrieving values. • *metadata* • *deletion_policy*

Example definition of the parent template:

```
resources:
  my_server:
    type: my_actual_server.yaml
    metadata:
      key: value
      some: more stuff
```

Example definition of the resource template "my_actual_server.yaml":

```
resources:
  _actual_server_:
    type: OS::Nova::Server
    metadata: { resource_facade : metadata }
```

Part 4: Supported resource types

Topics:

4.1 Supported resource types

The following resource types can be specified in the Resources section.
Refer to "Resource type details" for details on each resource type. Refer to "Resource type properties" for a list of the properties that can be used with each resource.

Services	Resource Types	Note
Auto Scaling	FCX::AutoScaling::AutoScalingGroup	
	FCX::AutoScaling::LaunchConfiguration	
	FCX::AutoScaling::ScalingPolicy	
Telemetry	OS::Ceilometer::Alarm	
	OS::Ceilometer::CombinationAlarm	
Block Storage	OS::Cinder::Volume	
	OS::Cinder::VolumeAttachment	
Compute	OS::Nova::Server	
	OS::Nova::ServerGroup	
	OS::Nova::KeyPair	
Network	OS::Neutron::Firewall	
	OS::Neutron::FirewallPolicy	
	OS::Neutron::FirewallRule	
	OS::Neutron::FloatingIP	
	OS::Neutron::FloatingIPAssociation	
	OS::Neutron::Net	
	OS::Neutron::Port	
	OS::Neutron::Router	
	OS::Neutron::RouterInterface	
	OS::Neutron::SecurityGroup	
	OS::Neutron::Subnet	
	FCX::Neutron::NetworkConnector	
	FCX::Neutron::NetworkConnectorEndpoint	
	FCX::Neutron::NetworkConnectorEndpointConnection	
Expandable Load Balancing	FCX::ExpandableLoadBalancer::LoadBalancer	
Database	FCX::Database::DBInstance	
	FCX::Database::DBSubnetGroup	
	FCX::Database::DBParameterGroup	
Object Storage	OS::Swift::Container	

Part 5: Resource type details

Topics:

5.1 Resource type details

5.1.1 Auto Scaling

5.1.1.1 FCX::AutoScaling::AutoScalingGroup

5.1.1.1.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: FCX::AutoScaling::AutoScalingGroup
    properties:
      AvailabilityZones: [Value, Value, ...]
      Cooldown: Number
      HealthCheckGracePeriod: Integer
      HealthCheckType: String
      LaunchConfigurationName: String
      LoadBalancerNames: [Value, Value, ...]
      MaxSize: Integer
      MinSize: Integer
      Tags: [{"Value": String, "Key": String}, {"Value": String, "Key": String}, ...]
      VPCZoneIdentifier: [String, String, ...]
```

5.1.1.1.2 Notes

- When an abnormal instance detected using the health check of Load Balancer is automatically recovered, and scaling by CPU load, etc., is used at the same time, the event for which the abnormal instance was detected may be recorded in the stack event during the scale-in operation. It is not necessary to address this as it is an instance-related event that will be deleted by scaling in.
- Instances that have already been created cannot be incorporated into the AutoScalingGroup.
- To assign the following information to a virtual server, it is necessary to have a virtual router connected to the network to which the virtual server is connected.
  - Host name (Computer name)
  - Administrator password
  - Authentication key (keypair)

5.1.1.1.3 Properties

AvailabilityZones

- Not Implemented.

  Specify an appropriate value to support future feature enhancements.
- Required property.
- Type: List

Cooldown

- Cooldown period, in seconds.

  Set sufficient time (in seconds) for execution of scaling to be completed.

  The time of stack creation will be recorded as the initial scaling execution time.

  Alarms that occur during the standby time leading up to the next scale are discarded.

When using scaling with multiple alarms, of the values obtained using the estimation formula described later in this document, set the largest value for this item.

When abnormal instances detected using the health check of Load Balancer are automatically recovered, and scaling by CPU load, etc., is used at the same time, set this item only, and do not set Cooldown of FCX::AutoScaling::ScalingPolicy.

The estimation formula is as follows, with the largest value obtained using this formula to be set.

- When using scale-out:

  (Time required to create one instance x ScalingAdjustment value)
  - FCX::AutoScaling::Time specified for HealthCheckGracePeriod of AutoScalingGroup
  - OS::Ceilometer::Alarm period x OS::Ceilometer::Alarm evaluation_periods
  - Number of Grade(ELB VM) subnets x Number of Grades (ELB VM) x 10 (*1)
  - 60
- When using scale-in:

  Time required to delete one instance x (ScalingAdjustment value x -1)
  - FCX::AutoScaling::Time specified for HealthCheckGracePeriod of AutoScalingGroup
  - OS::Ceilometer::Alarm period x OS::Ceilometer::Alarm evaluation_periods
  - Number of Grade(ELB VM) subnets x Number of Grades (ELB VM) x 10 (*1)
  - 60
- When an abnormal instance detected using the health check of Load Balancer is automatically recovered

  (Time required to create one instance x ScalingAdjustment value)
  - FCX::AutoScaling::Time specified for HealthCheckGracePeriod of AutoScalingGroup
  - (Time required to delete one instance x 5)
  - FCX::AutoScaling::Time specified for HealthCheckGracePeriod of AutoScalingGroup
  - (Time required to create one instance x FCX::AutoScaling::MinSize value of AutoScalingGroup)
  - FCX::AutoScaling::Time specified for HealthCheckGracePeriod of AutoScalingGroup
  - OS::Ceilometer::Alarm period x OS::Ceilometer::Alarm evaluation_periods
  - Number of Grade(ELB VM) subnets x Number of Grades (ELB VM) x 10 (*1) x 3
  - 60


Notice

- "Number of Grade(ELB VM) subnets" is the number of subnets specified in Subnets in the properties of FCX::ExpandableLoadBalancer::LoadBalancer in the template.
- "Number of Grades (ELB VM)" is the number of instances registered under the name specified in LoadBalancerName of FCX::ExpandableLoadBalancer::LoadBalancer in the template when a stack is created.
- *1: This is rough estimate for times of low loads, so this will vary depending on communication performance.

- Optional property, defaults to 0.
- Type: Number

HealthCheckGracePeriod

- The amount of time until LoadBalancer starts the health check, after an instance is created.
- Optional property, defaults to 0.
- Type: Integer

HealthCheckType

- The type of health check.

Only "ELB" is supported.

Only Load Balancer "ELB" is supported.

When LoadBalancerNames and this parameter are specified, abnormal instances detected using the health check of Load Balancer are automatically recovered.

- Optional property.
- Type: String

LaunchConfigurationName

- The reference to a LaunchConfiguration resource.
- Required property.
- Type: String

LoadBalancerNames

- List of LoadBalancer resources.
- Optional property.
- Type: List

MaxSize

- Maximum number of instances in the group.

  When an abnormal instance detected using the health check of Load Balancer is automatically recovered, set a value of MinSize + 1 or higher.
- Required property.
- Type: Integer

MinSize

- Minimum number of instances in the group.
- Required property.
- Type: Integer

Tags

- Tags to attach to this group.
- Optional property.
- Type: List
- List contents:
  - Optional property.
  - Type: Map
  - Map properties:
    - Key
      - Required property.
      - Type: String
    - Value
      - Required property.
      - Type: String

  Below is an example of specifying a password when the operating system is Windows.

```
Tags: [ { Value: 'password', Key: 'admin_pass'} ]
```

VPCZoneIdentifier

- To list the internal subnet to which the instance will be attached.

  When specified, set one or more subnets.

When LoadBalancerNames and multiple subnets are specified for this parameter, the first subnet is targeted for distribution by Load Balancer.

- Optional property.
- Type: List
- List contents:
  - UUID of the internal subnet to which the instance will be attached.
  - Optional property.
  - Type: String

5.1.1.1.4 Attributes

InstanceList

A comma-delimited list of server ip addresses. (Heat extension).

5.1.1.2 FCX::AutoScaling::LaunchConfiguration

5.1.1.2.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: FCX::AutoScaling::LaunchConfiguration
    properties:
      BlockDeviceMappingsV2: [{"source_type": String, "destination_type": String,
 "boot_index": String, "device_name": String, "volume_size": String, "uuid": String,
 "delete_on_termination": Boolean, "volume_type": String}, ...]
      ImageId: String
      InstanceType: String
      KeyName: String
      NovaSchedulerHints: [{"Value": String, "Key": String}, {"Value": String, "Key":
 String}, ...]
      SecurityGroups: [Value, Value, ...]
      UserData: String
```

5.1.1.2.2 Instance that will be created

The format of the instance name to be created is as follows:

"First 2 characters of the stack name" + "-" + "Last 11 characters of the resource name of the AutoScalingGroup" + "-" + "Random ID (12 characters)" + "-" + "Random ID (12 characters)" + "-" + "Random ID (12 characters)"

Example: au-aling_group-knu4eeueo2c5-cyrtttd6lwbu-xsge7xcbkxum

The instance name is set as the host name with the hyphen (-) converted into an underscore (_).

5.1.1.2.3 Notes

- To enable access to a created instance from an external network, it is necessary to separately assign an IP address (floating IP address) for external connections to an instance after it is created using a stack.
- After an instance is created, loads will not be distributed to the new IP address even if the IP address of a distribution destination instance registered in Load Balancer is changed.
- When the properties are changed on updating a stack, the properties after the change are reflected to newly added instances or redeployed instances.

5.1.1.2.4 Properties

BlockDeviceMappingsV2

- Block device mappings to attach to instance.
- Required property.
- Type: List
- List contents:
  - Optional property.
  - Type: Map
  - Map properties:
    - source_type
      - Describes the volume source type for the volume.
      - Required property.
      - Allowed values: image, volume, snapshot
      - Type: String
    - destination_type
      - Describes where the volume comes from.
      - Required property.
      - Allowed values: volume
      - Type: String
    - boot_index
      - Indicates a number designating the boot order of the device.

        Specifies continuous values from 0. For the boot disk, "0" is specified.
      - Required property.
      - Type: String
    - volume_size
      - Size of the volume (GB).

        This item must be specified when "image" is specified for source_type. Specify a value equal to or higher than the min_disk parameter of the image to be used. If the min_disk parameter of the image to be used has not been specified or is "0", check the minimum size with the image provider and specify the value accordingly.

        If "volume" was specified for source_type, this item will not be enabled even if a value is specified, and the volume size will not change.

        If "snapshot" was specified for source_type, and this item is omitted, the volume size of the snapshot collection source will be used.
      - Optional property.
      - Type: String
    - uuid
      - uuid of the resource specified for source_type.
      - Required property.
      - Type: String
    - delete_on_termination
      - Indicate whether the volume should be deleted when the instance is terminated.

        When "True" is specified, the volume that was created during scale-out and stack creation will be deleted during scale-in and during stack deletion.

        When "False" is specified, the volume that was created during scale-out and stack creation will not be deleted during scale-in and during stack deletion. If you want to retain the volume content even after an instance is deleted, specify "False".

        The volume where snapshots are collected will not be deleted even if "True" is specified.

- Optional property, defaults to "True".
- Type: Boolean

- device_name
  - A device name where the volume will be attached in the system at /dev/device_name.e.g. vdb

    Specify in the "/dev/vdx" format. "/dev/vd" is a fixed string, specify a letter that is valid as a device name for "x".

    When creating an instance with multiple volumes assigned, specify the highest priority letter among all of the device names of volumes for the boot volume.

    The order of priority is a > b > c > ...
  - Required property.
  - Type: String
- volume_type
  - Specify a volume type name.

    Volumes are only created with the specified volume type when image is specified for sourcetype. If omitted, M1 is used.

    When volume is specified for sourcetype, the volume type cannot be changed. Therefore, specification of values is not possible.

    When snapshot is specified for sourcetype, volumes are created using the volume type of the volume of the snapshot source. Specification of values is not possible.
  - Optional property.
  - Type: String

ImageId

- Glance image ID or name.
- Optional property.
- Type: String
- Value must be of type glance.image

InstanceType

- Nova instance type (flavor).
- Optional property.
- Type: String
- Value must be of type nova.flavor

KeyName

- Optional Nova keypair name.
- Optional property.
- Type: String
- Value must be of type nova.keypair

NovaSchedulerHints

- Scheduler hints to pass to Nova (Heat extension).

  When creating an instance with the server group uuid of "anti-affinity" specified, if there is no VM host where an instance can be created (the number of VM hosts that can be used is smaller than the number of instances belonging to the same server group), the instance status becomes ERROR after the instance creation request is received.
- Optional property.
- Type: List
- List contents:

- Optional property.
- Type: Map
- Map properties:
  - Key
    - Required property.
    - Type: String
  - Value
    - Required property.
    - Type: String

SecurityGroups

- Security group names to assign.

  SecurityGroup must be a security group with the permissions for TCP connections to the following IP address and port number.
  - IP address: 169.254.169.254
  - Port number: 80

  
  CAUTION

  If TCP connections are not permitted, setting of the host name (computer name) and administrator password may not be performed during instance creation.

- Optional property.
- Type: List

UserData

- User data to pass to instance.

  Specifies the script. The supported format is mainly as shown below.
  - Linux:
    - Shell script (begins with #!)
  - Windows:
    - PowerShell (begins with #ps1_sysnative or #ps1_x86)
    - Windows batch (begins with rem cmd)
- Optional property.
- Type: String

Below is an example in which the c:temp directory is created using PowerShell.

```
UserData: |
  #!ps1_sysnative
  New-Item "c:¥¥temp" -Type Directory
```

Below is an example in which unique processing is performed after a restart of the instance.

```
UserData: |
   #!/bin/sh
   * Declare the processing to be performed before a restart

   script_name="<*Any name>"

   CLOUD_DIR="/var/lib/cloud"
   SCRIPT_DIR="${CLOUD_DIR}/scripts"
   MODULE="per-instance"
   SCRIPT="${SCRIPT_DIR}/${MODULE}/${script_name}"

   cat > ${SCRIPT} <<EOS
   #!/bin/sh
   * Declare the processing to be performed after a restart
   EOS
```

```
chmod +x "${SCRIPT}"
rm /var/lib/cloud/instance/sem/config_scripts_per_instance

reboot #* Restart
```


CAUTION

In case operating system is Linux, specified value with cloud-config format invalid.

5.1.1.3 FCX::AutoScaling::ScalingPolicy

5.1.1.3.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: FCX::AutoScaling::ScalingPolicy
    properties:
      AdjustmentType: String
      AutoScalingGroupName: String
      Cooldown: Number
      ScalingAdjustment: Number
```

5.1.1.3.2 Notes

None.

5.1.1.3.3 Properties

AdjustmentType

- Type of adjustment (absolute or percentage).
- Required property.
- Allowed values: ChangeInCapacity, ExactCapacity, PercentChangeInCapacity

AutoScalingGroupName

- AutoScaling group name to apply policy to.
- Required property.
- Type: String

Cooldown

- Cooldown period, in seconds.

  Set sufficient time (in seconds) for execution of scaling to be completed.

  Alarms that occur during the standby time leading up to the next scale are discarded.

  When abnormal instances detected using the health check of Load Balancer are automatically recovered, and scaling by CPU load, etc., is used at the same time, only set Cooldown of FCX::AutoScaling::AutoScalingGroup, and do not specify this item.

  The estimation formula is shown below.

  - When using scale-out:

    (Time required to create one instance x ScalingAdjustment value)

    - FCX::AutoScaling::Time specified for HealthCheckGracePeriod of AutoScalingGroup
    - OS::Ceilometer::Alarm period x OS::Ceilometer::Alarm evaluation_periods
    - Number of Grade(ELB VM) subnets x Number of Grades (ELB VM) x 10 (*1)
    - 60

- When using scale-in:

  Time required to delete one instance x (ScalingAdjustment value x -1)
  - FCX::AutoScaling::Time specified for HealthCheckGracePeriod of AutoScalingGroup
  - OS::Ceilometer::Alarm period x OS::Ceilometer::Alarm evaluation_periods
  - Number of Grade(ELB VM) subnets x Number of Grades (ELB VM) x 10 (*1)
  - 60
- When an abnormal instance detected using the health check of Load Balancer is automatically recovered

  (Time required to create one instance x ScalingAdjustment value)
  - FCX::AutoScaling::Time specified for HealthCheckGracePeriod of AutoScalingGroup
  - (Time required to delete one instance x 5)
  - FCX::AutoScaling::Time specified for HealthCheckGracePeriod of AutoScalingGroup
  - (Time required to create one instance x FCX::AutoScaling::MinSize value of AutoScalingGroup)
  - FCX::AutoScaling::Time specified for HealthCheckGracePeriod of AutoScalingGroup
  - OS::Ceilometer::Alarm period x OS::Ceilometer::Alarm evaluation_periods
  - Number of Grade(ELB VM) subnets x Number of Grades (ELB VM) x 10 (*1) x 3
  - 60


Notice

- "Number of Grade(ELB VM) subnets" is the number of subnets specified in Subnets in the properties of FCX::ExpandableLoadBalancer::LoadBalancer in the template.
- "Number of Grades (ELB VM)" is the number of instances registered under the name specified in LoadBalancerName of FCX::ExpandableLoadBalancer::LoadBalancer in the template when a stack is created.
- *1: This is rough estimate for times of low loads, so this will vary depending on communication performance.

- Optional property, defaults to 0.
- Type: Number

ScalingAdjustment

- Size of adjustment.

  Notes when automatic recovery of abnormal instances detected by the health check of Load Balancer is used:
  - If 0 is specified, nothing will happen.
  - Specify a value smaller than MaxSize of FCX::AutoScaling::AutoScalingGroup, and in the range of 1 to 5.
- Required property.
- Type: Number

5.1.1.3.4 Attributes

AlarmUrl

A signed url to handle the alarm. (Heat extension).

5.1.2 Telemetry

5.1.2.1 OS::Ceilometer::Alarm

5.1.2.1.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: OS::Ceilometer::Alarm
    properties:
      alarm_actions: [Value, Value, ...]
      comparison_operator: String
      description: String
      enabled: Boolean
      evaluation_periods: Integer
      insufficient_data_actions: [Value, Value, ...]
      matching_metadata: {...}
      meter_name: String
      ok_actions: [Value, Value, ...]
      period: Integer
      repeat_actions: Boolean
      statistic: String
      threshold: Number
```

5.1.2.1.2 Notes

None

5.1.2.1.3 Properties

alarm_actions

- A list of URLs (webhooks) to invoke when state transitions to alarm.
- Optional property.
- Type: List

comparison_operator

- Operator used to compare specified statistic with threshold.
- Optional property.
- Allowed values: ge, gt, eq, ne, lt, le
- Type: String

description

- Description for the alarm.
- Optional property.
- Type: String

enabled

- True if alarm evaluation/actioning is enabled.
- Optional property, defaults to "true".
- Type: Boolean

evaluation_periods

- Number of periods to evaluate over.
- Optional property, defaults to 1.
- Type: Integer

insufficient_data_actions

- A list of URLs (webhooks) to invoke when state transitions to insufficient-data.

- Optional property.
- Type: List

matching_metadata

- Meter should match this resource metadata (key=value) additionally to the meter_name.

  When monitoring the CPU usage rate of instances in the AutoScalingGroup, metadata.user_metadata.groupname is specified as the key, and the AutoScalingGroup resource is specified for the value.

  When this item is used for automatic recovery of an abnormal instance detected using the health check of Load Balancer, resource_id is specified as the key, and the name of Load Balancer is specified for the value.
- Optional property, defaults to "{}".
- Type: Map

meter_name

- Meter name watched by the alarm.

  When monitoring the CPU usage rate of an instance, fcx.compute.cpu_util is specified.

  When this item is used for automatic recovery of an abnormal instance detected using the health check of Load Balancer, fcx.loadbalancing.instance.unhealthy is specified.
- Required property.
- Type: String

ok_actions

- A list of URLs (webhooks) to invoke when state transitions to ok.
- Optional property
- Type: List

period

- Period (seconds) to evaluate over.
- Optional property, defaults to 60.
- Type: Integer

repeat_actions

- False to trigger actions when the threshold is reached AND the alarm's state has changed. By default, actions are called each time the threshold is reached.

  Alarms that occur during the cooldown period leading up to the next scale are discarded, so it may not be possible to recover from events that have occurred.

  Specify "true" to periodically issue an alarm until recovery of the event(s) in question.

  When this item is used for automatic recovery of an abnormal instance detected using the health check of Load Balancer, "true" is specified.
- Optional property, defaults to "false".
- Type: Boolean

statistic

- Meter statistic to evaluate.

  When this item is used for automatic recovery of an abnormal instance detected using the health check of Load Balancer, "min" is specified.
- Optional property.
- Allowed values: count, avg, sum, min, max
- Type: String

threshold

- Threshold to evaluate against.

  Notes when automatic recovery of abnormal instances detected by the health check of Load Balancer is used:

  - Specify the same value as the ScalingAdjustment value specified in the policy settings.
  - When a value of 2 or higher has been set, automatic recovery will not take place until at least that number of instances are in an error state.
- Required property.
- Type: Number

5.1.3 Block Storage

5.1.3.1 OS::Cinder::Volume

5.1.3.1.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: OS::Cinder::Volume
    properties:
      availability_zone: String
      backup_id: String
      description: String
      image: String
      metadata: {...}
      name: String
      size: Integer
      snapshot_id: String
      source_volid: String
      volume_type: String
```

5.1.3.1.2 Notes

When snapshot is specified for sourcetype, volumes are created using the volume type of the volume of the snapshot source. Specification of values is not possible.

5.1.3.1.3 Properties

availability_zone

- The availability zone in which the volume will be created.
- Optional property.
- Type: String

backup_id

- If specified, the backup to create the volume from.
- Optional property.
- Type: String

description

- A description of the volume.

  This value will not be set when backup_id is specified.
- Optional property.

- Type: String

image

- If specified, the name or ID of the image to create the volume from.
- Optional property.
- Value must be of type glance.image

metadata

- Key/value pairs to associate with the volume.

  This value will not be set when backup_id is specified.
- Optional property.
- Type: Map

name

- A name used to distinguish the volume.

  When backup_id is specified, the value that is set will differ depending on the setting value of the volume from which the backup was collected.
  - The volume from which the backup was collected does not have a setting value

    This setting value will be set.
  - The volume from which the backup was collected has a setting value

    The value of the volume from which the backup was collected will be set.
- Optional property
- Type: String

size

- The size of the volume in GB. On update only increase in size is supported.
- Optional property.
- Type: Integer
- The value must be at least 1.

snapshot_id

- If specified, the snapshot to create the volume from.
- Optional property.
- Type: String
- Value must be of type cinder.snapshot

source_volid

- If specified, the volume to use as source.
- Optional property.
- Type: String
- Value must be of type cinder.volume

imageRef

CAUTION

DEPRECATED! - Use property image.

- The ID of the image to create the volume from.
- Optional property.
- Type: String

volume_type

- If specified, the type of volume to use, mapping to a specific backend.
  This value will not be set when backup_id is specified.
- Optional property.
- Type: String

5.1.3.1.4 Attributes

availability_zone

The availability zone in which the volume is located.

bootable

Boolean indicating if the volume can be booted or not.

created_at

The timestamp indicating volume creation.

display_description

Description of the volume.

display_name

Name of the volume.

metadata

Key/value pairs associated with the volume.

size

The size of the volume in GB.

snapshot_id

The snapshot the volume was created from, if any.

source_volid

The volume used as source, if any.

status

The current status of the volume.

volume_type

The type of the volume mapping to a backend, if any.

5.1.3.2 OS::Cinder::VolumeAttachment

5.1.3.2.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: OS::Cinder::VolumeAttachment
    properties:
      instance_uuid: String
      mountpoint: String
```

```
      volume_id: String
```

5.1.3.2.2 Notes

None

5.1.3.2.3 Properties

instance_uuid

- The ID of the server to which the volume attaches.
- Required property.
- Type: String

mountpoint

- The location where the volume is exposed on the instance. This assignment may not be honored and it is advised that the path /dev/disk/by-id/virtio-<VolumeId> be used instead.
- Optional property.
- Type: String

volume_id

- The ID of the volume to be attached.
- Required property.
- Type: String
- Value must be of type cinder.volume

5.1.4 Compute

5.1.4.1 OS::Nova::Server

5.1.4.1.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: OS::Nova::Server
    properties:
      availability_zone: String
      block_device_mapping: [{"volume_size": Integer, "volume_id": String, "snapshot_id":
 String, "delete_on_termination": Boolean, "device_name": String}, {"volume_size":
 Integer, "volume_id": String, "snapshot_id": String, "delete_on_termination": Boolean,
 "device_name": String}, ...]
      diskConfig: String
      flavor: String
      flavor_update_policy: String
      image: String
      image_update_policy: String
      key_name: String
      metadata: {...}
      name: String
      networks: [{"port": String, "fixed_ip": String, "uuid": String, "network": String},
 {"port": String, "fixed_ip": String, "uuid": String, "network": String}, ...]
      personality: {...}
      reservation_id: String
      scheduler_hints: {...}
      security_groups: [Value, Value, ...]
      software_config_transport: String
```

```
      user_data: String
      user_data_format: String
```

5.1.4.1.2 Notes

To assign the following information to a virtual server, it is necessary to have a virtual router connected to the network to which the virtual server is connected.

- Host name (Computer name)
- Administrator password
- Authentication key (keypair)

5.1.4.1.3 Properties

availability_zone

- Name of the availability zone for server placement.
- Optional property.
- Type: String

block_device_mapping

- Block device mappings for this server.
- Required property.
- Type: List
- List contents:
  - Optional property.
  - Type: Map
  - Map properties:
    - delete_on_termination
      - Indicate whether the volume should be deleted when the server is terminated.

        Specifies whether the volume that was created during scale-out and stack creation will be deleted during scale-in and during stack deletion.

        When "True" is specified, the volume that was created during scale-out and stack creation will be deleted during scale-in and during stack deletion.

        When "False" is specified, the volume that was created during scale-out and stack creation will not be deleted during scale-in and during stack deletion.

        If not specified, "False (do not delete)" will be used.

        The volume where snapshots are collected will not be deleted even if "True" is specified.
      - Optional property.
      - Type: Boolean
    - device_name
      - A device name where the volume will be attached in the system at /dev/device_name. This value is typically vda.

        Specify in the "/dev/vdx" format. "/dev/vd" is a fixed string, specify a letter that is valid as a device name for "x".

        When creating an instance with multiple volumes assigned, specify the highest priority letter among all of the device names of volumes for the boot volume.

        The order of priority is a > b > c > ...

        If vda is specified without an image being specified, only vda will be specified and "/dev/" will not be appended.
      - Required property.
      - Type: String
    - snapshot_id

        - The ID of the snapshot to create a volume from.

          If volume_id is not specified, this item must be specified.
        - Optional property.
        - Type: String
        - Value must be of type cinder.snapshot
      - volume_id
        - The ID of the volume to boot from. Only one of volume_id or snapshot_id should be provided.

          If snapshot_id is not specified, this item must be specified.
        - Optional property.
        - Type: String
        - Value must be of type cinder.volume
      - volume_size
        - The size of the volume, in GB. It is safe to leave this blank and have the Compute service infer the size.
          - If volume_id is specified and delete on termination is set to "True":

            This setting is mandatory. However, the specified volume size is ignored, and there is no change to the volume size specified for volume_id.
          - If volume_id is specified and delete on termination is not specified, or set to "False":

            This must not be specified. If specified, the specified volume size is ignored, and there is no change to the volume size specified for volume_id.
          - If snapshot_id is specified and delete on termination is set to "True":

            This setting is mandatory.
          - If snapshot_id is specified and delete on termination is not specified, or set to "False":

            This item is optional. If this item is omitted, the volume size of the snapshot collection source will be used.
        - Optional property.
        - Type: Integer

diskConfig

- Control how the disk is partitioned when the server is created.
- Optional property.
- Allowed values: AUTO, MANUAL

flavor

- The ID or name of the flavor to boot onto.
- Required property.
- Type: String
- Value must be of type nova.flavor

image

- The ID or name of the image to boot with.
- Optional property.
- Type: String
- Value must be of type glance.image

key_name

- Name of keypair to inject into the server.
- Optional property.

- Type: String
- Value must be of type nova.keypair

metadata

- Arbitrary key/value metadata to store for this server. Both keys and values must be 255 characters or less. Non-string values will be serialized to JSON (and the serialized string must be 255 characters or less).
- Optional property.
- Type: Map

Below is an example of specifying a password when the operating system is Windows.

The specified password is set for users specified in cloudbase-init.

```
metadata: { "admin_pass": 'password' } }
```

name

- Server name.

  Specify the value using up to 63 characters. Up to 255 characters can be specified, however, if the name is 64 characters or longer, the host name/computer name set for an instance will be as follows.

  - Linux:

    The host name will be "host-fixedIpAddressOfEth0".

  - Windows:

    The computer name is the default name set by Windows.

  The string set for the host name/computer name is changed as follows and set.

  - Spaces ( ) and underscores (_) are replaced with a hyphen (-).
  - Uppercase alphabetic characters are replaced with lowercase alphabetic characters.
  - Symbols other than periods (.) and hyphens (-) are removed.
  - Periods (.) are removed from the beginning and end of the string if any, and a string consisting of hyphens (-) is removed.

  Furthermore, if the operating system is Windows:

  - If the string contains a period (.) other than at the beginning or end, the characters preceding the period will be used for the name.
- Optional property.
- Type: String

networks

- An ordered list of NICs to be added to this server, with information about connected networks, fixed IPs, port etc.
- Optional property.
- Type: List
- List contents:
  - Optional property.
  - Type: Map
  - Map properties:
    - fixed_ip
      - Fixed IP address to specify for the port created on the requested network.
      - Optional property.
      - Type: String
    - network
      - Name or ID of network to create a port on.

- Optional property.
- Type: String
- Value must be of type neutron.network

- port
  - ID of an existing port to associate with this server.
  - Optional property.
  - Type: String
- uuid


CAUTION

DEPRECATED! - Use property network.

  - ID of network to create a port on.
  - Optional property.
  - Type: String
  - Value must be of type neutron.network

scheduler_hints

- Arbitrary key-value pairs specified by the client to help boot a server.

  When creating an instance with the server group uuid of "anti-affinity" specified, if there is no VM host where an instance can be created (the number of VM hosts that can be used is smaller than the number of instances belonging to the same server group), the instance status becomes ERROR after the instance creation request is received.
- Optional property.
- Type: Map

security_groups

- List of security group names or IDs. Cannot be used if neutron ports are associated with this server; assign security groups to the ports instead.

  SecurityGroup must be a security group with the permissions for TCP connections to the following IP address and port number.
  - IP address: 169.254.169.254
  - Port number: 80

  If TCP connections are not permitted, setting of the host name (computer name) and administrator password may not be performed during instance creation.
- Optional property, defaults to "[]".
- Type: List

user_data

- User data script to be executed by cloud-init.

  Specifies the script. The supported format is mainly as shown below.
  - Linux:
    - Shell script (begins with #!)
  - Windows:
    - PowerShell (begins with #ps1_sysnative or #ps1_x86)
    - Windows batch (begins with rem cmd)
- Optional property, defaults to "".
- Type: String

CAUTION If the operating system is Linux, the cloud-config format can be specified apart from scripts, however, it is recommended that scripts be specified.

user_data_format

- Specify "RAW".
- Optional property, defaults to "HEAT_CFNTOOLS".
- Type: String

CAUTION Only "RAW" is supported.
- RAW: The specified user_data is passed as is to Nova.

admin_user

CAUTION DEPRECATED!

- Name of the administrative user to use on the server. This property will be removed from Juno in favor of the default cloud-init user set up for each image (e.g. "ubuntu" for Ubuntu 12.04+, "fedora" for Fedora 19+ and "cloud-user" for CentOS/RHEL 6.5).
- Optional property.
- Type: String

5.1.4.1.4 Attributes

accessIPv4

The manually assigned alternative public IPv4 address of the server.

accessIPv6

The manually assigned alternative public IPv6 address of the server.

addresses

A dict of all network addresses with corresponding port_id. The port ID may be obtained through the following expression: "{get_attr: [<server>, addresses, <network name>, 0, port]}".

```
{get_attr: [<server>, addresses, <network name>, 0, port]}
```

first_address

CAUTION DEPRECATED! - Use the networks attribute instead of first_address. For example: "{get_attr: [<server name>, networks, <network name>, 0]}"

Convenience attribute to fetch the first assigned network address, or an empty string if nothing has been assigned at this time. Result may not be predictable if the server has addresses from more than one network.

instance_name

AWS compatible instance name.

networks

A dict of assigned network addresses of the form: {"public": [ip1, ip2...], "private": [ip3, ip4]}.

show

A dict of all server details as returned by the API.

5.1.4.2 OS::Nova::ServerGroup

5.1.4.2.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: OS::Nova::ServerGroup
    properties:
      name: String
      policies: [String, String, ...]
      availability_zone: String
```

5.1.4.2.2 Notes

When creating an instance with the server group uuid of "anti-affinity" specified, if there is no VM host where an instance can be created (the number of VM hosts that can be used is smaller than the number of instances belonging to the same server group), the instance status becomes ERROR after the instance creation request is received.

5.1.4.2.3 Properties

name

- Server Group name.
- Optional property.
- Type: String

policies

- A list of string policies to apply. Defaults to anti-affinity.
- Optional property, defaults to "['anti-affinity']".
- Allowed values: anti-affinity, affinity
- List contents:
  - *
    - Type: String

availability_zone

- Name of the availability zone for server group placement.
- Optional property.
- Type: String

5.1.4.3 OS::Nova::KeyPair

5.1.4.3.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: OS::Nova::KeyPair
    properties:
      name: String
      public_key: String
```

```
      save_private_key: Boolean
      availability_zones: [String, ...]
```

5.1.4.3.2 Notes

None

5.1.4.3.3 Properties

name

- The name of the key pair.

  Updates cause replacement.

  The length must be in the range 1 to 255.
- Required property.
- Type: String

public_key

- The optional public key. This allows users to supply the public key from a pre-existing key pair.

  If not supplied, a new key pair will be generated.

  Updates cause replacement.
- Optional property.
- Type: String

save_private_key

- True if the system should remember a generated private key; False otherwise.

  Updates cause replacement.
- Optional property.
- Type: Boolean
- Defaults to "False"

availability_zones

- Specifies the availability zone for keypair creation.

  Creates keypairs with the same public key in all availability zones if "all" is specified for the first item.

  Defaults to AZ as in the conventional manner if nothing is specified.

  Specification of multiple availability zones is unsupported.

  If multiple availability zones are specified, an error occurs.

  If the list contains no element (no availability zone specification) or if a null character is specified, an error also occurs.

  If you want to change the parameter, make changes by adding new keypair definitions while leaving existing keypairs undeleted.
- Optional property.
- List contents:
  - *
    - Type: String

5.1.5 Network

5.1.5.1 OS::Neutron::Firewall

5.1.5.1.1 HOT Syntax

Notice

For the Japan East1 / Japan West1 / Japan West2 regions

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: OS::Neutron::Firewall
    properties:
      admin_state_up: Boolean
      description: String
      firewall_policy_id: String
      name: String
      router_id: String
      availability_zone: String
```

Notice

For the Japan East2 regions

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: OS::Neutron::Firewall
    properties:
      admin_state_up: Boolean
      description: String
      firewall_policy_id: String
      name: String
      value_specs: {"router_ids": [String, String, ...]}
      router_id: String
      availability_zone: String
```

5.1.5.1.2 Notes

Notice

- For the Japan East1 / Japan West1 / Japan West2 regions

  None.
- For the Japan East2 regions

  When the router_id parameter and the router_ids parameter of value_spec are both specified, the value of router_ids will be used.

5.1.5.1.3 Properties

admin_state_up

- Administrative state of the firewall. If false (down), firewall does not forward packets and will drop all traffic to/from VMs behind the firewall.

  Can be updated without replacement.
- Optional property, defaults to "True".
- Type: Boolean

description

- Description for the firewall.

  Can be updated without replacement.
- Optional property.
- Type: String

firewall_policy_id

- The ID of the firewall policy that this firewall is associated with.
  Can be updated without replacement.
- Required property.
- Type: String

name

- Name for the firewall.
  Can be updated without replacement.
- Optional property.
- Type: String

value_specs


Notice

- This parameter can only be used in For the Japan East2 regions.

- Extra parameters to include in the request. Parameters are often specific to installed hardware or extensions.
  Can be updated without replacement.
- Optional properties, defaults to "{}".
- Type: Map
- Map contents:
  - * : Map
  - Optional property.
  - Map properties:
    - router_ids
      - The list of IDs for the routers that this firewall be applied.
        If you don't specify, the resource will be created at all routers in AZ.
        If you specify both router_ids and router_id at the same time, the resource will be created at all routers of router_ids.
        Can be updated without replacement.
      - Optional properties.
      - Type: String

router_id

- The ID of the router that this firewall be applied.
  If you don't specify, the resource will be created at all routers in AZ.


Notice

- For the Japan East1 / Japan West1 / Japan West2 regions
  Updates cause replacement.
- For the Japan East2 regions
  If you specify both router_ids and router_id at the same time, the resource will be created at all routers of router_ids.
  Can be updated without replacement.

- Optional property.
- Type: String

availability_zone

- The Availability Zone name.
  If you don't specify, the resource will be created in default AZ.
  Updates cause replacement.
- Optional property.
- Type: String

5.1.5.1.4 Attributes

admin_state_up

The administrative state of the firewall.

description

Description of the firewall.

firewall_policy_id

Unique identifier of the firewall policy used to create the firewall.

name

Name for the firewall.

show

All attributes.

status

The status of the firewall.

tenant_id

Id of the tenant owning the firewall.

router_id

The ID of the router that this firewall applied.

availability_zone

The Availability Zone name.

5.1.5.2 OS::Neutron::FirewallPolicy

5.1.5.2.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: OS::Neutron::FirewallPolicy
    properties:
      audited: Boolean
      description: String
      firewall_rules: [Value, Value, ...]
      name: String
      availability_zone: String
```

5.1.5.2.2 Notes

None.

5.1.5.2.3 Properties

audited

- Whether this policy should be audited. When set to True, each time the firewall policy or the associated firewall rules are changed, this attribute will be set to False and will have to be explicitly set to True through an update operation.

  Can be updated without replacement.
- Optional property, defaults to "False".
- Type: Boolean

description

- Description for the firewall policy.

  Can be updated without replacement.
- Optional property.
- Type: String

firewall_rules

- An ordered list of firewall rules to apply to the firewall.

  Can be updated without replacement.
- Required property.
- Type: List

name

- Name for the firewall policy.

  Can be updated without replacement.
- Optional property.
- Type: String

availability_zone

- The Availability Zone name.

  If you don't specify, the resource will be created in default AZ.

  Updates cause replacement.
- Optional property.
- Type: String

5.1.5.2.4 Attributes

audited

Audit status of this firewall policy.

description

Description of the firewall policy.

firewall_rules

List of firewall rules in this firewall policy.

name

Name for the firewall policy.

shared

Shared status of this firewall policy.

tenant_id

Id of the tenant owning the firewall policy.

availability_zone

The Availability Zone name.

5.1.5.3 OS::Neutron::FirewallRule

5.1.5.3.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: OS::Neutron::FirewallRule
    properties:
      action: String
      description: String
      destination_ip_address: String
      destination_port: String
      enabled: Boolean
      ip_version: String
      name: String
      protocol: String
      source_ip_address: String
      source_port: String
      availability_zone: String
```

5.1.5.3.2 Notes

- If it is possible that multiple firewall rules associated with a single firewall policy may be updated simultaneously, use "depends_on" to configure dependencies between firewall rule resources.

  If there are no dependencies, update of the stack may fail.

  Example:

```
  firewall_rule_1:
    type: OS::Neutron::FirewallRule
    properties:
      name:
   . . .

  firewall_rule_2:
* depends_on: firewall_rule_1
    type: OS::Neutron::FirewallRule
    properties:
      name:
   . . .

  firewall_rule_3:
* depends_on: firewall_rule_2
    type: OS::Neutron::FirewallRule
    properties:
      name:
   . . .
```

5.1.5.3.3 Properties

action

- Action to be performed on the traffic matching the rule.
  Can be updated without replacement.
- Optional property, defaults to "deny".
- Type: String
- Allowed values: allow, deny

description

- Description for the firewall rule.
  Can be updated without replacement.
- Optional property.
- Type: String

destination_ip_address

- Destination IP address or CIDR.
  Can be updated without replacement.
- Optional property.
- Type: String

destination_port

- Destination port number or a range.
  Can be updated without replacement.
- Optional property.
- Type: String

enabled

- Whether this rule should be enabled.
  Can be updated without replacement.
- Optional property, defaults to "True".
- Type: Boolean

ip_version

- Internet protocol version.
  Can be updated without replacement.
- Optional property, defaults to "4".
- Type: String
- Allowed values: 4

name

- Name for the firewall rule.
  Can be updated without replacement.
- Optional property.
- Type: String

protocol

- Protocol for the firewall rule.
  Can be updated without replacement.

- Optional property.
- Type: String
- Allowed values: tcp, udp, icmp, None

source_ip_address

- Source IP address or CIDR.
  Can be updated without replacement.
- Optional property.
- Type: String

source_port

- Source port number or a range.
  Can be updated without replacement.
- Optional property.
- Type: String

availability_zone

- The Availability Zone name.
  If you don't specify, the resource will be created in default AZ.
  Updates cause replacement.
- Optional property.
- Type: String

5.1.5.3.4 Attributes

action

Allow or deny action for this firewall rule.

description

Description of the firewall rule.

destination_ip_address

Destination ip_address for this firewall rule.

destination_port

Destination port range for this firewall rule.

enabled

Indicates whether this firewall rule is enabled or not.

firewall_policy_id

Unique identifier of the firewall policy to which this firewall rule belongs.

ip_version

Ip_version for this firewall rule.

name

Name for the firewall rule.

position

Position of the rule within the firewall policy.

protocol

Protocol value for this firewall rule.

shared

Shared status of this firewall rule.

source_ip_address

Source ip_address for this firewall rule.

source_port

Source port range for this firewall rule.

tenant_id

Id of the tenant owning the firewall.

availability_zone

The Availability Zone name.

5.1.5.4 OS::Neutron::FloatingIP

5.1.5.4.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: OS::Neutron::FloatingIP
    properties:
      fixed_ip_address: String
      port_id: String
      floating_network_id: String
      availability_zone: String
```

5.1.5.4.2 Notes

None.

5.1.5.4.3 Properties

fixed_ip_address

- IP address to use if the port has multiple addresses.

- For the Japan East1 / Japan West1 / Japan West2 regions
  Updates cause replacement.
- For the Japan East2 regions
  Can be updated without replacement.

- Optional property.
- Type: String

port_id

- ID of an existing port with at least one IP address to associate with this floating IP.


Notice

- For the Japan East1 / Japan West1 / Japan West2 regions
  Updates cause replacement.
- For the Japan East2 regions
  Can be updated without replacement.

- Optional property.
- Type: String

floating_network_id

- ID of network to allocate floating IP from.
  Updates cause replacement.
- Required property.
- Type: String

availability_zone

- The Availability Zone name.
  If you don't specify, the resource will be created in default AZ.
  Updates cause replacement.
- Optional property.
- Type: String

5.1.5.4.4 Attributes

fixed_ip_address

IP address of the associated port, if specified.

floating_ip_address

The allocated address of this IP.

floating_network_id

ID of the network in which this IP is allocated.

port_id

ID of the port associated with this IP.

router_id

ID of the router used as gateway, set when associated with a port.

show

All attributes.

tenant_id

The tenant owning this floating IP.

availability_zone

The Availability Zone name.

5.1.5.5 OS::Neutron::FloatingIPAssociation

5.1.5.5.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: OS::Neutron::FloatingIPAssociation
    properties:
      fixed_ip_address: String
      floatingip_id: String
      port_id: String
```

5.1.5.5.2 Notes

None.

5.1.5.5.3 Properties

floatingip_id

- ID of the floating IP to associate.
  Can be updated without replacement.
- Required property.
- Type: String

port_id

- ID of an existing port with at least one IP address to associate with this floating IP.
  Can be updated without replacement.
- Required property.
- Type: String

fixed_ip_address

- IP address to use if the port has multiple addresses.
  Can be updated without replacement.
- Optional property.
- Type: String

5.1.5.5.4 Attributes

None

5.1.5.6 OS::Neutron::Net

5.1.5.6.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: OS::Neutron::Net
    properties:
      admin_state_up: Boolean
      name: String
      availability_zone: String
```

5.1.5.6.2 Notes

None.

5.1.5.6.3 Properties

admin_state_up

- A boolean value specifying the administrative status of the network.
  Can be updated without replacement.
- Optional property, defaults to "True".
- Type: Boolean

name

- A string specifying a symbolic name for the network, which is not required to be unique.
  Can be updated without replacement.
- Optional property.
- Type: String

tenant_id

- The ID of the tenant which will own the network. Only administrative users can set the tenant identifier; this cannot be changed using authorization policies.
  Updates cause replacement.
- Optional property.
- Type: String

availability_zone

- The Availability Zone name.
  If you don't specify, the resource will be created in default AZ.
  Updates cause replacement.
- Optional property.
- Type: String

5.1.5.6.4 Attributes

admin_state_up

The administrative status of the network.

name

The name of the network

show

All attributes.

status

The status of the network.

subnets

Subnets of this network.

tenant_id

The tenant owning this network.

availability_zone

The Availability Zone name.

5.1.5.7 OS::Neutron::Port

5.1.5.7.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: OS::Neutron::Port
    properties:
      admin_state_up: Boolean
      allowed_address_pairs: [{"ip_address": String, "mac_address": String}, {"ip_address":
 String, "mac_address": String}, ...]
      fixed_ips: [{"ip_address": String, "subnet_id": String}, {"ip_address": String,
 "subnet_id": String}, ...]
      mac_address: String
      name: String
      security_groups: [Value, Value, ...]
      network_id: String
      availability_zone: String
```

5.1.5.7.2 Notes

None.

5.1.5.7.3 Properties

admin_state_up

- The administrative state of this port.
  Can be updated without replacement.
- Optional property, defaults to "True".
- Type: Boolean

allowed_address_pairs

- Additional MAC/IP address pairs allowed to pass through the port.
  Updates cause replacement.
- Optional property.
- Type: List
- List contents:
  - * : Map
  - Updates cause replacement.
  - Optional property.
  - Map properties:
    - ip_address
      - IP address to allow through this port.
        Updates cause replacement.
      - Required property.
      - Type: String
    - mac_address
      - MAC address to allow through this port.
        Updates cause replacement.

- Optional property.
- Type: String

fixed_ips

- Desired IPs for this port.
  Can be updated without replacement.
- Optional property, defaults to "[]".
- Type: List
- List contents:
  - * : Map
  - Updates cause replacement.
  - Optional property.
  - Map properties:
    - ip_address
      - IP address desired in the subnet for this port.
        Updates cause replacement.
      - Optional property.
      - Type: String
    - subnet_id
      - Updates cause replacement.
      - Optional property.
      - Type: String

mac_address

- MAC address to give to this port.
  Updates cause replacement.
- Optional property.
- Type: String

name

- A symbolic name for this port.
  Can be updated without replacement.
- Optional property.
- Type: String

security_groups

- Security group IDs to associate with this port.
  Can be updated without replacement.
- Optional property, defaults to "[]".
- Type: List

network_id

- Updates cause replacement.
- Required property.
- Type: String

availability_zone

- The Availability Zone name.

If you don't specify, the resource will be created in default AZ.
Updates cause replacement.

- Optional property.
- Type: String

5.1.5.7.4 Attributes

admin_state_up

The administrative state of this port.

allowed_address_pairs

Additional MAC/IP address pairs allowed to pass through a port.

device_id

Unique identifier for the device.

device_owner

Name of the network owning the port.

fixed_ips

Fixed IP addresses.

mac_address

MAC address of the port.

name

Friendly name of the port.

network_id

Unique identifier for the network owning the port.

security_groups

A list of security groups for the port.

show

All attributes.

status

The status of the port.

tenant_id

Tenant owning the port.

availability_zone

The Availability Zone name.

5.1.5.8 OS::Neutron::Router

5.1.5.8.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
```

```
resources:
  ...
  the_resource:
    type: OS::Neutron::Router
    properties:
      admin_state_up: Boolean
      external_gateway_info: {"network": String}
      name: String
      availability_zone: String
```

5.1.5.8.2 Notes

None.

5.1.5.8.3 Properties

admin_state_up

- The administrative state of the router.
  Can be updated without replacement.
- Optional property, defaults to "True".
- Type: Boolean

external_gateway_info

- External network gateway configuration for a router.
  Can be updated without replacement.
- Optional property.
- Type: Map(Update Only).
- Map properties:
  - network
    - ID or name of the external network for the gateway.
      Can be updated without replacement.
    - Required property.
    - Type: String

name

- The name of the router.
  Can be updated without replacement.
- Optional property.
- Type: String

availability_zone

- The Availability Zone name.
  If you don't specify, the resource will be created in default AZ.
  Updates cause replacement.
- Optional property.
- Type: String

5.1.5.8.4 Attributes

admin_state_up

Administrative state of the router.

external_gateway_info

Gateway network for the router.

name

Friendly name of the router.

show

All attributes.

status

The status of the router.

tenant_id

Tenant owning the router.

availability_zone

The Availability Zone name.

5.1.5.9 OS::Neutron::RouterInterface

5.1.5.9.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: OS::Neutron::RouterInterface
    properties:
      port_id: String
      router_id: String
      subnet_id: String
```

5.1.5.9.2 Notes

When setting multiple RouterInterfaces for the same router, set depends_on: between resources of OS::Neutron::RouterInterface so that setting of RouterInterface is not done at the same time. (The italicized part of the example below is the setting of the dependency of router_interface.)

```
service_router1:
    type: OS::Neutron::Router
    properties:
      availability_zone: { get_param: az }
      name: { get_param: service_router1_name }
  service_router_interface1:
    depends_on: service_router1
    type: OS::Neutron::RouterInterface
    properties:
      router_id: { get_resource: service_router1 }
      port_id: { get_resource: gw_port1 }
  service_router_interface2:
    depends_on: [ service_router1, service_router_interface1 ]
    type: OS::Neutron::RouterInterface
    properties:
      router_id: { get_resource: service_router1 }
      port_id: { get_resource: gw_port2 }
```

5.1.5.9.3 Properties

port_id

- The port id, either subnet_id or port_id should be specified.
  Updates cause replacement.
- Optional property.
- Type: String

router_id

- The router id.
  Updates cause replacement.
- Required property.
- Type: String

subnet_id

- The subnet id, either subnet_id or port_id should be specified.
  Updates cause replacement.
- Optional property.
- Type: String

5.1.5.9.4 Attributes

None.

5.1.5.10 OS::Neutron::SecurityGroup

5.1.5.10.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: OS::Neutron::SecurityGroup
    properties:
      description: String
      name: String
      rules: [{"protocol": String, "remote_group_id": String, "port_range_max": Integer,
 "remote_ip_prefix": String, "port_range_min": Integer, "ethertype": String, "direction":
String, "remote_mode": String}, {"protocol": String, "remote_group_id": String,
 "port_range_max": Integer, "remote_ip_prefix": String, "port_range_min": Integer,
 "ethertype": String, "direction": String, "remote_mode": String}, ...]
```

5.1.5.10.2 Notes

None.

5.1.5.10.3 Properties

description

- Description of the security group.
  Can be updated without replacement.
- Optional property.
- Type: String

name

- A string specifying a symbolic name for the security group, which is not required to be unique.
  Can be updated without replacement.
- Optional property.
- Type: String

availability_zone

- The Availability Zone name.
  This parameter has been deprecated. If it is specified, it will be ignored.
  Updates cause replacement.
- Optional property.
- Type: String

rules

- List of security group rules.
  Can be updated without replacement.
- Optional property, defaults to "[]".
- Type: List
- List contents:
  - * : Map
  - Updates cause replacement.
  - Optional property.
  - Map properties:
    - direction
      - The direction in which the security group rule is applied. For a compute instance, an ingress security group rule matches traffic that is incoming (ingress) for that instance. An egress rule is applied to traffic leaving the instance.
        Updates cause replacement.
      - Optional property, defaults to "ingress".
      - Type: String
      - Allowed values: ingress, egress
    - ethertype
      - Ethertype of the traffic.
        Updates cause replacement.
      - Optional property, defaults to "IPv4".
      - Allowed values: IPv4, IPv6
      - Type: String
    - port_range_max
      - The maximum port number in the range that is matched by the security group rule. The port_range_min attribute constrains the port_range_max attribute. If the protocol is ICMP, this value must be an ICMP type.
        Updates cause replacement.
      - Optional property.
      - Type: Integer
    - port_range_min
      - The minimum port number in the range that is matched by the security group rule. If the protocol is TCP or UDP, this value must be less than or equal to the value of the port_range_max attribute. If the protocol is ICMP, this value must be an ICMP type.
        Updates cause replacement.

- Optional property.
- Type: Integer

- protocol
  - The protocol that is matched by the security group rule. Valid values include tcp, udp, and icmp.

    Updates cause replacement.
  - Optional property.
  - Type: String
- remote_group_id
  - The remote group ID to be associated with this security group rule. If no value is specified then this rule will use this security group for the remote_group_id.

    Updates cause replacement.
  - Optional property.
  - Type: String
- remote_ip_prefix
  - The remote IP prefix (CIDR) to be associated with this security group rule.

    Updates cause replacement.
  - Optional property.
  - Type: String
- remote_mode
  - Whether to specify a remote group or a remote IP prefix.

    Updates cause replacement.
  - Optional property, defaults to "remote_ip_prefix".
  - Type: String
  - Allowed values: remote_ip_prefix, remote_group_id
- availability_zone
  - The Availability Zone name.

    This parameter has been deprecated. If it is specified, it will be ignored.

    Updates cause replacement.
  - Optional property.
  - Type: String

5.1.5.10.4 Attributes

None.

5.1.5.11 OS::Neutron::Subnet

5.1.5.11.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: OS::Neutron::Subnet
    properties:
      allocation_pools: [{"end": String, "start": String}, {"end": String, "start":
 String}, ...]
      cidr: String
      dns_nameservers: [Value, Value, ...]
      enable_dhcp: Boolean
      gateway_ip: String
```

```
    host_routes: [{"nexthop": String, "destination": String}, {"nexthop": String,
"destination": String}, ...]
    ip_version: Integer
    name: String
    network_id: String
    availability_zone: String
```

5.1.5.11.2 Notes

- When updating the values of parameters with "Updates cause replacement." indicated in the explanation of each parameter of subnet in 5.1.5.11.3, perform the following.

  Update the stack once, using a template from which subnet has been deleted. Then, update the stack a second time using a template which includes the new values for the parameters of subnet.

5.1.5.11.3 Properties

allocation_pools

- The start and end addresses for the allocation pools.

  Updates cause replacement.
- Optional property.
- List contents:
  - * : Map
  - Updates cause replacement.
  - Optional property.
  - Map properties:
    - end
      - Updates cause replacement.
      - Required property.
      - Type: String
    - start
      - Updates cause replacement.
      - Required property.
      - Type: String

cidr

- The CIDR.

  Don't specified ISP shared address(100.64.0.0/10) and divided to it.

  Updates cause replacement.
- Required property.
- Type: String

dns_nameservers

- A specified set of DNS name servers to be used.

  Can be updated without replacement.
- Optional property, defaults to "[]".
- Type: List

enable_dhcp

- Set to true if DHCP is enabled and false if DHCP is disabled.

  Can be updated without replacement.
- Optional property, defaults to "True".

- Type: Boolean

gateway_ip

- The gateway IP address.
  Can be updated without replacement.
- Optional property.
- Type: String

host_routes

The behavior varies depending on the region.

Notice

For the Japan East1 / Japan West1 / Japan West2 regions

- Updates cause replacement.
- Optional property.
- List contents:
  - Type: List
  - * : Map
  - Updates cause replacement.
  - Optional property.
  - Map properties:
    - destination
      - Updates cause replacement.
      - Required property.
      - Type: String
    - nexthop
      - Updates cause replacement.
      - Required property.
      - Type: String

Notice

For the Japan East2 regions

- Can be updated without replacement.
- Optional property.
- List contents:
  - Type: List
  - * : Map
  - Can be updated without replacement.
  - Optional property.
  - Map properties:
    - destination
      - Can be updated without replacement.
      - Required property.
      - Type: String
    - nexthop
      - Can be updated without replacement.
      - Required property.
      - Type: String

ip_version

- The IP version

  Updates cause replacement.
- Optional property, defaults to "4".
- Type: Integer
- Allowed values: 4

name

- The name of the subnet.

  Can be updated without replacement.
- Optional property.
- Type: String

tenant_id

- The ID of the tenant who owns the network. Only administrative users can specify a tenant ID other than their own.

  Updates cause replacement.
- Optional property.
- Type: String

network_id

- Updates cause replacement.
- Required property.
- Type: String

availability_zone

- The Availability Zone name.

  If you don't specify, the resource will be created in default AZ.

  Updates cause replacement.
- Optional property.
- Type: String

5.1.5.11.4 Attributes

allocation_pools

Ip allocation pools and their ranges.

cidr

CIDR block notation for this subnet.

dns_nameservers

List of dns nameservers.

enable_dhcp

'true' if DHCP is enabled for this subnet; 'false' otherwise.

gateway_ip

Ip of the subnet's gateway.

host_routes

Additional routes for this subnet.

ip_version

Ip version for the subnet.

name

Friendly name of the subnet.

network_id

Parent network of the subnet.

show

All attributes.

tenant_id

Tenant owning the subnet.

availability_zone

The Availability Zone name.

5.1.5.12 FCX::Neutron::NetworkConnector

5.1.5.12.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: FCX::Neutron::NetworkConnector
    properties:
      name: String
      network_connector_pool_id: String
```

5.1.5.12.2 Notes

None.

5.1.5.12.3 Properties

name

- Name for network connector.

  Can be updated without replacement.
- Optional property.
- Type: String

network_connector_pool_id

- A network connector pool id for this network connector. When this value not specified and only one pool exists, use it.

  Updates cause replacement.
- Optional property.
- Type: String

tenant_id

- When requester has admin privileges, requester is able to specify any tenant's id as owner of this network connector. Otherwise, this parameter is restricted for tenant in which requester joins.

  Updates cause replacement.
- Optional property.
- Type: String

5.1.5.12.4 Attributes

name

Name for network connector.

network_connector_pool_id

A network connector pool id for this network connector

tenant_id

Tenant's ID to which this network connector belongs

5.1.5.13 FCX::Neutron::NetworkConnectorEndpoint

5.1.5.13.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: FCX:: Neutron::NetworkConnectorEndpoint
    properties:
      name: String
      network_connector_id: String
      endpoint_type: String
location: String
```

5.1.5.13.2 Notes

None.

5.1.5.13.3 Properties

name

- Name for this network connector endpoint.

  Can be updated without replacement.
- Optional property.
- Type: String

network_connector_id

- ID of network connector to which this network connector endpoint belongs.

  Updates cause replacement.
- Required property.
- Type: String

endpoint_type

- type of this network connector endpoint. This value must be one of "availability_zone" or "remote".

  Updates cause replacement.
- Required property.
- Type: String

location

- location of this network connector endpoint in the endpoint_type. When type is "availability_zone", this value must be one of availability zone name. When type is "remote", this value express label of location, such as 'intra'.

  Updates cause replacement.
- Required property.
- Type: String

tenant_id

- When requester has admin privileges, requester is able to specify any tenant's id as owner of this network connector. Otherwise, this parameter is restricted for tenant in which requester joins.

  Updates cause replacement.
- Optional property.
- Type: String

5.1.5.13.4 Attributes

name

Name for this network connector endpoint.

network_connector_id

ID of network connector to which this network connector endpoint belongs.

endpoint_type

type of this network connector endpoint.

location

Location of this network connector endpoint profile in the endpoint_type.

tenant_id

Tenant's ID to which this network connector belongs

5.1.5.14 FCX::Neutron::NetworkConnectorEndpointConnection

5.1.5.14.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: FCX:: Neutron:: NetworkConnectorEndpointConnection
    properties:
      network_connector_endpoint_id: String
      port_id: String
```

5.1.5.14.2 Notes

None.

5.1.5.14.3 Properties

network_connector_endpoint_id

- ID for this network connector endpoint.
  Updates cause replacement.
- Required property.
- Type: String

port_id

- Port resource on the availability zone. The port's device_owner must be 'network:router_interface'.
  Updates cause replacement.
- Required property.
- Type: String

5.1.5.14.4 Attributes

None.

5.1.6 Expandable Load Balancing

5.1.6.1 HOT Syntax

The ELB template format is as below.

```
heat_template_version: 2013-05-23
description:
resources:
  elb:
    type: FJ::ExpandableLoadBalancer::LoadBalancer
    properties:
        LoadBalancerName: String
        Subnets: [String, String]
        Listeners: [{
            'LoadBalancerPort': Integer,
            'InstancePort': Integer,
            'Protocol': String,
            'InstanceProtocol': String,
            'SSLCertificateId': String
            }]
        HealthCheck: {
            'Interval': Integer,
            'Target': String,
            'HealthyThreshold': Integer,
            'Timeout': Integer,
            'UnhealthyThreshold': Integer
            }
        Instances: [String, String]
        InstancesPorts: [{
            'InstanceId': String,
            'PortId': String
            }]
        LBCookieStickinessPolicies: [{
            'PolicyName': String,
            'CookieExpirationPeriod': Integer
            }]
```

```
        SorryServerRedirectionPolicies: [{
            'PolicyName': String,
            'Location': String
            }]
        OtherPolicies: [{
              'PolicyName': String,
              'PolicyTypeName': String,
              'PolicyAttributes': [{
                     'AttributeName': String,
                     'AttributeValue': String
                     }]
              }]
        ListenersPolicies: [{
            'PolicyNames': [String, String],
            'LoadBalancerPort': Integer
             }]
SecurityGroups: [String, String]
Grade: String
        Scheme: String
        LoadBalancerAttributes: {
             'ConnectionSettings': {
                   'IdleTimeout': Integer
                  }
             }
        Version: String
```

5.1.6.2 FJ::ExpandableLoadBalancer::LoadBalancer

```
description:
resources:
  elb:
    type: FJ::ExpandableLoadBalancer::LoadBalancer
    properties:
        LoadBalancerName: String
        Subnets: List
        Listeners: List
        List contents:
        * : Map
            'LoadBalancerPort': Integer,
            'InstancePort': Integer,
            'Protocol': String,
            'InstanceProtocol': String,
            'SSLCertificateId': String
        HealthCheck:
        * : Map
            'Interval': Integer,
            'Target': String,
            'HealthyThreshold': Integer,
            'Timeout': Integer,
            'UnhealthyThreshold': Integer
        Instances: List
        InstancesPorts: List
        List contents:
        * : Map
            'InstanceId': String,
            'PortId': String
        LBCookieStickinessPolicies: List
        List contents:
        * : Map
            'PolicyName': String,
            'CookieExpirationPeriod': Integer
        SorryServerRedirectionPolicies: List
        List contents:
        * : Map
            'PolicyName': String,
            'Location': String
        OtherPolicies: List
        List contents:
```

```
        * : Map
            'PolicyName': String,
            'PolicyTypeName': String
            'PolicyAttributes': List
            List contents:
            * : Map
                'AttributeName': String,
                'AttributeValue': String
        ListenersPolicies: List
        List contents:
        * : Map
            'PolicyNames': List,
            'LoadBalancerPort': Integer
        SecurityGroups: List
        Grade: String
        Scheme: String
        LoadBalancerAttributes:
        * : Map
            ConnectionSettings:
            * : Map
                'IdleTimeout': Integer
        Version: String
```

5.1.6.2.1 Basic parameters

LoadBalancerName

The load balancer name. The name must be unique within the set of load balancers associated with the user account.

- Type : String
- Default : -
- Required : Yes
- Valid values : -
- Updated : No

Grade

The load balancer grade (performance type).

Settable strings are "Standard", "Middle", and "High".

- Type : String
- Default : Standard
- Required : No
- Valid values : Standard, Middle, High
- Updated : No

Scheme

The load balancer type.

Settable strings are "public" and "internal".

- Type : String
- Default : public
- Required : No
- Valid values : public, internal
- Updated : No

Version

The version of the API associated with the written requirements. Format: Indicated as YYYY-MM-DD.

- Type : String
- Default : -
- Required : Yes
- Valid values : -
- Updated : No

5.1.6.2.2 Subnets parameters

Subnets

A list of subnet IDs.

The subnet ID type is String.

- Type : List
- Default : -
- Required : Yes
- Valid values : -
- Updated : Yes

5.1.6.2.3 Listeners parameters

Listeners

A list of listeners including their data types, LoadBalancerPort, InstancePort, and Protocol details.

- Type : List
- Default : -
- Required : Yes
- Valid values : -
- Updated : Yes
  - LoadBalancerPort

    The front-end port number. Unchangeable for load balancers with "InService" status.
    - Type : Integer
    - Default : -
    - Required : Yes
    - Valid values : -
    - Updated : Yes
  - InstancePort

    The TCP port number of the distribution destination server. Unchangeable for load balancers with "InService" status.

    Only one InstancePort can be specified for each load balancer.
    - Type : Integer
    - Default : -
    - Required : Yes
    - Valid values : -
    - Updated : Yes
  - Protocol

    The load balancer's transport protocol (HTTP, HTTPS, TCP, or SSL). Unchangeable for load balancers with "InService" status.

    Settable strings are "HTTP", "HTTPS", "TCP", "SSL", "http", "https", "tcp", and "ssl".
    - Type : String
    - Default : -

- Required : Yes
- Valid values : HTTP, HTTPS, TCP, SSL, http, https, tcp, ssl
- Updated : Yes

- InstanceProtocol

  The protocol (HTTP, HTTPS, TCP, or SSL) used for routing traffic to back-end instances. Unchangeable for load balancers with "InService" status. Only one InstanceProtocol can be specified for each load balancer.

  Settable strings are "HTTP", "HTTPS", "TCP", "SSL", "http", "https", "tcp", and "ssl".

  - Type : String
  - Default : -
  - Required : Yes
  - Valid values : HTTP, HTTPS, TCP, SSL, http, https, tcp, ssl
  - Updated : Yes

- SSLCertificateId

  The server certificate ID. Only one server certificate can be specified for each load balancer. If a different server certificate is specified for each listener, the last server certificate specified is enabled.

  - Type : String
  - Default : -
  - Required : No
  - Valid values : -
  - Updated : Yes

5.1.6.2.4 HealthCheck parameters

HealthCheck

Health check configuration information

- Type : Map
- Default : -
- Required : No
- Valid values : -
- Updated : Yes

  - HealthyThreshold

    The number of consecutive successful health checks based on which the target instance is determined to have been recovered from failure and is incorporated into the assignment destination.

    - Type : Integer
    - Default : -
    - Required : Yes
    - Valid values : -
    - Updated : Yes

  - Interval

    Health checking time interval (seconds).

    - Type : Integer
    - Default : -
    - Required : Yes
    - Valid values : -
    - Updated : Yes

  - Target

The protocol, port number, and URL of a health check instance.

- Specify these in the following format.

```
protocol:port[url]
```

  - Specify protocol with TCP, HTTP, HTTPS, or SSL.
  - Specify port with a value in the range of 1 to 65535.
  - Specify url with the URL path. This may be omitted. This can be specified if protocol is HTTP or HTTPS.
- Type : String
- Default : -
- Required : Yes
- Valid values : -
- Updated : Yes

- Timeout

  The period of timeout for a health check response, in seconds. This value must be smaller than the Interval value.
  - Type : Integer
  - Default : -
  - Required : Yes
  - Valid values : -
  - Updated : Yes
- UnhealthyThreshold

  The number of consecutive unsuccessful health checks based on which the target instance is determined to have failed and is eliminated from the assignment destination.
  - Type : Integer
  - Default : -
  - Required : Yes
  - Valid values : -
  - Updated : Yes

5.1.6.2.5 Instances parameters

Instances

A list of instance IDs registered in the load balancer.

The instance ID type is String.

- Type : List
- Default : -
- Required : No
- Valid values : -
- Updated : Yes
- For example:

```
Instances: ['instance1uuid']
```

5.1.6.2.6 InstancesPorts parameters

InstancesPorts

A list of instance ID and port ID combinations registered in the load balancer.

- Type : List
- Default : -

- Required : No
- Valid values : -
- Updated : Yes
  - InstanceId

    The instance ID.
    - Type : String
    - Default : -
    - Required : Yes
    - Valid values : -
    - Updated : Yes
  - PortId

    The instance port ID.

    The target port ID in cases where the instance has multiple ports.
    - Type : String
    - Default : -
    - Required : Yes
    - Valid values : -
    - Updated : Yes
    - For example:

```
InstancesPorts: [{'InstanceId': 'instance1uuid', 'PortId': 'instance1portuuid'}]
InstancesPorts: [{'InstanceId': 'instance1uuid', 'PortId': 'instance1portuuid'},
                                        {'InstanceId': 'instance2uuid', 'PortId':
 'instance2portuuid'}]
```

CAUTION

Note that the Instances and the InstancesPorts parameters are non-exclusive.

- Below is a normal example.

```
Instances: ['instance1uuid']
InstancesPorts: [{'InstanceId': 'instance2uuid', 'PortId':
 'instance2portuuid'}]
```

- Below is an abnormal example. (Only one of the two parameters can be used for the same instance.)

```
Instances: ['instance1uuid']
InstancesPorts: [{'InstanceId': 'instance1uuid', 'PortId':
 'instance1portuuid'}]
```

5.1.6.2.7 LBCookieStickinessPolicies parameters

LBCookieStickinessPolicies

A list of policies to create

- Type : List
- Default : -
- Required : No
- Valid values : -
- Updated : Yes
  - PolicyName

    The name of the policy to create. The name must be unique among policies available for the target load balancer.
    - Type : String
    - Default : -
    - Required : Yes

- Valid values : -
- Updated : Yes

- CookieExpirationPeriod

  Specifies the maximum period during which a cookie maintains the session in units of seconds. By default, the expiration period is not set.

  CAUTION
  CookieExpirationPeriod may not be changed alone. To change "CookieExpirationPeriod", it is also necessary to change "PolicyName".

  - Type : Integer
  - Default : -
  - Required : No
  - Valid values : -
  - Updated : Yes*

5.1.6.2.8 SorryServerRedirectionPolicies parameters

SorryServerRedirectionPolicies

A list of policies for redirection to SorryServer

- Type : List
- Default : -
- Required : No
- Valid values : -
- Updated : Yes

- PolicyName

  The name of the policy to create.

  - Type : String
  - Default : -
  - Required : Yes
  - Valid values : -
  - Updated : Yes

- Location

  The URI of the location that is the redirection destination

  CAUTION
  Location may not be changed alone. To change "Location", it is also necessary to change "PolicyName".

  - Type : String
  - Default : -
  - Required : Yes
  - Valid values : -
  - Updated : Yes*

5.1.6.2.9 OtherPolicies parameters

OtherPolicies

A list of policies related to the SSL encryption protocol

- Type : List
- Default : -
- Required : No

- Valid values : -
- Updated : Yes
  - PolicyName
    The name of the policy to create.
    - Type : String
    - Default : -
    - Required : Yes
    - Valid values : -
    - Updated : Yes
  - PolicyTypeName
    The type of the policy to create.
    - Type : String
    - Default : -
    - Required : Yes
    - Valid values : SSLNegotiationPolicyType
    - Updated : Yes
  - PolicyAttributes
    A list of attributes associated with the policy.
    - Type : String
    - Default : -
    - Required : Yes
    - Valid values : -
    - Updated : Yes
      - AttributeName
        The attribute name of the policy
        - Type : String
        - Default : -
        - Required : Yes
        - Valid values : Protocol-TLSv1, Protocol-SSLv3, Protocol-TLSv1.1, Protocol-TLSv1.2
        - Updated : Yes
      - AttributeValue
        The attribute value of the policy
        - Type : String
        - Default : -
        - Required : Yes
        - Valid values : true, false
        - Updated : Yes

5.1.6.2.10 ListenersPolicies parameters

ListenersPolicies

A list of listener port numbers and policies configured for front-end connections.

- Type : List
- Default : -
- Required : No
- Valid values : -
- Updated : Yes
  - PolicyNames

A list of policies applied to listeners.

Specifies "LBCookieStickinessPolicies" or "SorryServerRedirectionPolicies". If more than one line is specified, only the value specified last is set for "LBCookieStickinessPolicies".

If the list is blank, the current policies are deleted from listeners.

- Type : List
- Default : -
- Required : No
- Valid values : -
- Updated : Yes

- LoadBalancerPort

  The port number for front-end connections configured for the listener the policy is applied to.

  - Type : Integer
  - Default : -
  - Required : Yes
  - Valid values : -
  - Updated : Yes

5.1.6.2.11 SecurityGroups parameters

SecurityGroups

A list of security group IDs.

The security group ID type is String.

- Type : List
- Default : -
- Required : No
- Valid values : -
- Updated : Yes

5.1.6.2.12 LoadBalancerAttributes parameters

LoadBalancerAttributes

Load balancer attributes

- Type : Map
- Default : -
- Required : No
- Valid values : -
- Updated : Yes
  - ConnectionSettings

    ConnectionSettings attributes

    - Type : Map
    - Default : -
    - Required : No
    - Valid values : -
    - Updated : Yes
      - IdleTimeout

        The period during which front-end and back-end connections are maintained in the idle state.

        This period is set in units of seconds. Minimum value: 1, Maximum value: 3600

・Type : Integer
・Default : 60
・Required : Yes
・Valid values : [1,3600]
・Updated : Yes

5.1.6.3 Description of Attributes parameters

The Attributes parameters are runtime data that can be disclosed to other resources within the stack.
The attributes that can be defined for the ELB plugin are as below.

・LoadBalancerName

The name of the load balancer

・DNSName

The DNS name of the load balancer

5.1.7 Database

5.1.7.1 HOT Syntax

The format of the Database template is as follows.

```
heat_template_version: 2013-05-23
description: database plugin test

resources:
  Test_db_instance:
    type: FCX::Database::DBInstance
    properties:
      name: String
      flavor: String
      size: Integer,
      disk_type: String
      id: String
      availability_zone: String
      subnet_group_id: String
      multi_az: Boolean
      port: Integer
      preferred_backup_window: String
      preferred_maintenance_window: String
      publicly_accessible: Boolean
      security_group_ids: [ Value, Value...]
      parameter_group_id: String
      backup_retention_period: Integer
      auto_minor_version_upgrade: Boolean
      engine: String
      engine_version: String
      masteruser_password: String
      character_set: String
      collate: String
      databases: [{"name": String},...]
      users: [{"name": String, "password": String, databases: [Value, Value,...]},...]

  Test_db_subnetgroup:
    type: FCX::Database::DBSubnetGroup
    properties:
      id: String
      name: String
      subnet_ids: [{"subnet_id":String}, {"subnet_id": String},...]
      description: String
```

```
Test_db_parametergroup:
  type: FCX::Database::DBParameterGroup
  properties:
    id: String
    name: String
    parameter_group_family: String
    description: String
```

5.1.7.2 FCX::Database::DBInstance

This section describes the parameters that can be specified for creating DB instances.

5.1.7.2.1 Description of properties parameters

Basic parameters

1. List of parameters

Parameter	Description	Type	Mandatory	Constraints/ default value	Remarks
Flavor	List ID of the predefined hardware resource	String	Y		Specifies the flavor ID. The flavor ID can be retrieved by using the get flavor list (GET /v1.0/{tenantId}/flavors) and get flavor information (GET /v1.0/{tenantId}/flavors/{flavorId}) APIs
Size	Size of the data volume	Integer	Y	10-10240	
disk_type		String	N	M1	
availability_zone	Availability zone where the instance is created	String	Y		jp-east-1a | jp-east-1b
subnet_group_id	Subnet group where the DB instance will be deployed to	String	Y	*Only subnets with DHCP ON can be specified	It is necessary to have a set of subnets that include at least two availability zones The specifiable values can be retrieved using the get DB subnet group list API (GET /v1.0/{tenantId}/subnetgroups)

Parameter	Description	Type	Mandatory	Constraints/ default value	Remarks
publicly_ accessible	Whether an Internet connection is possible during deployment of the DB instance	Boolean	N	true | false Default value: true	*true: External Internet connections are permitted If True is specified, the subnet specified in subnet_group_id must be connected to ext-net false: Access from within VPC only
security_ group_ids.	ID list of the VPC security group	List (String list)	N	List of the VPC security group Default: default security group	*Specify the VPC security group. The specifiable values can be retrieved by executing the Networking service API
multi	Multi DB option	Boolean	N	true | false Default value: false	
multi_az	Multi availability zone option	Boolean	N	true | false Default value: false	
Id	ID of the DB instance	String	N	Default : random value	
name	Name of the DB instance	String	N	Default : Random value	
backup_ retention_ period	Backup retention period (in days)	Integer	N	0 to 10 Default: 0	*Automatic backup will not be performed when the value is 0.

Parameter	Description	Type	Mandatory	Constraints/ default value	Remarks
preferred_ backup_ window	Backup time slot	String	N	Format: hh24: mi-hh24:mi Default: A random thirty-minute-period within the ten-hour-period prescribed for each region	*If automatic backups are enabled, specify the timeslot when the daily backups are to be performed *It is necessary to specify a timeslot from thirty minutes to twenty-three hours and 30 minutes • Eastern Japan Region 1 (jp-east-1): 17:00-03:00 UTC • Western Japan Region 1 (jp-west-1): 17:00-03:00 UTC *Specify times in UTC format *You cannot specify a timeslot that overlaps the preferred maintenance window *Backups may take longer than thirty minutes, depending on the backup conditions
preferred_ maintenance_ window	Maintenance time slot	String	N	Format: ddd:hh24:mi-ddd:hh24:mi Default: A random thirty-minute period within the ten-hour period prescribed for each region (the day of the week is also determined randomly)	*Specify the timeslot when weekly maintenance is to be performed. *It is necessary to specify a timeslot from thirty minutes to twenty-three hours and 30 minutes A thirty-minute period within the ten-hour period prescribed for each region will be randomly determined (the day of the week is also determined randomly) Example: Sun:0500-Sun: 06:00 Day value: Mon, Tue, Wed, Thu, Fri, Sat, Sun

Parameter	Description	Type	Mandatory	Constraints/ default value	Remarks
auto_minor_ version_ upgrade	Automatic minor version upgrade	Boolean	N	Default: true	True: Perform automatic minor version upgrade
port	Port number	Integer	N	1024 to 32767 Default value: 26500	
masteruser_ name	Administrator user name	String	N	Default : postgres • Up to 63 alpha-numeric characters can be used • The first character must be an alphabetic character or an underscore	
masteruser_ password	Administrator password	String	Y	Up to 1024 characters	
character_set	Character encoding	String	N	Default: UTF8	The specifiable values can be retrieved using get DB engine information (GET /v1.0/{tenantId}/ engineversion)
collate	Collating sequence	String	N	Default: C	The specifiable values can be retrieved using get DB engine information (GET /v1.0/{tenantId}/ engineversion)

Parameter	Description	Type	Mandatory	Constraints/ default value	Remarks
parameter_ group_id	Name of the DB parameter group	String	N	DB parameter group ID Default: default parameter group	The specifiable values can be retrieved using get DB parameter group list (GET /v1.0/{tenantId}/ parametergroups)
engine	Name of the DB engine	String	N	enterprise postgres | symfoware Default: enterprise postgres	
engine_version	DB version	String	N	Default: Latest version	The specifiable values can be retrieved using get DB engine information (GET /v1.0/{tenantId}/ engineversion)
databases	List of the DB structure	Map (Data base struc ture)	N		
users	List of the user structure	Map (User struc ture)	N		

Parameter	Description	Type	Mandatory	Constraints/ default value	Remarks
auto_ maintenance	Whether auto maintenance is to be performed English: Whether an auto maintenance is possible	Boolean	N	true | false Default: true	If this parameter is set to false, no auto maintenance is performed even if preferred MaintenanceWindow is encountered. If it is changed to true, auto maintenance starts with the next preferred MaintenanceWindow.

2. Database structure

Parameters	Description	Type	Mandatory	Value	Remarks
name	DB name	String	Y	DB identifier • Up to 63 alpha-numeric characters can be used • The first character must be an alphabetic character or an underscore	

3. User structures

Parameters	Description	Type	Mandatory	Value	Remarks
name	DB user name	String	Y	• Up to 63 alpha-numeric characters can be used • The first character must be an alphabetic character or an underscore	
password	Password	String	Y	Up to 1024 characters	
databases	Databases that this user can log in to	List (String list)	Y		

5.1.7.2.2 Description of the attributes parameters

Attributes is data as of the execution time that can be disclosed to other resources in a stack.
The attributes that can be retrieved using the DBaaS plug-in are as follows.

Name	Description
PUBLICADDRESS	External address of a Database instance (FQDN)
PRIVATEADDRESS	Internal address of a Database instance (FQDN)
PRIVATEIP	Internal IP address of a Database instance
PUBLICIP	External IP address of a Database instance
SUBPRIVATEIP	Internal IP address of a Standby Database instance
SUBPUBLICIP	External IP address of a Standby Database instance

5.1.7.3 FCX::Database::DBSubnetGroup

5.1.7.3.1 Description of properties parameters

Basic parameters

1. List of parameters

Parameters	Description	Type	Mandatory	Constraints/ default value	Remarks
id	ID of DB subnet group	String	N		
name	Name of DB subnet group	String	Y	Default: random value	
subnet_ids	List of subnet structure	MAP (SubnetId structures)	Y		
description	Description of DB subnet group	String	N	Default: None	

2. SubnetId structures

Parameters	Description	Type	Mandatory	Value	Remarks
subnet_id	ID of subnet	String	Y		

5.1.7.4 FCX::Database::DBParameterGroup

5.1.7.4.1 Description of properties parameters

Basic parameters

1. List of parameters

Parameters	Description	Type	Mandatory	Constraints/ default value	Remarks
parameterGroup Family	Type of parameter group, determined by the DB engine and version	String	Y	enterprise postgres_v9.6 | symfoware_ v12.1	
id	ID of the DB parameter group	String	N	Default : random value	
parameterGroup Name	Name of the DB parameter group	String	Y		
description	Description of the DB parameter group	String	N	Default: None	

5.1.8 Object Storage

5.1.8.1 OS::Swift::Container

5.1.8.1.1 HOT Syntax

```
heat_template_version: 2013-05-23
...
resources:
  ...
  the_resource:
    type: OS::Swift::Container
    properties:
      X-Account-Meta: {...}
      X-Container-Meta: {...}
      X-Container-Read: String
      X-Container-Write: String
      name: String
```

5.1.8.1.2 Notes

None.

5.1.8.1.3 Properties

X-Account-Meta

- A map of user-defined meta data to associate with the account. Each key in the map will set the header X-Account-Meta-{key} with the corresponding value.
- Updates cause replacement.
- Optional property, defaults to “{}”.
- Type: Map

X-Container-Meta

- A map of user-defined meta data to associate with the container. Each key in the map will set the header X-Container-Meta-{key} with the corresponding value.
- Updates cause replacement.
- Optional property, defaults to “{}”.
- Type: Map

X-Container-Read

- Specify the ACL permissions on who can read objects in the container.
- Updates cause replacement.
- Optional property.
- Type: String

X-Container-Write

- Specify the ACL permissions on who can write objects to the container.
- Updates cause replacement.
- Optional property.
- Type: String

name

- Name for the container. If not specified, a unique name will be generated.
- Updates cause replacement.
- Optional property.
- Type: String

5.1.8.1.4 Attributes

BytesUsed

The number of bytes stored in the container.

DomainName

The host from the container URL.

HeadContainer

A map containing all headers for the container.

ObjectCount

The number of objects stored in the container.

RootURL

The parent URL of the container.

WebsiteURL

The URL of the container.

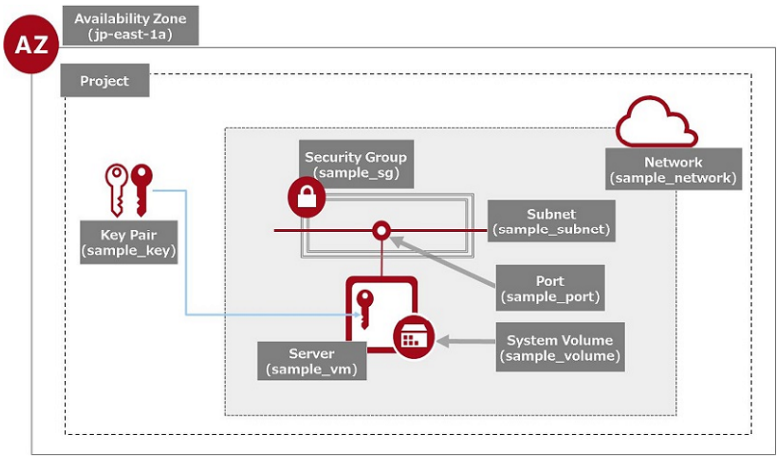
Part 6: Example templates

Topics:

6.1 Sample system configuration - Example Heat template

Heat_template_sample.yaml

Below is an example of a template for creating resources.



Heat_template_sample.yaml

```
#
# This is a hello world HOT template just defining a single compute
# server.
#
heat_template_version: 2013-05-23

description: >
  This HOT template that just defines a single server and network.
  Contains just base features to verify base HOT support.

parameters:
  az:
    type: string
    description: availability zone
    default: jp-east-1a

  network_name:
    type: string
    description: name of network
    default: sample_network

  subnet_name:
    type: string
    description: name of subnet
    default: sample_subnet

  subnet_cidr:
    type: string
    description: subnet CIDR
    default: 192.168.0.0/24

  port_name:
    type: string
    description: name of vm
    default: sample_port

  sg_name:
    type: string
    description: security group
    default: sample_sg
```

```
  key_name:
    type: string
    description: name of keypair
    default: sample_key

  image:
    type: string
    description: Image ID or image name to use for the server
    default: 383ed3f8-0773-4b14-96c8-feb387dd3935

  volume_name:
    type: string
    description: name of volume
    default: sample_volume

  flavor:
    type: string
    description: Flavor for the server to be created
    default: T-1

  vm_name:
    type: string
    description: name of vm
    default: sample_vm


resources:
  network:
    type: OS::Neutron::Net
    properties:
      name: { get_param : network_name }
      availability_zone: { get_param : az }

  subnet:
    type: OS::Neutron::Subnet
    properties:
      name: { get_param : subnet_name }
      network_id: { get_resource : network }
      availability_zone: { get_param : az }
      cidr: { get_param : subnet_cidr }

  port:
    type: OS::Neutron::Port
    properties:
      name: { get_param : port_name }
      network_id: { get_resource: network }
      availability_zone: { get_param: az }
      security_groups:
        - {get_resource: sg }
      fixed_ips:
        - subnet_id: { get_resource: subnet }

  sg:
    type: OS::Neutron::SecurityGroup
    properties:
      name: { get_param : sg_name }
      rules:
        # HTTP
        - { direction: egress, ethertype: IPv4, port_range_min: 80, port_range_max: 80,
 protocol: tcp, remote_ip_prefix: 0.0.0.0/0 }
        # HTTPS
        - { direction: egress, ethertype: IPv4, port_range_min: 443, port_range_max: 443,
 protocol: tcp, remote_ip_prefix: 0.0.0.0/0 }
        # DNS
        - { direction: egress, ethertype: IPv4, port_range_min: 53, port_range_max: 53,
 protocol: tcp, remote_ip_prefix: 0.0.0.0/0 }
        - { direction: egress, ethertype: IPv4, port_range_min: 53, port_range_max: 53,
 protocol: udp, remote_ip_prefix: 0.0.0.0/0 }
```

```
  key:
    type: OS::Nova::KeyPair
    properties:
      name: { get_param: key_name }
      save_private_key: true
      availability_zones: [{ get_param: az }]

  sys-vol:
    type: OS::Cinder::Volume
    properties:
      name: { get_param: volume_name }
      size: 30
      volume_type: "M1"
      availability_zone: { get_param: az }
      image : { get_param: image }

  server:
    type: OS::Nova::Server
    properties:
      key_name: { get_resource: key }
      image: { get_param: image }
      flavor: { get_param: flavor }
      networks: ["port": {get_resource: port} ]
      name: { get_param: vm_name }
      block_device_mapping:
      - device_name: vda
        volume_id: {get_resource: sys-vol}

outputs:
  private_key:
    description: private key of created key pair
    value: { get_attr: [key, private_key] }
```

In this template, based on the system configuration, the virtual network and the virtual server are created inside the same stack, but to improve readability users are advised to divide resources to prevent inter-resource dependency.

In addition, dividing templates makes it possible to reduce the range of effect when an error occurs.

Regarding the example template for a virtual network and a virtual server, see below.

6.2 Example virtual server creation Heat template

6.2.1 Example virtual server creation Heat template

The following is an example template that creates a volume and a single server using the specified key pair.

Creating_a_virtual_server.yaml

```
#
# This template creates a volume and a single virtual server using
# the specified key pair.
#
heat_template_version: 2013-05-23

description: >
  This template creates a virtual server using the specified key pair.

parameters:
  VOLUME_NAME:
    type: string
    description: Name of the system volume of the Virtual Server
```

```
    default: Sample_Volume_01

  VOLUME_SIZE:
    type: string
    description: Volume size of the system volume of the Virtual Server
    default: 30

  AZ:
    type: string
    description: Name of the Availability Zone to deploy the Virtual Server in
    default: jp-east-1a

  FLAVOR:
    type: string
    description: Name or ID of the Flavor type of the Virtual Server to create
    default: S-1

  IMAGE_ID:
    type: string
    description: Image ID or image name to use for the Virtual Server
    default: c3867e5e-afd6-4858-918e-c445f9041c9d

  KEY_NAME:
    type: string
    description: Name of the existing key pair used for the Virtual Server
    default: Sample_Key_01

  SERVER_NAME:
    type: string
    description: Name of the Virtual Server to create
    default: Sample_Server_01

  NETWORK_NAME:
    type: string
    description: NAME of the Network to deploy the Virtual Server on
    default: Sample_Network_01

  NETWORK_ID:
    type: string
    description: ID of the Network to deploy the Virtual Server on
    default: 6eft72d8-a4ad-4h4f-8d2d-76soe8ba122b

  SECURITY_GROUP_NAME:
    type: string
    description: ID of the Security Group associated with the Virtual Server
    default: Sample_Security_Group_01

resources:
  VOLUME_01:
    type: OS::Cinder::Volume
    properties:
      name: { get_param: VOLUME_NAME }
      size: { get_param: VOLUME_SIZE }
      volume_type: "M1"
      image : { get_param: IMAGE_ID }
      availability_zone: { get_param: AZ }

  SERVER_01:
    type: OS::Nova::Server
    properties:
      availability_zone: { get_param: AZ }
      block_device_mapping: [{"volume_size":{ get_param: VOLUME_SIZE }, "volume_id":
 { get_resource: VOLUME_01 }, "delete_on_termination": True, "device_name": "/dev/vda" }]
      flavor: { get_param: FLAVOR }
      flavor_update_policy: RESIZE
      image: { get_param: IMAGE_ID }
      key_name: { get_param: KEY_NAME }
      name: { get_param: SERVER_NAME }
```

```
      networks: [{"network": { get_param: NETWORK_ID }}]
      security_groups: [{ get_param: SECURITY_GROUP_NAME }]
      user_data_format: RAW


outputs:
  SERVER_01_DETAIL:
    description:
    value: { get_attr: [SERVER_01, show] }

  SERVER_01_NETWORK:
    description:
    value: { get_attr: [SERVER_01, networks] }

  SERVER_01_IP:
    description:
    value: { get_attr: [SERVER_01, addresses, { get_param: NETWORK_NAME }, 0, port] }
```

6.2.2 Example Heat template for changing the flavor of a virtual server using stack update

This template performs "Updating a stack" for a virtual server that was created using "Creating_a_virtual_server.yaml" of Example virtual server creation Heat template, and changes the flavor of the virtual server.

For the procedure for "Updating a stack", refer to "Updating a stack".

Updating_a_virtual_server.yaml

```
#
# This template changes the flavor of a virtual server created
# using "Creating_a_virtual_server.yaml".
#
heat_template_version: 2013-05-23

description: >
  This template changes the flavor of a virtual server created using
 "Creating_a_virtual_server.yaml".

parameters:
  VOLUME_NAME:
    type: string
    description: Name of the system volume of the Virtual Server
    default: Sample_Volume_01

  VOLUME_SIZE:
    type: string
    description: Volume size of the system volume of the Virtual Server
    default: 30

  AZ:
    type: string
    description: Name of the Availability Zone to deploy the Virtual Server in
    default: jp-east-1a

  FLAVOR:
    type: string
    description: Name or ID of the Flavor type of the Virtual Server to update
    default: S-2

  IMAGE_ID:
    type: string
    description: Image ID or image name to use for the Virtual Server
    default: c3867e5e-afd6-4858-918e-c445f9041c9d
```

```
  KEY_NAME:
    type: string
    description: Name of the existing key pair used for the Virtual Server
    default: Sample_Key_01

  SERVER_NAME:
    type: string
    description: Name of the Virtual Server to update
    default: Sample_Server_01

  NETWORK_NAME:
    type: string
    description: NAME of the Network to deploy the Virtual Server on
    default: Sample_Network_01

  NETWORK_ID:
    type: string
    description: ID of the Network to deploy the Virtual Server on
    default: 6eft72d8-a4ad-4h4f-8d2d-76soe8ba122b

  SECURITY_GROUP_NAME:
    type: string
    description: NAME of the Security Group associated with the Virtual Server
    default: Sample_Security_Group_01

resources:
  VOLUME_01:
    type: OS::Cinder::Volume
    properties:
      name: { get_param: VOLUME_NAME }
      size: { get_param: VOLUME_SIZE }
      volume_type: "M1"
      image : { get_param: IMAGE_ID }
      availability_zone: { get_param: AZ }

  SERVER_01:
    type: OS::Nova::Server
    properties:
      availability_zone: { get_param: AZ }
      block_device_mapping: [{"volume_size":{ get_param: VOLUME_SIZE }, "volume_id":
 { get_resource: VOLUME_01 }, "delete_on_termination": True, "device_name": "/dev/vda" }]
      flavor: { get_param: FLAVOR }
      flavor_update_policy: RESIZE
      image: { get_param: IMAGE_ID }
      key_name: { get_param: KEY_NAME }
      name: { get_param: SERVER_NAME }
      networks: [{"network": { get_param: NETWORK_ID }}]
      security_groups: [{ get_param: SECURITY_GROUP_NAME }]
      user_data_format: RAW


outputs:
  SERVER_01_DETAIL:
    description:
    value: { get_attr: [SERVER_01, show] }

  SERVER_01_NETWORK:
    description:
    value: { get_attr: [SERVER_01, networks] }

  SERVER_01_IP:
    description:
    value: { get_attr: [SERVER_01, addresses,{ get_param: NETWORK_NAME }, 0, port] }
```

6.3 Example virtual server creation (Windows OS) Heat template with computer name specification

The following is an example template that specifies the computer name when creating a virtual server (Windows OS).

Indicate the computer name to specify in the following location in the template.


Notice

Example: Rename-Computer -Force -NewName "computer_name_to_specify" -Restart

Creating_a_virtual_server_for_WindowsOS.yaml

```
#
#
# This template specifies the computer name when creating a virtual server (Windows OS).
#
#
heat_template_version: 2013-05-23

description: >
  This template specifies the computer name when creating a virtual server (Windows OS).

parameters:
  VOLUME_NAME:
    type: string
    description: Name of the system volume of the Virtual Server
    default: Sample_Volume_01

  VOLUME_SIZE:
    type: string
    description: Volume size of the system volume of the Virtual Server
    default: 80

  AZ:
    type: string
    description: Name of the Availability Zone to deploy the Virtual Server in
    default: jp-east-1a

  FLAVOR:
    type: string
    description: Name or ID of the Flavor type of the Virtual Server to create
    default: S-1

  IMAGE_ID:
    description: Image ID or image name to use for the Virtual Server
    default: 0e9e37b7-5514-4e9a-95d9-b6927a74e200

  KEY_NAME:
    type: string
    description: Name of the existing key pair used for the Virtual Server
    default: Sample_Key_01

  SERVER_NAME:
    type: string
    description: Name of the Virtual Server to create
    default: Sample_Server_01

  NETWORK_NAME:
    type: string
    description: NAME of the Network to deploy the Virtual Server on
    default: Sample_Network_01

  NETWORK_ID:
    type: string
```

```
    description: ID of the Network to deploy the Virtual Server on
    default: 6eft72d8-a4ad-4h4f-8d2d-76soe8ba122b

  SECURITY_GROUP_NAME:
    type: comma_delimited_list
    description: ID of the Security Group associated with the Virtual Server
    default: Sample_Security_Group_01


resources:
  VOLUME_01:
    type: OS::Cinder::Volume
    properties:
      name: { get_param: VOLUME_NAME }
      size: { get_param: VOLUME_SIZE }
      volume_type: M1
      image : { get_param: IMAGE_ID }
      availability_zone: { get_param: AZ }

  SERVER_01:
    type: OS::Nova::Server
    properties:
      availability_zone: { get_param: AZ }
      block_device_mapping: [{"volume_size": { get_param: VOLUME_SIZE }, "volume_id":
 {get_resource: VOLUME_01 }, "delete_on_termination": True, "device_name": "/dev/vda" }]
      flavor: { get_param: FLAVOR }
      image: { get_param: IMAGE_ID }
      key_name: { get_param: KEY_NAME }
      name: { get_param: SERVER_NAME }
      networks: ["uuid": { get_param: NETWORK_ID }]
      security_groups: { get_param: SECURITY_GROUP_NAME }
      user_data_format: RAW
      user_data: |
        #ps1
        Rename-Computer -Force -NewName SampleNameWindowsV01 -Restart


outputs:
  SERVER_01_DETAIL:
    description:
    value: { get_attr: [SERVER_01, show] }

  SERVER_01_NETWORK:
    description:
    value: { get_attr: [SERVER_01, networks] }

  SERVER_01_IP:
    description:
    value: { get_attr: [SERVER_01, addresses,{ get_param: NETWORK_NAME }, 0, port] }
```

6.4 Example AutoScale Heat template

hello_world_autoscaling.yaml

The following is an example template for AutoScale.

```
heat_template_version: 2013-05-23

description:
  Autoscaling test HOT.

parameters:

  az:
    type: string
    default: jp-east-1b
```

```
  param-image-id:
    type: string
    default: 839c1db6-738c-4e2b-9a1d-c14977564203

  param-flavor:
    type: string
    default: (The server type of the virtual server.)

  key-name:
    type: string
    description: SSH key to connect to the servers
    default: (Arbitrary key pair name)

  autoscale-security-group:
    type: comma_delimited_list
    default: (Arbitrary security group name)

  subnet-id:
    type: string
    description: subnet id
    default: (Arbitrary subnet ID)

resources:

  web-server-group:
    depends_on:
      - launch_config
     - fj-elb
    type: FCX::AutoScaling::AutoScalingGroup
    properties:
      AvailabilityZones: [{get_param: az}]
      LaunchConfigurationName: {get_resource: launch_config}
      MinSize: '2'
      MaxSize: '3'
      VPCZoneIdentifier: [{get_param: subnet-id}]
      LoadBalancerNames:
      - {get_resource: fj-elb}

  launch_config:
    type: FCX::AutoScaling::LaunchConfiguration
    properties:
      ImageId: { get_param: param-image-id }
      InstanceType: { get_param: param-flavor }
      KeyName: {get_param: key-name}
      SecurityGroups: {get_param: autoscale-security-group}
      BlockDeviceMappingsV2: [{source_type: 'image', destination_type: 'volume',
 boot_index: '0', device_name: '/dev/vda', volume_size: '30', uuid: {get_param: param-
image-id}, delete_on_termination: true, volume_type: 'F2'}]

  fj-elb:
    type: FCX::ExpandableLoadBalancer::LoadBalancer
    properties:
      Subnets: [{get_param: subnet-id}]
      Listeners:
      - {LoadBalancerPort: '80', InstancePort: '80',
        Protocol: 'HTTP', InstanceProtocol: 'HTTP' }
      HealthCheck: {Target: 'HTTP:80/healthcheck', HealthyThreshold: '3',
                    UnhealthyThreshold: '5', Interval: '30', Timeout: '5'}
      Version: 2014-09-30
      Scheme: internal
      LoadBalancerName: (Arbitrary Load Balancer name)
        * Do not specify underscores (_) in the optional name for the load balancer.
          If an underscore is specified during creation, the status of the stack becomes
          "CREATE_FAILED".

  web_server_scaleup_policy:
    type: FCX::AutoScaling::ScalingPolicy
```

```
    properties:
      AdjustmentType: ChangeInCapacity
      AutoScalingGroupName: {get_resource: web-server-group}
      Cooldown: '60'
      ScalingAdjustment: '1'

  web_server_scaledown_policy:
    type: FCX::AutoScaling::ScalingPolicy
    properties:
      AdjustmentType: ChangeInCapacity
      AutoScalingGroupName: {get_resource: web-server-group}
      Cooldown: '60'
      ScalingAdjustment: '-1'

  cpu_alarm_high:
    type: OS::Ceilometer::Alarm
    properties:
      description: Scale-up if the average CPU > 50% for 1 minute
      meter_name: fcx.compute.cpu_util
      statistic: avg
      period: '60'
      evaluation_periods: '1'
      threshold: '50'
      alarm_actions:
      - {get_attr: [web_server_scaleup_policy, AlarmUrl]}
      matching_metadata: {'metadata.user_metadata.groupname': {get_resource: 'web-server-
group'}}
      comparison_operator: gt

  cpu_alarm_low:
    type: OS::Ceilometer::Alarm
    properties:
      description: Scale-down if the average CPU < 15% for 1 minute
      meter_name: fcx.compute.cpu_util
      statistic: avg
      period: '60'
      evaluation_periods: '1'
      threshold: '15'
      alarm_actions:
      - {get_attr: [web_server_scaledown_policy, AlarmUrl]}
      matching_metadata: {'metadata.user_metadata.groupname': {get_resource: 'web-server-
group'}}
      comparison_operator: lt
```

6.5 Example AutoScale Heat template (Windows)

hello_world_autoscaling_windows.yaml

The following is an example template for AutoScale (Windows).

```
heat_template_version: 2013-05-23

description:
  Autoscaling Windows

parameters:

  az:
    type: string
    default: jp-east-1a
  param_image_id:
    type: string
    default: 5ab16551-c229-4611-834b-a16e074c187e
  param_flavor:
    type: string
    default: (The server type of the virtual server.)
  autoscale_security_group_name:
```

```
    type: comma_delimited_list
    default: (Arbitrary security group name)
  autoscale_security_group_id:
    type: comma_delimited_list
    default: (Arbitrary security group ID)
  autoscale_subnet_id:
    type: comma_delimited_list
    default: (Arbitrary subnet ID)
  autoscale_elb_name:
    type: string
    default: m0918WinELB1

resources:

  AutoScaleWindows:
    type: FCX::AutoScaling::AutoScalingGroup
    properties:
      AvailabilityZones: [{get_param: az}]
      LaunchConfigurationName: {get_resource: launch_config}
      MinSize: '1'
      MaxSize: '3'
      VPCZoneIdentifier: {get_param: autoscale_subnet_id}
      HealthCheckGracePeriod: '110'
      HealthCheckType: 'ELB'
      Cooldown: 750
      LoadBalancerNames: [{get_resource: fj_elb}]
      Tags: [{"Key": "admin_pass", "Value": "(Arbitrary password (*1))"}]
        *1: Specify a password that satisfies the complexity
          requirements of Windows.

  launch_config:
    type: FCX::AutoScaling::LaunchConfiguration
    properties:
      ImageId: { get_param: param_image_id }
      InstanceType: { get_param: param_flavor }
      SecurityGroups: {get_param: autoscale_security_group_name}
      BlockDeviceMappingsV2: [{source_type: 'image', destination_type: 'volume',
 boot_index: '0', device_name: '/dev/vda', volume_size: '80', uuid: {get_param:
 param_image_id}, delete_on_termination: true, volume_type: 'F2'}]
      UserData: |
        #ps1
        New-Item "c:¥¥test" -itemType Directory

  fj_elb:
    type: FCX::ExpandableLoadBalancer::LoadBalancer
    properties:
      Subnets: {get_param: autoscale_subnet_id}
      SecurityGroups: {get_param: autoscale_security_group_id}
      Listeners:
      - {LoadBalancerPort: '80', InstancePort: '80',
        Protocol: 'HTTP', InstanceProtocol: 'HTTP' }
      HealthCheck: {Target: 'HTTP:80/iisstart.htm', HealthyThreshold: '3',
                    UnhealthyThreshold: '5', Interval: '30', Timeout: '5'}
      Version: 2014-09-30
      Scheme: internal
      LoadBalancerName: {get_param: autoscale_elb_name}

  web_server_scaleup_policy:
    type: FCX::AutoScaling::ScalingPolicy
    properties:
      AdjustmentType: ChangeInCapacity
      AutoScalingGroupName: {get_resource: AutoScaleWindows}
      ScalingAdjustment: '1'

  web_server_scaledown_policy:
    type: FCX::AutoScaling::ScalingPolicy
    properties:
      AdjustmentType: ChangeInCapacity
```

```
      AutoScalingGroupName: {get_resource: AutoScaleWindows}
      ScalingAdjustment: '-1'

  cpu_alarm_high:
    type: OS::Ceilometer::Alarm
    properties:
      description: Scale-up if the average CPU > 80% for 1 minute
      meter_name: fcx.compute.cpu_util
      statistic: avg
      period: '180'
      evaluation_periods: '1'
      threshold: '80'
      alarm_actions:
      - {get_attr: [web_server_scaleup_policy, AlarmUrl]}
      matching_metadata: {'metadata.user_metadata.groupname': {get_resource:
'AutoScaleWindows'}}
      comparison_operator: gt

  cpu_alarm_low:
    type: OS::Ceilometer::Alarm
    properties:
      description: Scale-down if the average CPU < 20% for 1 minute
      meter_name: fcx.compute.cpu_util
      statistic: avg
      period: '180'
      evaluation_periods: '1'
      threshold: '20'
      alarm_actions:
      - {get_attr: [web_server_scaledown_policy, AlarmUrl]}
      matching_metadata: {'metadata.user_metadata.groupname': {get_resource:
'AutoScaleWindows'}}
      comparison_operator: lt
```

6.6 Example health check Heat template (Windows)

hello_world_autoscaling_healthcheck.yaml

The following is an example template for automatically recovering abnormal instances included in AutoScale (Windows) that are detected using the health check of Load Balancer.

```
heat_template_version: 2013-05-23

description:
  Autoscaling Windows Health HTTP80

parameters:

  az:
    type: string
    default: jp-east-1a
  param_image_id:
    type: string
    default: 5ab16551-c229-4611-834b-a16e074c187e
  param_flavor:
    type: string
    default: (The server type of the virtual server.)
  autoscale_security_group_name:
    type: comma_delimited_list
    default: (Arbitrary security group name)
  autoscale_security_group_id:
    type: comma_delimited_list
    default: (Arbitrary security group ID)
  autoscale_subnet_id:
    type: comma_delimited_list
    default: (Arbitrary subnet ID)
```

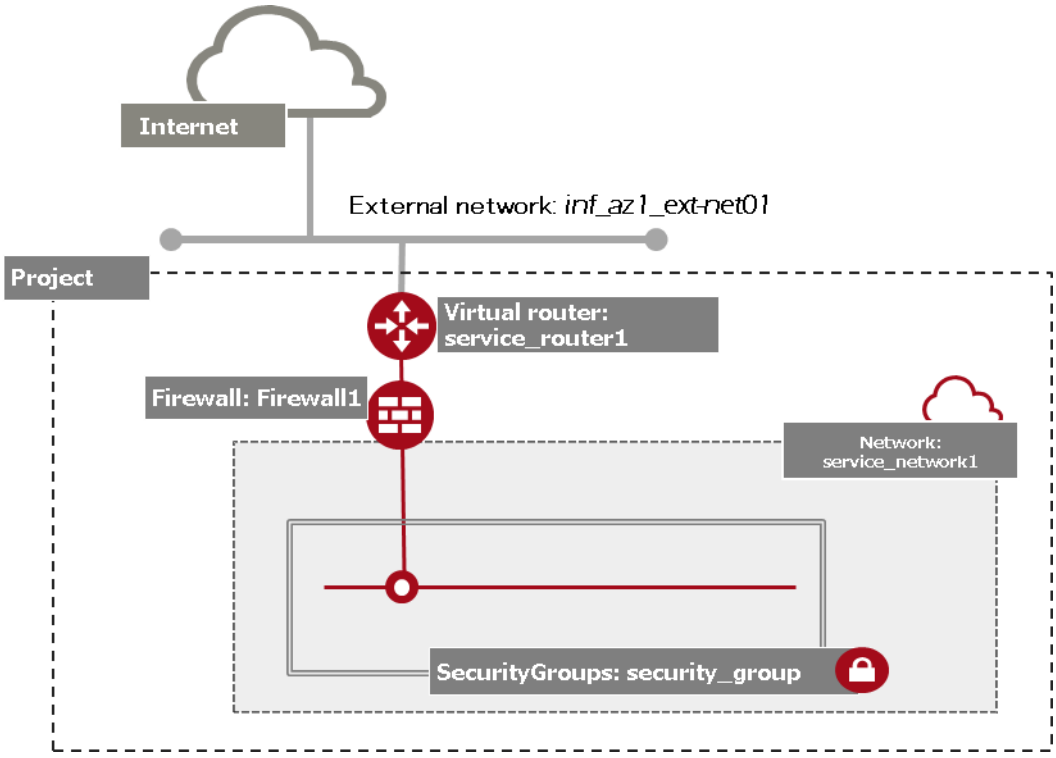
```
  autoscale_elb_name:
    type: string
    default: m0918WinELB2


resources:

  autoScaleWindows:
    type: FCX::AutoScaling::AutoScalingGroup
    properties:
      AvailabilityZones: [{get_param: az}]
      LaunchConfigurationName: {get_resource: launch_config}
      MinSize: '1'
      MaxSize: '2'
      VPCZoneIdentifier: {get_param: autoscale_subnet_id}
      HealthCheckGracePeriod: '110'
      HealthCheckType: 'ELB'
      Cooldown: 750
      LoadBalancerNames: [{get_resource: fj_elb}]
      Tags: [{"Key": "admin_pass", "Value": "(Arbitrary password (*1))"}]
        *1: Specify a password that satisfies the complexity
          requirements of Windows.

  launch_config:
    type: FCX::AutoScaling::LaunchConfiguration
    properties:
      ImageId: { get_param: param_image_id }
      InstanceType: { get_param: param_flavor }
      SecurityGroups: {get_param: autoscale_security_group_name}
      BlockDeviceMappingsV2: [{source_type: 'image', destination_type: 'volume',
 boot_index: '0', device_name: '/dev/vda', volume_size: '80', uuid: {get_param:
 param_image_id}, delete_on_termination: true, volume_type: 'F2'}]
      UserData: |
        #ps1
        New-Item "c:¥¥test" -itemType Directory

  fj_elb:
    type: FCX::ExpandableLoadBalancer::LoadBalancer
    properties:
      Subnets: {get_param: autoscale_subnet_id}
      SecurityGroups: {get_param: autoscale_security_group_id}
      Listeners:
      - {LoadBalancerPort: '80', InstancePort: '80',
        Protocol: 'HTTP', InstanceProtocol: 'HTTP' }
      HealthCheck: {Target: 'HTTP:80/iisstart.htm', HealthyThreshold: '3',
                    UnhealthyThreshold: '5', Interval: '30', Timeout: '5'}
      Version: 2014-09-30
      Scheme: internal
      LoadBalancerName: {get_param: autoscale_elb_name}

  vm_recover_policy:
    type: FCX::AutoScaling::ScalingPolicy
    properties:
      AdjustmentType: ChangeInCapacity
      AutoScalingGroupName: {get_resource: autoscalewindows}
      ScalingAdjustment: '1'

  elb_status_abnormal:
    type: OS::Ceilometer::Alarm
    properties:
      description: elb_unhealty_recovery
      meter_name: fcx.loadbalancing.instance.unhealthy
      statistic: min
      period: '180'
      evaluation_periods: '1'
      repeat_actions: true
      threshold: '1'
      alarm_actions:
```

```
      - {get_attr: [vm_recover_policy, AlarmUrl]}
      matching_metadata: {'resource_id': {get_param: autoscale_elb_name}}
      comparison_operator: ge
```

6.7 Example network creation Heat template

hello_world_Network_1.yaml, hello_world_Network_2.yaml

The following is an example template that defines a network with the structure shown below.



CAUTION

It is not possible to create a stack using a template in which a router that will connect to an external network is defined. It is necessary to first create a stack using a template in which a router that will not connect to external networks is defined, and then update the stack by adding the property "external_gateway_info" to the router definition in the template.

To enable connections to external networks, use the file "hello_world_Network_1.yaml" described in this section to create the stack, and use "hello_world_Network_2.yaml" to update it.

An error will also occur when creating a stack using a Heat template that includes "load balancer service" or "database service" resources that can communicate with external networks. However, it is possible to successfully deploy these resources by performing stack updates similar to the one described above.

hello_world_Network_1.yaml

```
heat_template_version: 2013-05-23
description: Network part for service.
parameters:
  az:
    type: string
    description: Availability zone
    default: (The availability zone to be used. Example: jp-east-1a)
  service_network1_name:
    type: string
    description: Name of the service network
    default: (Arbitrary network name)
  service_subnet1_name:
```

```
    type: string
    description: Name of the service subnetwork.
    default: (Arbitrary subnet name)
  service_router1_name:
    type: string
    description: Name of the service vrouter.
    default: (Arbitrary router name)
  service_subnet1_cidr:
    type: string
    description: CIDR representation of the service subnet.
    default: (Arbitrary CIDR)
  service_subnet1_gw_ip:
    type: string
    description: Gateway IP of Subnet
    default: (The gateway address to be configured for the subnet)
  service_subnet1_gw_port_name:
    type: string
    description: Gateway port name of Subnet
    default: (Arbitrary port name)
  security_group_name:
    type: string
    description: Security Group name
    default: (Arbitrary security group name)
  nameserver_ip1:
    type: string
    description: IP of the dns nameserver1.
    default: (The IP address of DNS server 1 to be used by servers deployed to the subnet)
  nameserver_ip2:
    type: string
    description: IP of the dns nameserver2.
    default: (The IP address of DNS server 2 to be used by servers deployed to the subnet)
  firewall1_name:
    type: string
    description: Name of the firewall1
    default: (Arbitrary firewall name)
  firewall1_policy_name:
    type: string
    description: Name of the firewall1 Policy
    default: (Arbitrary firewall policy name)

resources:
  service_network1:
    type: OS::Neutron::Net
    properties:
      availability_zone: {get_param: az }
      name: { get_param: service_network1_name }
  service_subnet1:
    type: OS::Neutron::Subnet
    properties:
      availability_zone: { get_param: az }
      cidr: { get_param: service_subnet1_cidr }
      name: { get_param: service_subnet1_name }
      gateway_ip: { get_param: service_subnet1_gw_ip }
      network_id: { get_resource: service_network1 }
      dns_nameservers: [{ get_param: nameserver_ip1 }, { get_param: nameserver_ip2 }]
  gw_port1:
    type: OS::Neutron::Port
    properties:
      availability_zone: { get_param: az }
      network_id: { get_resource: service_network1 }
      fixed_ips: [{"ip_address": {get_param: service_subnet1_gw_ip }, "subnet_id":
 {get_resource: service_subnet1 }}]
      name: { get_param: service_subnet1_gw_port_name }
  service_router1:
    type: OS::Neutron::Router
    properties:
      availability_zone: { get_param: az }
      name: { get_param: service_router1_name }
```

```
  service_router_interface1:
    depends_on: service_router1
    type: OS::Neutron::RouterInterface
    properties:
      router_id: { get_resource: service_router1 }
      port_id: { get_resource: gw_port1 }
  security_group:
    type: OS::Neutron::SecurityGroup
    properties:
      description: test Security groups rule
      name: { get_param: security_group_name }
      availability_zone: {get_param: az}
      rules: [{"direction": ingress, "port_range_max": 22, "port_range_min": 22,
 "protocol": tcp, "remote_ip_prefix": 192.168.0.0/16 },
              {"direction": ingress, "protocol": icmp, "remote_ip_prefix":
 192.168.0.0/16 }]
  firewall1:
    type: OS::Neutron::Firewall
    properties:
      description: test Firewall
      name: { get_param: firewall1_name }
      availability_zone: {get_param: az }
      firewall_policy_id: {get_resource: firewall1_policiy }
# For the Japan East1 / Japan West1 / Japan West2 regions, remove the comment out "#" from
# the following comment.
#      router_id: {get_resource: service_router1 }
# For the Japan East2 regions, remove the comment out "#" from the following comment.
#      value_specs: {"router_ids": [{get_resource: service_router1}]}
  firewall1_policiy:
    type: OS::Neutron::FirewallPolicy
    properties:
      audited: true
      description: test Firewall Policy
      firewall_rules: [{ get_resource: firewall_rule1 },{ get_resource: firewall_rule2 }]
      name: { get_param: firewall1_policy_name }
      availability_zone: {get_param: az }
  firewall_rule1:
    type: OS::Neutron::FirewallRule
    properties:
      description: test Firewall rule
      destination_port: "80"
      protocol: tcp
      source_ip_address: {get_param: service_subnet1_cidr}
      availability_zone: {get_param: az }
      action: allow
  firewall_rule2:
    type: OS::Neutron::FirewallRule
    depends_on: firewall_rule1
    properties:
      description: test Firewall rule2
      source_port: "53"
      protocol: udp
      source_ip_address: {get_param: service_subnet1_cidr}
      destination_ip_address: {get_param: nameserver_ip1}
      availability_zone: {get_param: az}
      action: allow
```

hello_world_Network_2.yaml

```
heat_template_version: 2013-05-23
description: Network part for service.
parameters:
  az:
    type: string
    description: Availability zone
    default: (The availability zone to be used. Example: jp-east-1a)
  service_network1_name:
    type: string
    description: Name of the service network
```

```
    default: (Arbitrary network name)
  service_subnet1_name:
    type: string
    description: Name of the service subnetwork.
    default: (Arbitrary subnet name)
  service_router1_name:
    type: string
    description: Name of the service vrouter.
    default: (Arbitrary router name)
  service_subnet1_cidr:
    type: string
    description: CIDR representation of the service subnet.
    default: (Arbitrary CIDR)
  service_subnet1_gw_ip:
    type: string
    description: Gateway IP of Subnet
    default: (The gateway address to be configured for the subnet)
  service_subnet1_gw_port_name:
    type: string
    description: Gateway port name of Subnet
    default: (Arbitrary port name)
  security_group_name:
    type: string
    description: Security Group name
    default: (Arbitrary security group name)
  nameserver_ip1:
    type: string
    description: IP of the dns nameserver1.
    default: (The DNS server to be used by servers deployed to the subnet)
  nameserver_ip2:
    type: string
    description: IP of the dns nameserver2.
    default: (The DNS server to be used by servers deployed to the subnet)
  firewall1_name:
    type: string
    description: Name of the firewall1
    default: (Arbitrary firewall name)
  firewall1_policy_name:
    type: string
    description: Name of the firewall1 Policy
    default: (Arbitrary firewall policy name)
  ext-net:
    type: string
    default:  (The name of the external network to be used. Example:inf_az1_ext-net01)

resources:
  service_network1:
    type: OS::Neutron::Net
    properties:
      availability_zone: {get_param: az }
      name: { get_param: service_network1_name }
  service_subnet1:
    type: OS::Neutron::Subnet
    properties:
      availability_zone: { get_param: az }
      cidr: { get_param: service_subnet1_cidr }
      name: { get_param: service_subnet1_name }
      gateway_ip: { get_param: service_subnet1_gw_ip }
      network_id: { get_resource: service_network1 }
      dns_nameservers: [{ get_param: nameserver_ip1 }, { get_param: nameserver_ip2 }]
  gw_port1:
    type: OS::Neutron::Port
    properties:
      availability_zone: { get_param: az }
      network_id: { get_resource: service_network1 }
      fixed_ips: [{"ip_address": {get_param: service_subnet1_gw_ip }, "subnet_id":
 {get_resource: service_subnet1 }}]
      name: { get_param: service_subnet1_gw_port_name }
```

```
  service_router1:
    type: OS::Neutron::Router
    properties:
# Update the stack by adding the following line to enable
# the router to connect to external networks.
      external_gateway_info: {"network": {get_param: ext-net} }
      availability_zone: { get_param: az }
      name: { get_param: service_router1_name }
  service_router_interface1:
    depends_on: service_router1
    type: OS::Neutron::RouterInterface
    properties:
      router_id: { get_resource: service_router1 }
      port_id: { get_resource: gw_port1 }
  security_group:
    type: OS::Neutron::SecurityGroup
    properties:
      description: test Security groups rule
      name: { get_param: security_group_name }
      availability_zone: {get_param: az}
      rules: [{"direction": ingress, "port_range_max": 22, "port_range_min": 22,
 "protocol": tcp, "remote_ip_prefix": 192.168.0.0/16 },
              {"direction": ingress, "protocol": icmp, "remote_ip_prefix":
 192.168.0.0/16 }]
  firewall1:
    type: OS::Neutron::Firewall
    properties:
      description: test Firewall
      name: { get_param: firewall1_name }
      availability_zone: {get_param: az }
      firewall_policy_id: {get_resource: firewall1_policiy }
# For the Japan East1 / Japan West1 / Japan West2 regions, remove the comment out "#" from
# the following comment.
#      router_id: {get_resource: service_router1 }
# For the Japan East2 regions, remove the comment out "#" from the following comment.
value_specs: {"router_ids": [{get_resource: service_router1}]}
  firewall1_policiy:
    type: OS::Neutron::FirewallPolicy
    properties:
      audited: true
      description: test Firewall Policy
      firewall_rules: [{ get_resource: firewall_rule1 },{ get_resource: firewall_rule2 }]
      name: { get_param: firewall1_policy_name }
      availability_zone: {get_param: az }
  firewall_rule1:
    type: OS::Neutron::FirewallRule
    properties:
      description: test Firewall rule
      destination_port: "80"
      protocol: tcp
      source_ip_address: {get_param: service_subnet1_cidr}
      availability_zone: {get_param: az }
      action: allow
  firewall_rule2:
    type: OS::Neutron::FirewallRule
    depends_on: firewall_rule1
    properties:
      description: test Firewall rule2
      source_port: "53"
      protocol: udp
      source_ip_address: {get_param: service_subnet1_cidr}
      destination_ip_address: {get_param: nameserver_ip1}
      availability_zone: {get_param: az}
      action: allow
```

6.8 Example Security Group creation Heat template

Creating_a_security_group.yaml

This is an example of a template for creating Security Groups and the Security Group rules that are associated with them.

```
#
# This template creates Security Groups and the Security Group rules that are
# associated with them.
#
heat_template_version: 2013-05-23

description: >
  Creating a Security Group Sample template.

parameters:
  AZ:
    type: string
    description: Name of the Availability Zone
    default: jp-west-2a

  SG_NAME:
    type: string
    description: Name of the Security Group
    default: Sample_Security_Group_01

  REMOTE_IP:
    type: string
    description: Remote IP prefix to associate with the Security Group rule
    default: 192.168.10.0/24


resources:
  SECURITY_GROUP:
    type: OS::Neutron::SecurityGroup
    properties:
      availability_zone: { get_param: AZ }
      name: { get_param: SG_NAME }
      description: >
        Security group rule to create
        Allows connection from remote IP (icmp, tcp22, 80, 443, 3389, 1688, 53, and udp 53)
      rules:
         #1 ingress: ICMP/RemoteIP
         - { remote_ip_prefix: { get_param: REMOTE_IP }, direction: ingress, protocol:
 icmp }
         #2 ingress: TCP/SSH/22/RemoteIP
         - { remote_ip_prefix: { get_param: REMOTE_IP }, direction: ingress, protocol: tcp,
 port_range_min: 22, port_range_max: 22}
         #3 ingress: TCP/HTTP/80/RemoteIP
         - { remote_ip_prefix: { get_param: REMOTE_IP }, direction: ingress, protocol: tcp,
 port_range_min: 80, port_range_max: 80}
         #4 ingress: TCP/HTTPS/443/RemoteIP
         - { remote_ip_prefix: { get_param: REMOTE_IP }, direction: ingress, protocol: tcp,
 port_range_min: 443, port_range_max: 443}
         #5 ingress: TCP/RDP/3389/RemoteIP
         - { remote_ip_prefix: { get_param: REMOTE_IP }, direction: ingress, protocol: tcp,
 port_range_min: 3389, port_range_max: 3389}
         #6 ingress: TCP/KMS/1688/RemoteIP
         - { remote_ip_prefix: { get_param: REMOTE_IP }, direction: ingress, protocol: tcp,
 port_range_min: 1688, port_range_max: 1688}
         #7 ingress: TCP/DNS/53/RemoteIP
         - { remote_ip_prefix: { get_param: REMOTE_IP }, direction: ingress, protocol: tcp,
 port_range_min: 53, port_range_max: 53}
         #8 ingress: UDP/DNS/53/RemoteIP
```

```
        - { remote_ip_prefix: { get_param: REMOTE_IP }, direction: ingress, protocol: udp,
port_range_min: 53, port_range_max: 53}
```

6.9 Example Virtual Server creation with global IP assigned Heat template

Creating_a_virtual_server_with_floating_IP.yaml

This is an example of a template for creating a single Virtual Server with a global IP assigned.

```
#
# This template creates a single Virtual Server with a global IP assigned.
#
heat_template_version: 2013-05-23

description: >
  Create a Virtual Server with a Floating IP assigned.

parameters:
  KEY_NAME:
    type: string
    description: Name of the existing key pair used for the Virtual Server
    default: TEST_KEY_01

  FLAVOR_NAME:
    type: string
    description: Name or ID of the Flavor type of the Virtual Server to create
    default: S-1

  IMAGE_ID:
    type: string
    description: Image ID or image name to use for the Virtual Server (CentOS)
    default: c3867e5e-afd6-4858-918e-c445f9041c9d

  AZ:
    type: string
    description: Name of the Availability Zone to deploy the Virtual Server in
    default: jp-west-2a

  SERVER_NAME:
    type: string
    description: Name of the Virtual Server to create
    default: Sample_Server_01

  SECURITY_GROUP_ID:
    type: string
    description: ID of the Security Group associated with the Virtual Server
    default: 935ce03a-73d2-475a-a21e-dff8ecb5gtsd

  EXT_NETWORK_ID:
    type: string
    description: External Network ID assigned to the Floating IP
    default: b3ca0a26-abc5-46e8-82f2-3f27b5e7a12u

  PRIVATE_NETWORK_ID:
    type: string
    description: ID of the Private Network to deploy the Virtual Server on
    default: 913478d6-3259-4b8d-acs8-9bca0b54769n

  SUBNET_ID:
    type: string
    description: ID of the Subnet to deploy the Virtual Server on
    default: 13ye9b10-85mb-4791-aced-c3a8bddc675e

  PORT_IP_ADDRESS:
    type: string
```

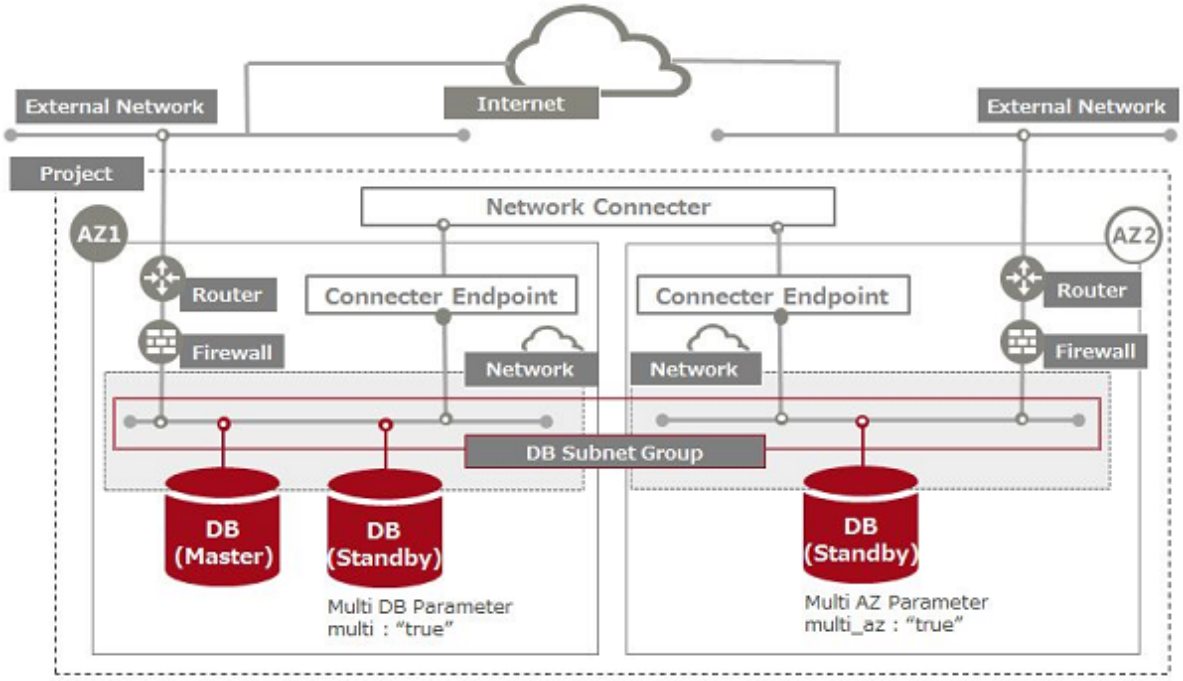
```
    description: IP address assigned to the Virtual Server
    default: 192.168.10.11

  VOLUME_NAME:
    type: string
    description: Name of the system volume of the Virtual Server
    default: Sample_Volume_01

  VOLUME_SIZE:
    type: string
    description: Volume size of the system volume of the Virtual Server
    default: 30

resources:
  SERVER_PORT:
    type: OS::Neutron::Port
    properties:
      admin_state_up: true
      network_id: { get_param: PRIVATE_NETWORK_ID }
      fixed_ips: [{"ip_address": { get_param: PORT_IP_ADDRESS }, "subnet_id": { get_param:
 SUBNET_ID }}]
      security_groups: [{ get_param: SECURITY_GROUP_ID }]
      availability_zone: { get_param: AZ }

  SERVER_FLOATING_IP:
    type: OS::Neutron::FloatingIP
    properties:
      fixed_ip_address: { get_param: PORT_IP_ADDRESS }
      floating_network_id: { get_param: EXT_NETWORK_ID }
      port_id: { get_resource: SERVER_PORT }
      availability_zone: { get_param: AZ }

  SYS-VOL:
    type: OS::Cinder::Volume
    properties:
      name: { get_param: VOLUME_NAME }
      size: { get_param: VOLUME_SIZE }
      volume_type: "M1"
      image : { get_param: IMAGE_ID }
      availability_zone: { get_param: AZ }

  SERVER:
    type: OS::Nova::Server
    properties:
      name: { get_param: SERVER_NAME }
      availability_zone: { get_param: AZ }
      block_device_mapping: [{"volume_size": { get_param: VOLUME_SIZE }, "volume_id":
 { get_resource: SYS-VOL }, "delete_on_termination": True, "device_name": "/dev/vda" }]
      flavor: { get_param: FLAVOR_NAME }
      image: { get_param: IMAGE_ID }
      key_name: { get_param: KEY_NAME }
      networks: [{"port": { get_resource: SERVER_PORT }}]


outputs:
  SERVER_PRIVATE_IP:
    description: IP address of server in private network
    value: { get_attr: [ SERVER, first_address ] }

  SERVER_EXT_IP:
    description: Floating IP address of server in public network
    value: { get_attr: [ SERVER_FLOATING_IP, floating_ip_address ] }

  SERVER_DETAILS:
    description: Shows details of all virtual servers.
    value: { get_attr: [ SERVER, show ] }
```

6.10 Example Database Virtual Server creation Heat template

Creating_a_database_virtual_server.yaml

This is an example of a template for creating a DB Subnet Group, a DB Parameter Group, and a Database Virtual Server.

It creates the following configuration.




```
#
# This template creates a DB Subnet Group, a DB Parameter Group, and a Database Virtual
 Server.
#
heat_template_version: 2013-05-23

description: >
  Create a database Virtual Server.


parameters:
  DB_SUBNET_GROUP_NAME:
    type: string
    description: Name of the DB Subnet Group
    default: SampleDbSubnetGroup01

  DB_SECURITY_GROUP_ID:
    type: string
    description: ID of the Security Group to deploy the Database Virtual Server in
    default: 935ce03a-73d2-475a-a21e-dff8ecb5gtsd

  DB_SUBNET_ID_1:
    type: string
    description: Subnet ID 1 to register in the DB Subnet Group
    default: 13ye9b10-85mb-4791-aced-c3a8bddc675e

  DB_SUBNET_ID_2:
    type: string
    description: Subnet ID 2 to register in the DB Subnet Group
    default: 196e758b-54ae-4f3c-9f00-ce8714dc072m
```

```
 DB_PARAMETER_GROUP_NAME:
   type: string
   description: Name of the DB Parameter Group
   default: SampleDbParameterGroup01

 PARAMETER_GROUP_FAMILY:
   type: string
   description: Type of Parameter Group decided based on the Database engine and version
   default: enterprisepostgres_v9.6

 DB_INSTANCE_NAME:
   type: string
   description: Name of the Database Virtual Server
   default: SampleDbInstance01

 AZ:
   type: string
   description: Name of the Availability Zone
   default: jp-west-2a

 DB_VOLUME_SIZE:
   type: string
   description: Volume size of the Database Virtual Server
   default: "30"

 DB_STRUCTURE_NAME:
   type: string
   description: Database name
   default: database01

 DB_MASTER_USER_NAME:
   type: string
   description: Database administrator user name
   default: root

 DB_MASTER_PASSWORD:
   type: string
   description: Database administrator user password
   hidden: true
   default: rootroot

 DB_USER_NAME:
   type: string
   description: Database system user name
   hidden: true
   default: user01

 DB_USER_PASSWORD:
   type: string
   description: Database system user password
   hidden: true
   default: user01passwd

 DB_PORT:
   type: string
   description: Number of the port used by the Database
   default: 5432

 MULTI_FLAG:
   type: string
   description: Specify "true" or "false" for the Multi-DB setting (Redundancy inside the
same Availability Zone)
   default: "true"

 MULTI_AZ_FLAG:
   type: string
   description: Specify "true" or "false" for the Multi-AZ setting (Redundancy inside a
different Availability Zone)
```

```
    default: "false"

  FLAVOR_ID:
    type: string
    description: Flavor ID of the Database Virtual Server
    default: "1101"

resources:
  DB_SUBNET_GROUP:
    type: FCX::Database::DBSubnetGroup
    properties:
      name: { get_param: DB_SUBNET_GROUP_NAME }
      subnet_ids: [{"subnet_id": { get_param: DB_SUBNET_ID_1 }}, { "subnet_id":
 { get_param: DB_SUBNET_ID_2 }}]

  DB_PARAMETER_GROUP:
    type: FCX::Database::DBParameterGroup
    properties:
      name: { get_param: DB_PARAMETER_GROUP_NAME }
      parameter_group_family: { get_param: PARAMETER_GROUP_FAMILY }

  DB_INSTANCE:
    type: FCX::Database::DBInstance
    properties:
      name: {get_param: DB_INSTANCE_NAME }
      flavor: { get_param: FLAVOR_ID }
      size: { get_param: DB_VOLUME_SIZE }
      disk_type: "M1"
      availability_zone: { get_param: AZ }
      subnet_group_id: { get_resource: DB_SUBNET_GROUP }
      multi: { get_param: MULTI_FLAG }
      multi_az: { get_param: MULTI_AZ_FLAG }
      port: { get_param: DB_PORT }
      publicly_accessible: false
      security_group_ids: [{get_param: DB_SECURITY_GROUP_ID}]
      parameter_group_id: { get_resource: DB_PARAMETER_GROUP }
      backup_retention_period: 0
      masteruser_name: { get_param: DB_MASTER_USER_NAME }
      masteruser_password: { get_param: DB_MASTER_PASSWORD }
      databases: ["name": { get_param: DB_STRUCTURE_NAME} ]
      users: [{ "name": { get_param: DB_USER_NAME }, "password": { get_param:
 DB_USER_PASSWORD }, databases: [{ get_param: DB_STRUCTURE_NAME}] }]

outputs:
  db_instance_internal_fqdn:
    description: db_instance internal fqdn
    value: { get_attr: [DB_INSTANCE, PRIVATEADDRESS] }

  db_instance_internal_ip:
    description: db_instance internal ip
    value: { get_attr: [DB_INSTANCE, PRIVATEIP] }

  db_instance_internal_sub_ip:
    description: db_instance slave ip
    value: { get_attr: [DB_INSTANCE, SUBPRIVATEIP] }
```

A: Resource type properties

A.1 Preface

Refer to "Resource type details" for details on the resource types.

- About the "Mandatory" column:
  - Y: Item that must be specified
  - (blank): Item that can be set if desired (Optional)
- About the "Updateable" column:
  - Y: Can be updated using "Update stack"
  - Y*: Can be only be updated using "Update stack" in For the Japan East2 regions
  - (blank): Resources are created/deleted when you attempt to change a resource type using "Update stack"

Notice

*1: This parameter can only be used in For the Japan East2 regions

For details on "Update Stack", refer to the"API Reference - Application Platform Service".

A.2 Auto scaling

Resource Types	Properties	Mandatory	Updateable
FCX::AutoScaling::AutoScalingGroup	AvailabilityZones	Y	
	Cooldown		Y
	HealthCheckGracePeriod		
	HealthCheckType		
	LaunchConfigurationName	Y	Y
	LoadBalancerNames		
	MaxSize	Y	Y
	MinSize	Y	Y
	Tags		
	VPCZoneIdentifier	Y	
FCX::AutoScaling::LaunchConfiguration	BlockDeviceMappingsV2		
	ImageId	Y	
	InstanceType	Y	
	KeyName		
	NovaSchedulerHints		
	SecurityGroups		
	UserData		

Resource Types	Properties	Mandatory	Updateable
FCX::AutoScaling::ScalingPolicy	AdjustmentType	Y	Y
	AutoScalingGroupName	Y	
	Cooldown		Y
	ScalingAdjustment	Y	Y

A.3 Telemetry

Resource Types	Properties	Mandatory	Updateable
OS::Ceilometer::Alarm	meter_name	Y	
	alarm_actions		Y
	ok_actions		Y
	description		Y
	matching_metadata		
	evaluation_periods		Y
	statistic		Y
	enabled		Y
	period		Y
	insufficient_data_actions		Y
	repeat_actions		Y
	threshold	Y	Y
	comparison_operator		Y
OS::Ceilometer::CombinationAlarm	alarm_actions		Y
	ok_actions		Y
	description		Y
	enabled		Y
	alarm_ids	Y	Y
	insufficient_data_actions		Y
	repeat_actions		Y
	operator		Y

A.4 Block Storage

Resource Types	Properties	Mandatory	Updateable
OS::Cinder::Volume	availability_zone		
	backup_id		
	description		
	image		
	metadata		
	name		
	size		
	snapshot_id		
	source_volid		
	imageRef		
	volume_type		
OS::Cinder::VolumeAttachment	instance_uuid	Y	Y
	mountpoint		Y
	volume_id	Y	Y

A.5 Compute

Resource Types	Properties	Mandatory	Updateable
OS::Nova::Server	availability_zone		
	block_device_mapping		
	diskConfig		
	flavor	Y	Y
	image		
	key_name		
	metadata		Y
	name		Y
	networks		Y
	scheduler_hints		
	security_groups		
	user_data		
	user_data_format		
	admin_user		

A.6 Network

Resource Types	Properties	Mandatory	Updateable
OS::Neutron::Firewall	admin_state_up		Y
	description		Y
	firewall_policy_id	Y	Y
	name		Y
	value_specs (*1)		Y*
	router_ids (*1)		Y*
	router_id		Y*
	availability_zone		
OS::Neutron::FirewallPolicy	audited		Y
	description		Y
	firewall_rules	Y	Y
	name		Y
	availability_zone		
OS::Neutron::FirewallRule	action		Y
	description		Y
	destination_ip_address		Y
	destination_port		Y
	enabled		Y
	ip_version		Y
	name		Y
	protocol		Y
	source_ip_address		Y
	source_port		Y
	availability_zone		
OS::Neutron::FloatingIP	fixed_ip_address		Y*
	port_id		Y*
	floating_network_id	Y	
	availability_zone		
OS::Neutron::FloatingIPAssociation	floatingip_id	Y	Y
	port_id	Y	Y
	fixed_ip_address		Y

Resource Types	Properties	Mandatory	Updateable
OS::Neutron::Net	admin_state_up		Y
	name		Y
	tenant_id		
	availability_zone		
OS::Neutron::Port	admin_state_up		Y
	allowed_address_pairs		
	ip_address	Y	
	mac_address		
	fixed_ips		Y
	ip_address		
	subnet_id		
	mac_address		
	name		Y
	security_groups		Y
	network_id	Y	
	availability_zone		
OS::Neutron::Router	admin_state_up		Y
	external_gateway_info		Y
	network	Y	Y
	name		Y
	availability_zone		
OS::Neutron::RouterInterface	port_id		
	router_id	Y	
	subnet_id		
OS::Neutron::SecurityGroup	description		Y
	name		Y
	availability_zone		
	rules		Y
	direction		
	ethertype		
	port_range_max		
	port_range_min		
	protocol		
	remote_group_id		

Resource Types	Properties	Mandatory	Updateable
	remote_ip_prefix		
	remote_mode		
	availability_zone		
OS::Neutron::Subnet	allocation_pools		
	end	Y	
	start	Y	
	cidr	Y	
	dns_nameserver		Y
	enable_dhcp		Y
	gateway_ip		Y
	host_routes		Y*
	destination	Y	Y*
	nexthop	Y	Y*
	ip_version		
	name		Y
	tenant_id		
	network_id	Y	
	availability_zone		
FCX::Neutron::NetworkConnector	name		Y
	network_connector_pool_id		
	tenant_id		
FCX::Neutron::NetworkConnectorEndpoint	name		Y
	network_connector_id	Y	
	endpoint_type	Y	
	location	Y	
	tenant_id		
FCX::Neutron::NetworkConnectorEndpointConnection	network_connector_endpoint_id	Y	
	port_id	Y	

A.7 Expandable Load Balancing

Resource Types	Properties	Mandatory	Updateable
FCX::ExpandableLoadBalancer::LoadBalancer	LBCookieStickinessPolicies		Y
	Subnets	Y	Y
	LoadBalancerAttributes		Y
	Grade		
	HealthCheck		Y
	SorryServerRedirectionPolicies		Y
	OtherPolicies		Y
	Instances		Y
	Listeners	Y	Y
	Version	Y	
	SecurityGroups		Y
	LoadBalancerName	Y	
	Scheme		
	InstancesPorts		Y
	ListenersPolicies		Y

A.8 Database

Resource Types	Properties	Mandatory	Updateable
FCX::Database::DBInstance	backup_retention_period		Y
	availability_zone	Y	
	publicly_accessible		
	auto_minor_version_upgrade		Y
	flavor	Y	Y
	id		
	users		
	disk_type		Y
	port		Y
	collate		
	masteruser_password	Y	Y
	preferred_backup_window		Y
	size	Y	Y

Resource Types	Properties	Mandatory	Updateable
	engine		
	description		Y
	multi		Y
	security_group_ids		Y
	masteruser_name		
	parameter_group_id		Y
	subnet_group_id	Y	
	name		Y
	engine_version		Y
	multi_az		Y
	databases		
	character_set		
	preferred_maintenance_window		Y
	auto_maintenance		Y
FCX::Database::DBSubnetGroup	id		
	name	Y	
	subnet_ids	Y	
	description		
FCX::Database::DBParameterGroup	id		
	name	Y	
	parameter_group_family	Y	
	description		

A.9 Object Storage

Resource Types	Properties	Mandatory	Updateable
OS::Swift::Container	X-Container-Meta		Y
	X-Container-Read		Y
	name		
	X-Account-Meta		Y
	X-Container-Write		Y

B: Creating stacks using APIs

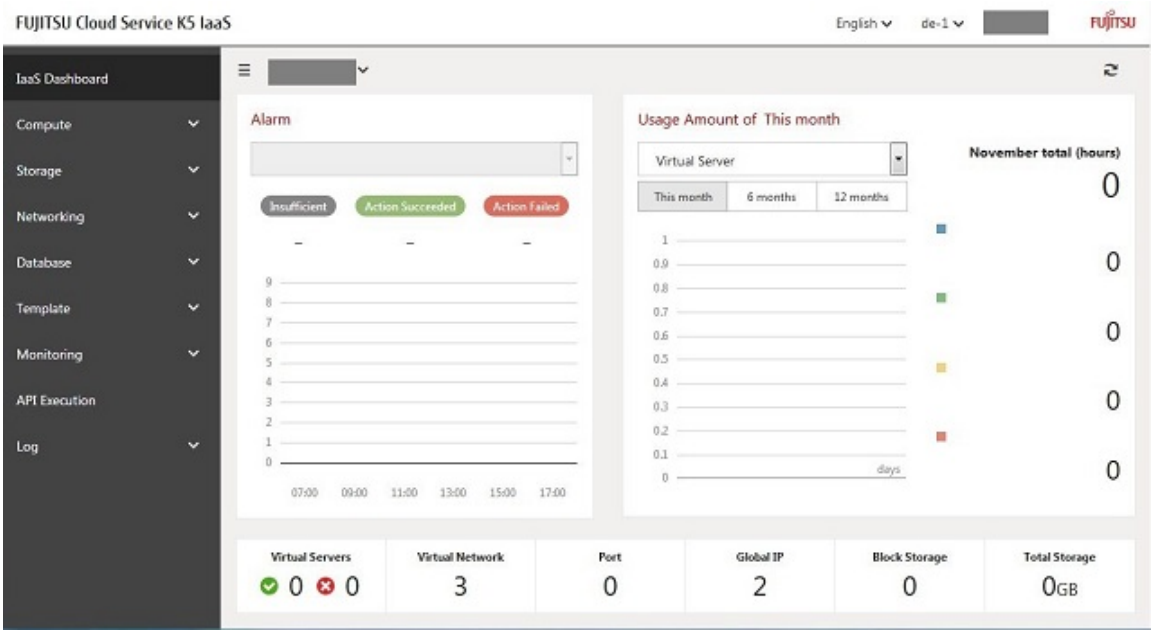
B.1 Preface

Here we explain the procedure for creating a stack using the API execution function of the IaaS service portal.
The procedure is as follows.

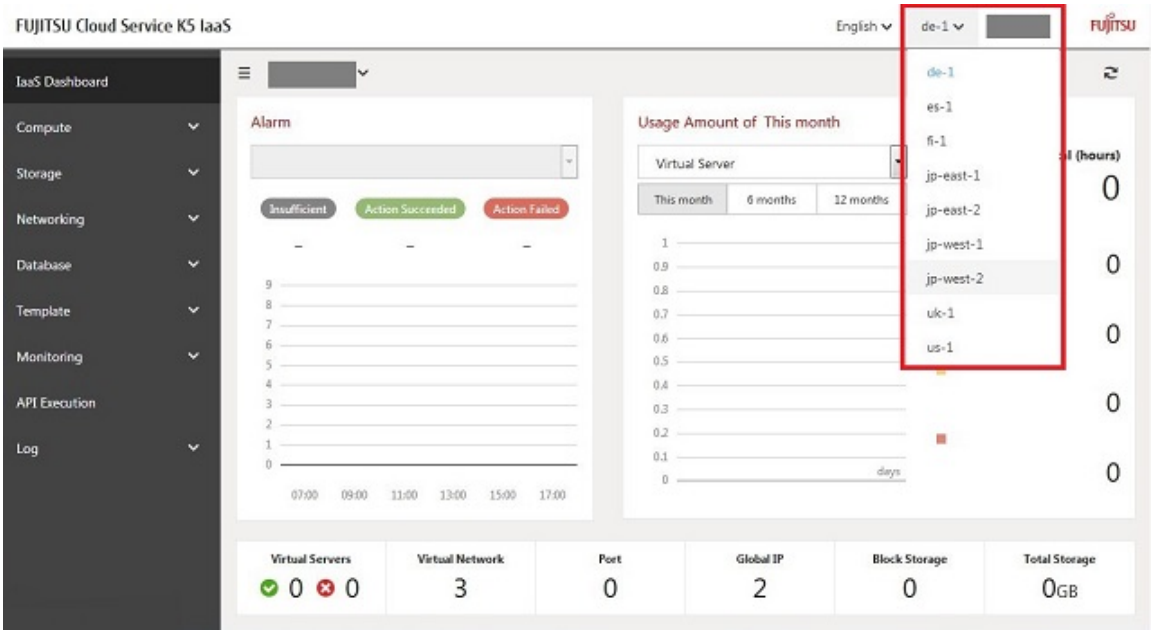
B.2 Request creation procedure

Procedure

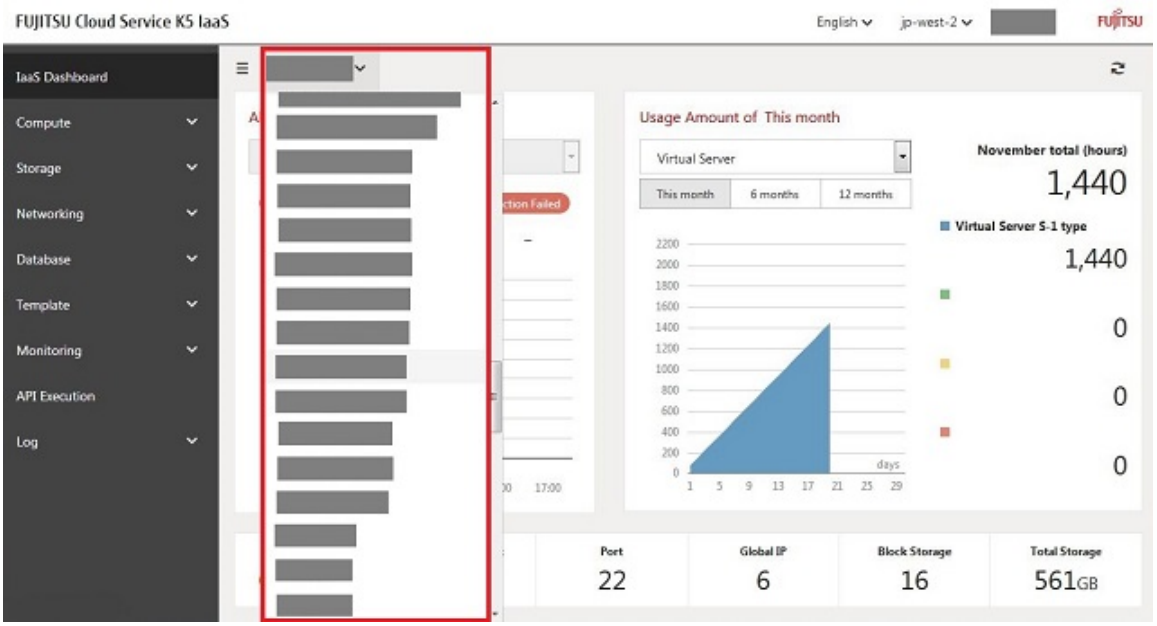
1. After logging in to the Portal, selecting "IaaS Portal" from the "Management" tab displays the "IaaS Dashboard" as below.



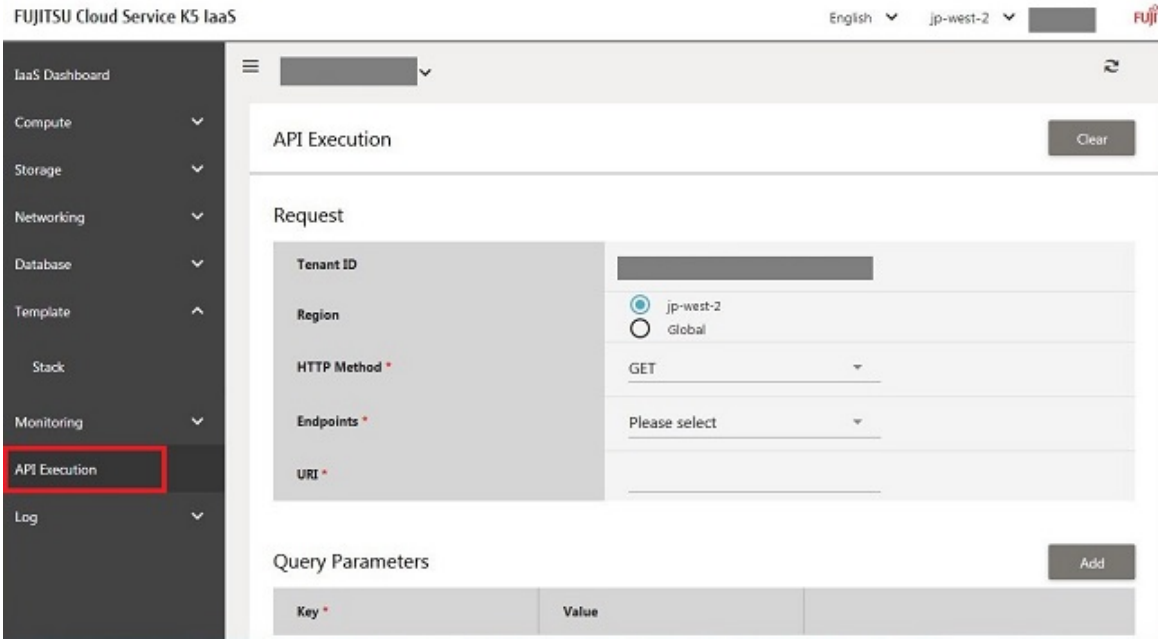
2. Select the region you want to create the stack in.



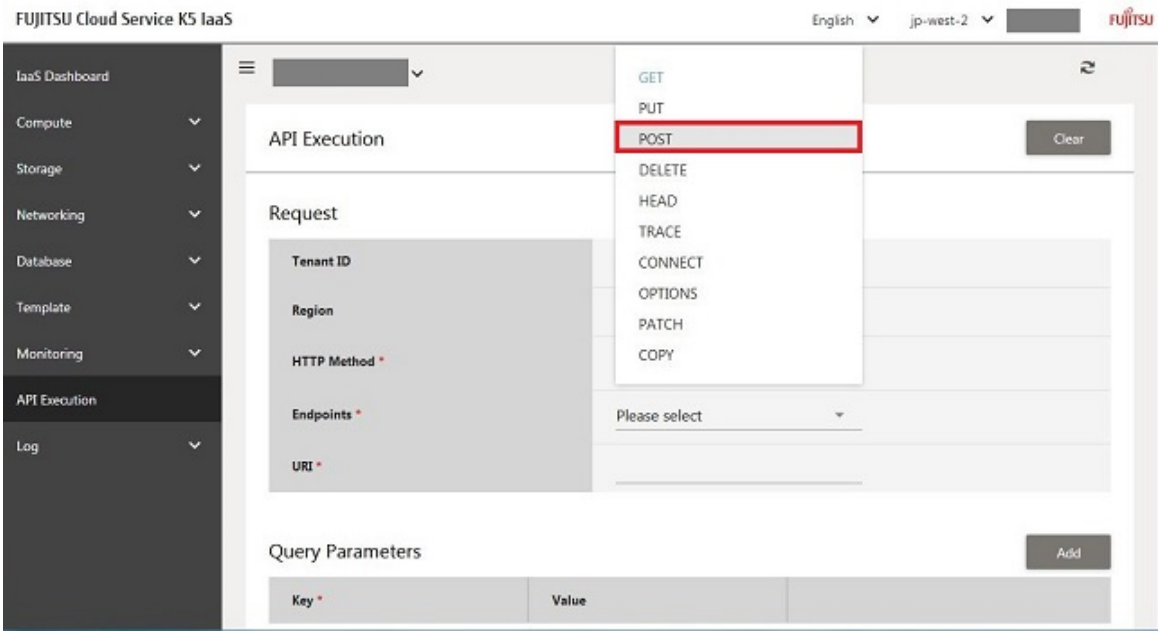
3. Select the project you want to create the stack in.



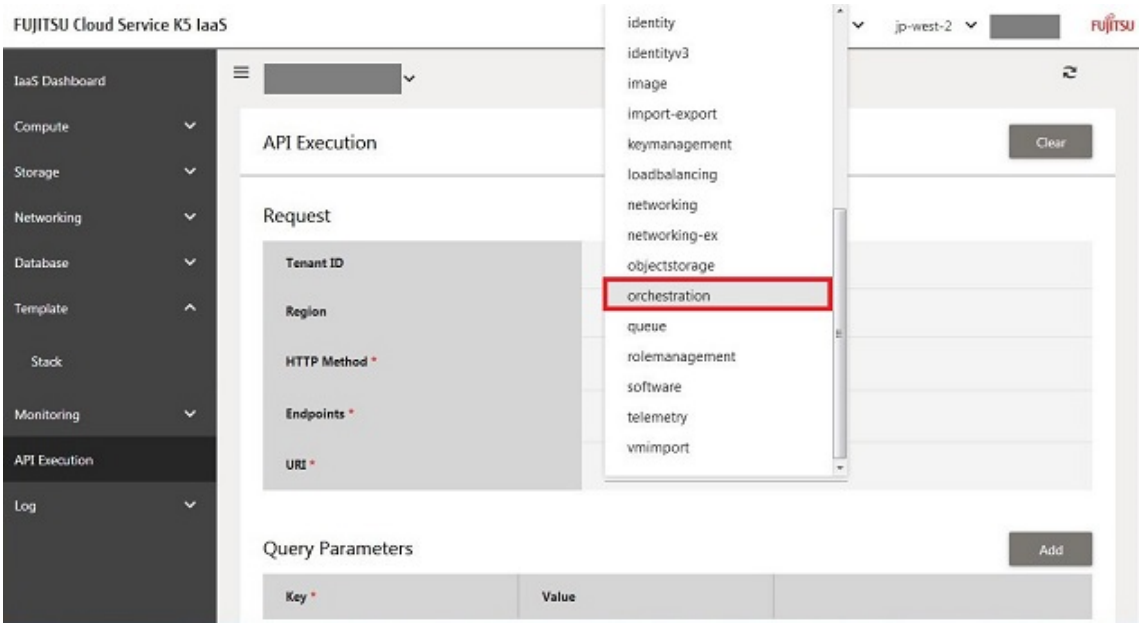
4. Clicking "API Execution" in the menu on the left displays the "API Execution" window.



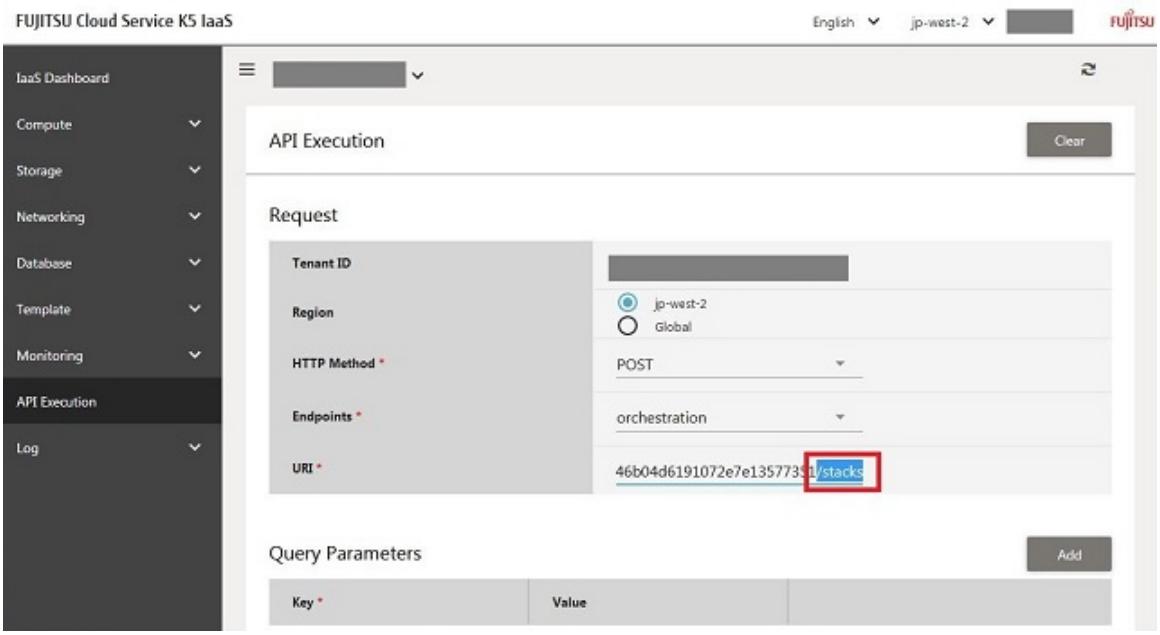
5. Select "POST" from the HTTP methods.



6. Select "orchestration" from the endpoints.



7. Add "/stacks" to the end of the displayed URI.

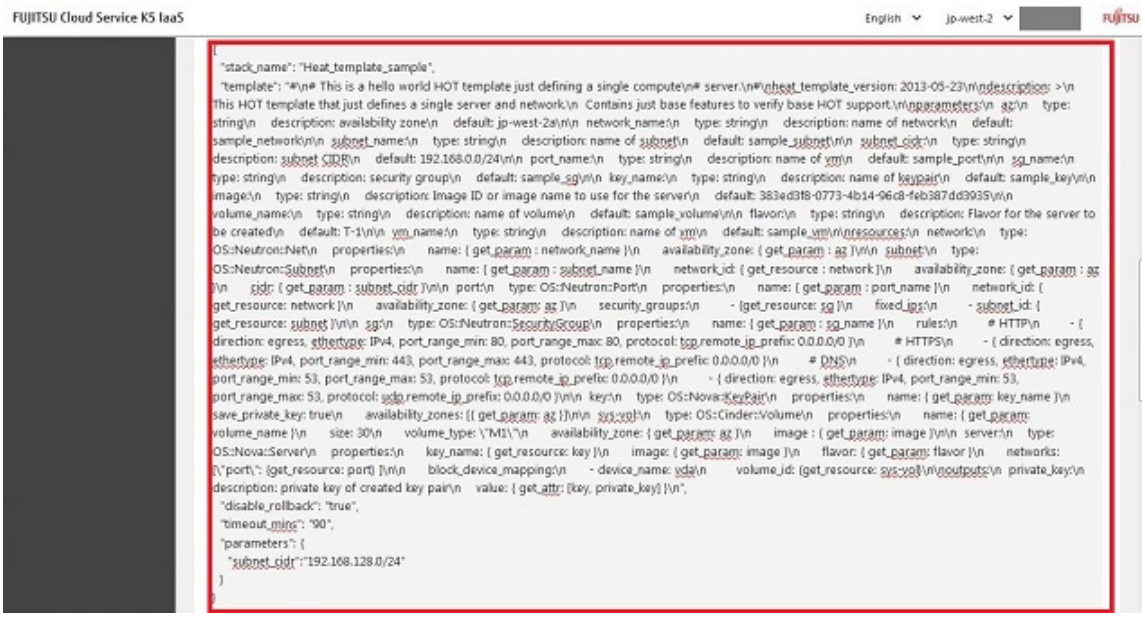


8. Input the JSON string in the request body, using the format below.

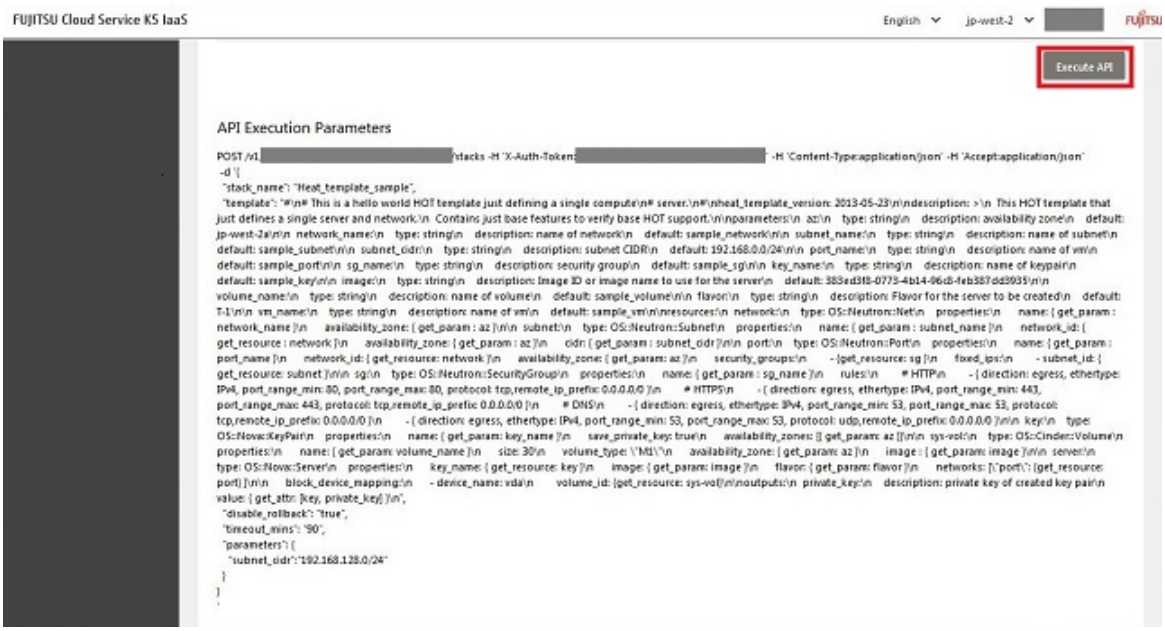
```
{
 "stack_name": "<stack name> ",
 "template": "<template text value>",
 "template_url": "<template file url >",
  "parameters": {
    "<param_name-n>":"<param_value-n>",
...
  }
}
```

*There are other options for the API, but they are not introduced here. For details, refer to the "API Reference - Application Platform Service".

The following is an example of an actual input window. The content input in "template" is an escaped version of the content in "Sample system configuration - Example Heat template".



9. Click "Execute API".



When "Response": 201 is returned in the Response field, the creation request was successful.

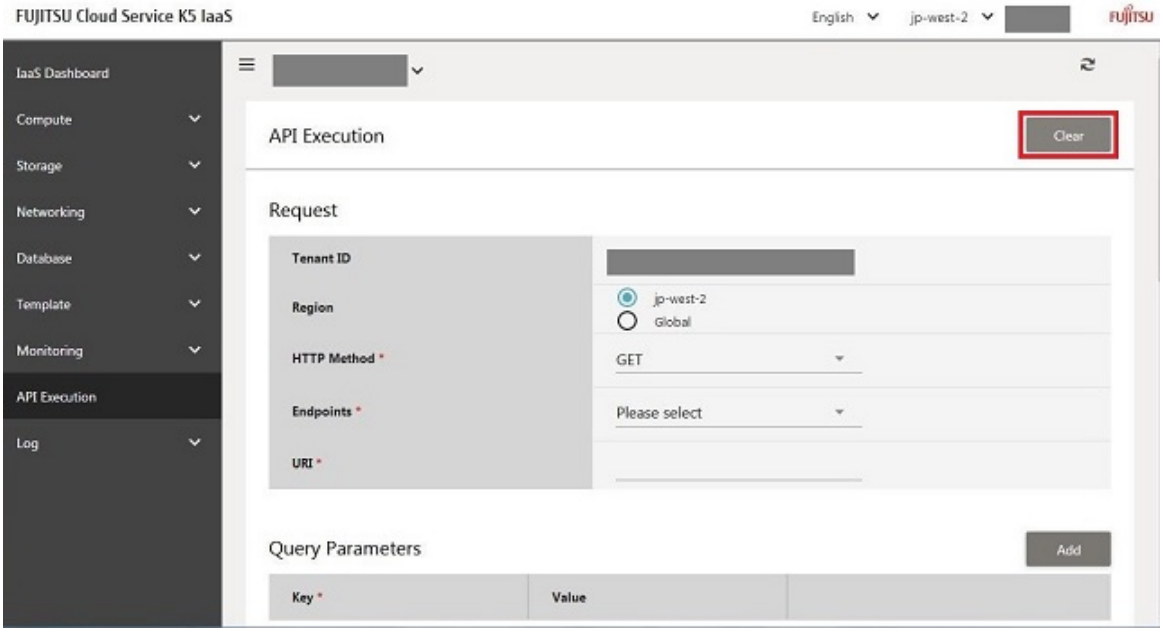
10.As confirmation is necessary later on, make a note of the stack ID in the response.

B.3 Request confirmation procedure

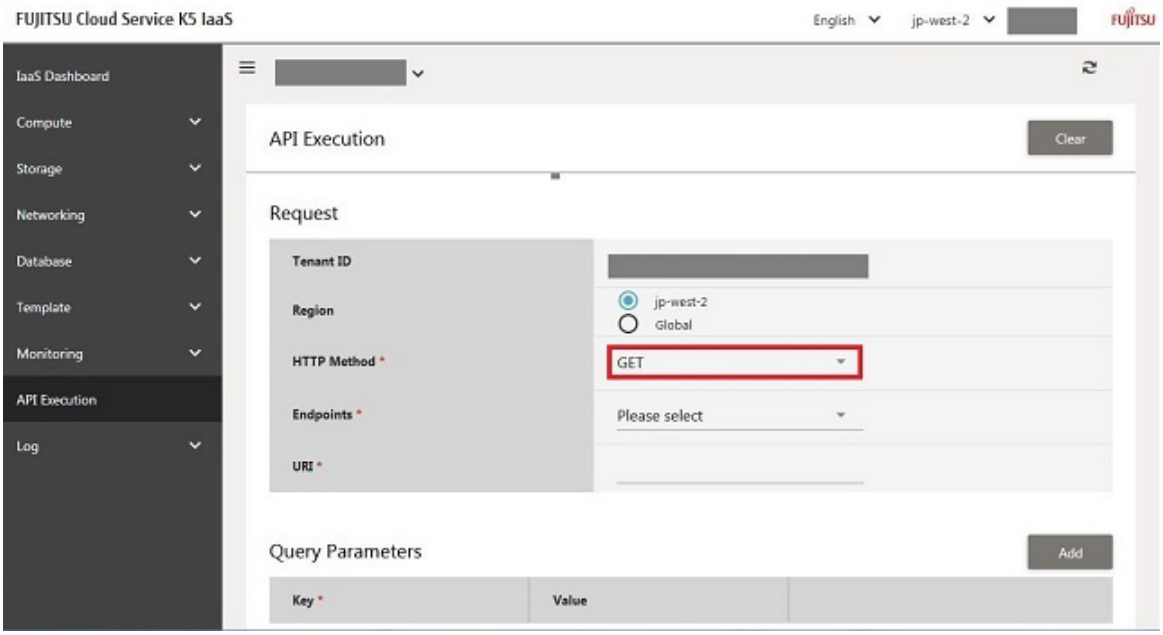
Stack creation is not completed at the timing of the creation request. Therefore it is necessary to confirm the status of creation of the stack using the following procedure.

Procedure

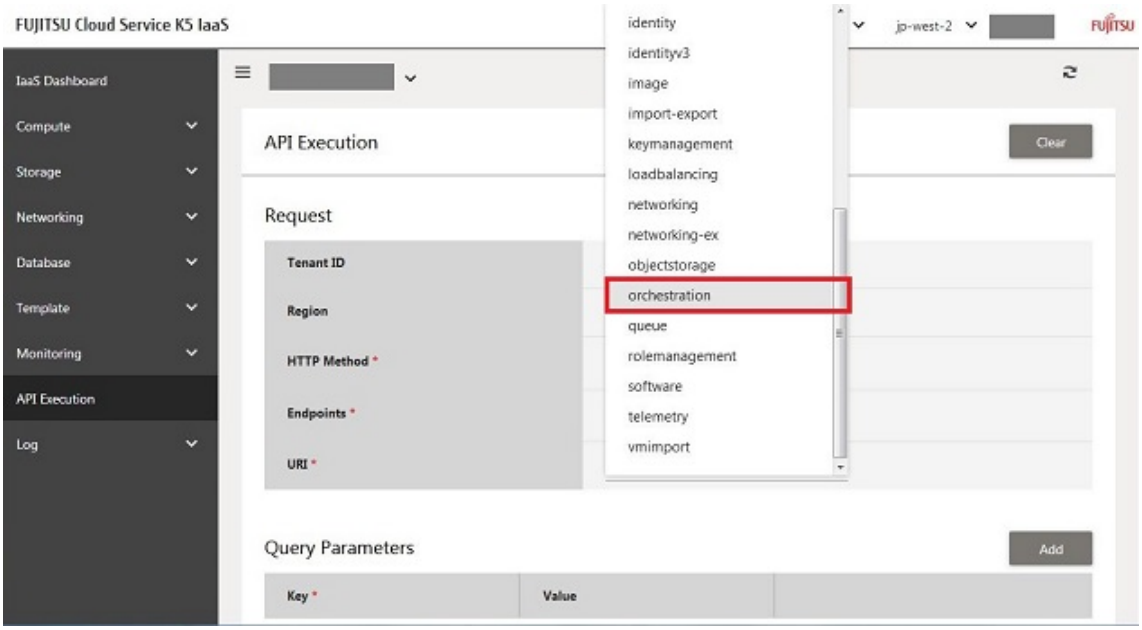
1. Click "Clear" on the upper part of the window to purge the currently displayed content.



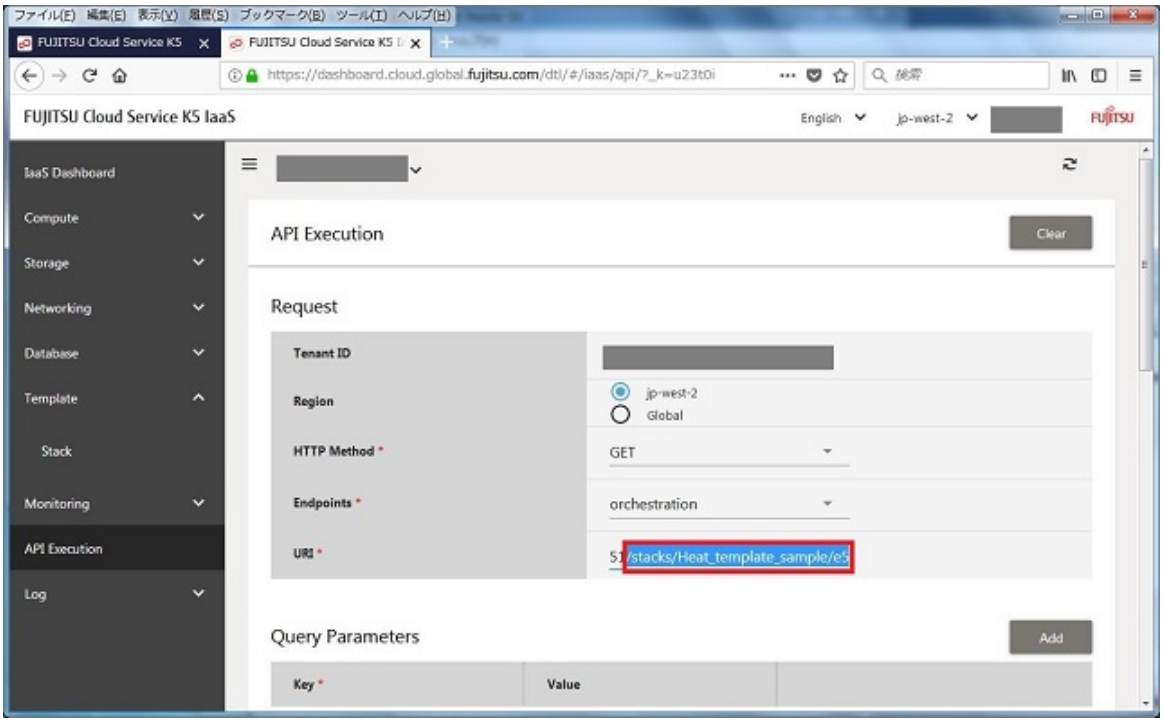
2. Confirm that the HTTP method is "GET".



3. Select "orchestration" from the endpoints.

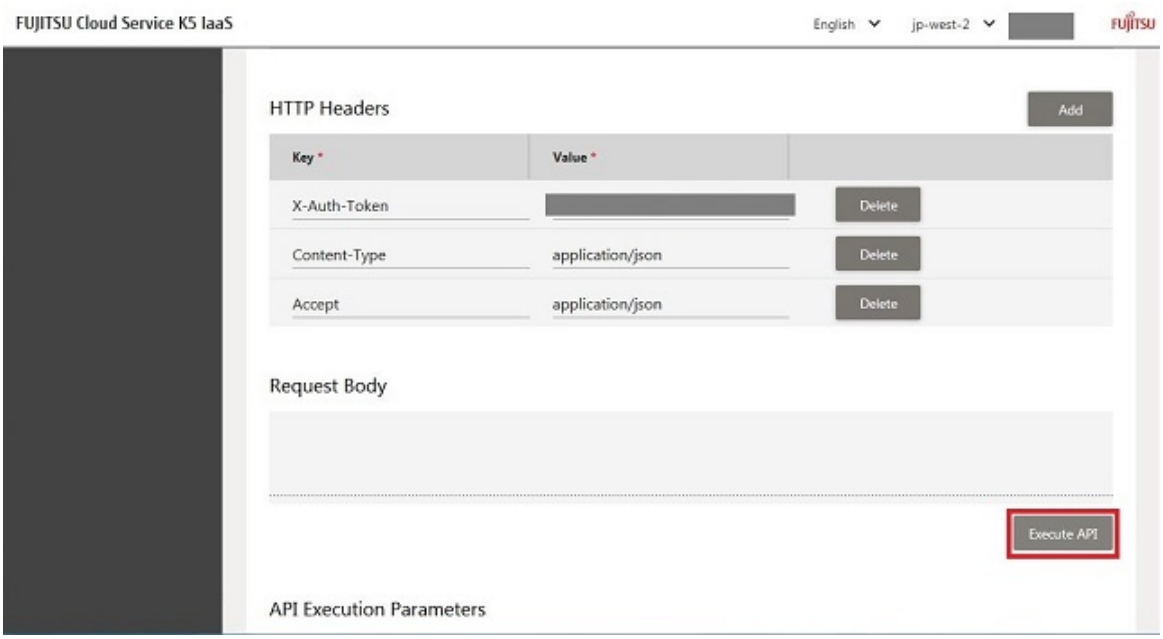


4. Add "/stacks/<name of created stack>/<noted stack ID>" to the end of the displayed URI.

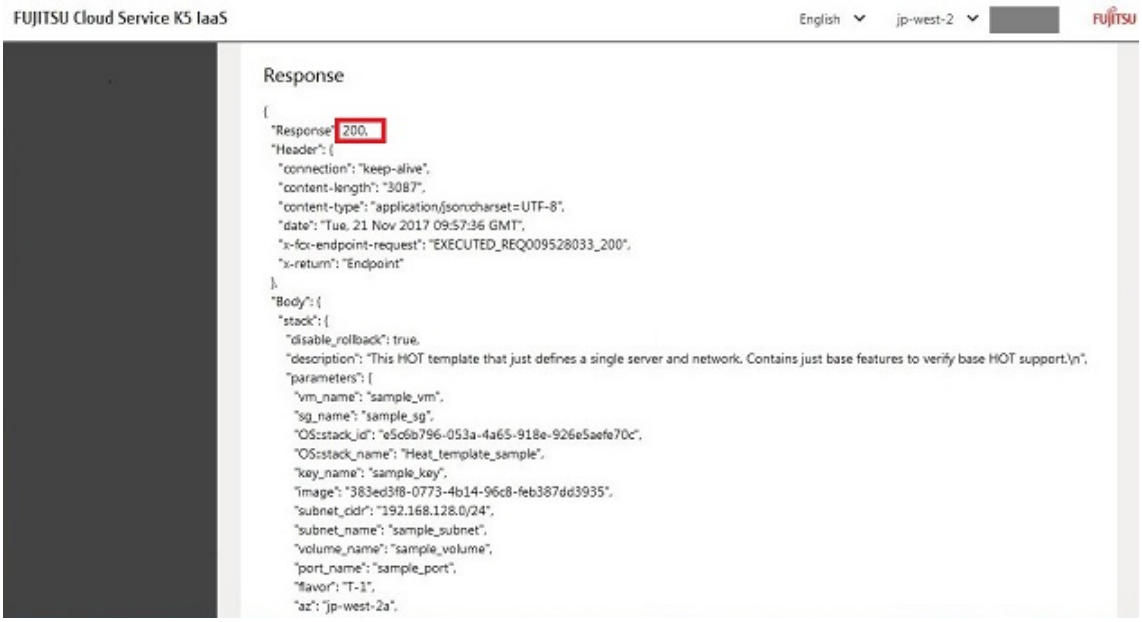


*It is only partially shown in the image, but input all of the stack ID.

5. Click "Execute API".



6. When "Response": 200 is returned in the Response field, the information was successfully obtained.



7. In the response field, check Body > stack > stack_status, and if the value is "CREATE_COMPLETE", then creation is complete.

If the value of stack_status is "CREATE_IN_PROGRESS", creation is still being performed, so wait for a while and then execute the API again.

If the value of stack_status is "CREATE_FAILED", creation has failed for some reason. Refer to "Handling errors".

C: Handling errors

C.1 Handling errors

This section explains the corrective actions to take when an error occurs during stack creation, update, or deletion.
Please understand that not all errors are covered in this section, so consider it to be reference information.

Errors during requests

When an error occurs during a request for creation, update, or deletion, handle it base on the status code in the response.

When the API was executed from the "API Execution" window of the IaaS service portal, the status code will be displayed in the response field, as shown below.

Response

```
{
  "Response":401,
  "Header":{
    "content-type": "text/plain",
    "date": "Wed 13 Sep 2017 07:54:09 GMT",
    "x-fcx-endpoint-request": "EXECUTED_REQ003200632_401",
    "x-return": "Endpoint"
  }
}
Authentication required
```

Status Code	Description
400	There may be a mistake in the content of the request body. The location where the error occurred in the response body is often described, so check the description and handle the error.
401	The error is related to authentication. Check whether an X-Auth-Token header has been specified, and if one has been specified that the value (token) is correct. Be careful, as issued tokens expire after a certain period of time.
404	The error is that the resource could not be found. There is a chance that the wrong URL was specified, so check that the stack name, ID, etc. are correct.
409	The error is a duplicate resource error. Check that the name of the stack is not the same as that of an existing stack.

Status Code	Description
500	The error is an internal error. Wait for a while and then try again. If the error occurs repeatedly, please contact the help desk.

Errors during creation, update, or deletion

After a request for creation, update, or deletion has completed, if the process has not completed correctly, determine the cause from the stack information and handle the problem.

Therefore, first, use the API to obtain the stack information. For the method to obtain the information, refer to "Request confirmation procedure".

Check Body > stack > stack_status_reason in the response field.

When the API was executed from the "API Execution" window of the IaaS service portal, the response field is displayed as shown below.

Response


```
{
 "Response": 200,
 "Header": {
  "content-type": "application/json;charset=UTF-8",
  "date": "Wed, 13 Sep 2017 09:12:01 GMT",
  "x-fcx-endpoint-request": "EXECUTED_REQ003140287_200",
  "x-return": "Endpoint"
 },
 "Body": {
  "stack": {
   "disable_rollback": true,
   "description": "This HOT template that just defines a single server and network. Contains just base features to verify base HOT support.\n",
   "parameters": {
    "vm_name": "sample_vm",
    "sg_name": "sample_sg",
    "OS::stack_id": "40effed8-cdaa-40b2-abf9-749744668dcc",
    "OS::stack_name": "HOT_sample",
    "key_name": "sample_key",
    "image": "383ed3f8-0773-4b14-96c8-feb387dd3935",
    "subnet_cidr": "bad parameter",
    "subnet_name": "sample_subnet",
    "volume_name": "sample_volume",
    "port_name": "sample_port",
    "flavor": "T-1",
    "az": "jp-west-2a",
    "network_name": "sample_network"
   },
   "stack_status_reason": "Resource CREATE failed: NeutronClientException: Invalid input for cidr. Reason: 'bad parameter' is not a valid IP subnet.",
   "stack_name": "HOT_sample",
   "creation_time": "2017-09-13T09:10:52Z",
   "links": [
    {
     "href": "https://orchestration.jp-west-2.cloud.global.fujitsu.com/v1[redacted]/stacks/HOT_sample/40effed8-cdaa-40b2-abf9-749744668dcc",
     "rel": "self"
    }
```


Take corrective action based on the content described in stack_status_reason.

The following are cases that are often the cause.

- During creation and update

Cause	Corrective Action
An input value of a parameter does not match the format required by the parameter.	Check and correct the content of the template.
The limit of the number of resources that can be created has been reached.	Refer to the Features Handbook, check the value of the limit, and delete any unnecessary resources.
A resource that no longer exists is being referenced.	Check and correct the content of the template.

- During deletion

Cause	Corrective Action
A resource that is not managed a part of the stack has been included in the stack. (For example, a manually created virtual server has been connected to the network created in a stack)	Either delete the resource that is not managed as part of a stack, or perform modification so no resource managed in the stack has a dependency with the resource.
There is a resource that is being created or updated.	Perform the operation again, after the other operation is complete.
A resource that no longer exists is being referenced.	Check and correct the content of the template.

If the cause cannot be determined, please contact the help desk.

D: Referring to outputs

D.1 Referring to outputs

This section indicates the method for viewing the value of the outputs parameter from a stack created using a Heat template in which the Outputs Section has been defined.

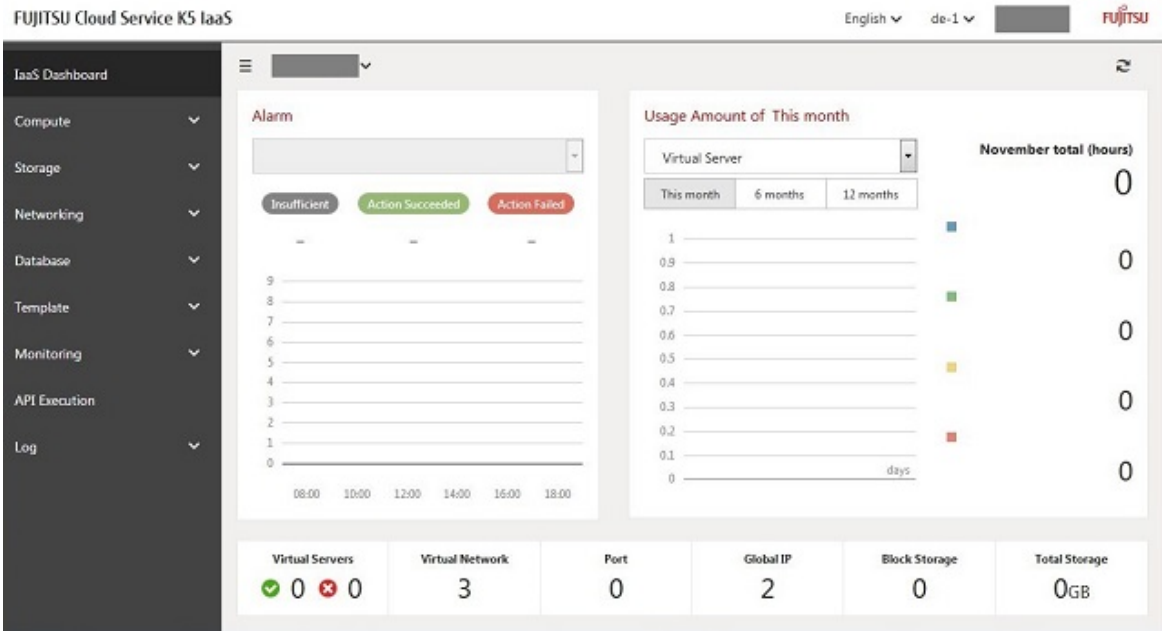
The following example indicates how to check using the API execution function of the IaaS service portal.

"Sample system configuration - Example Heat template" is used for the example.

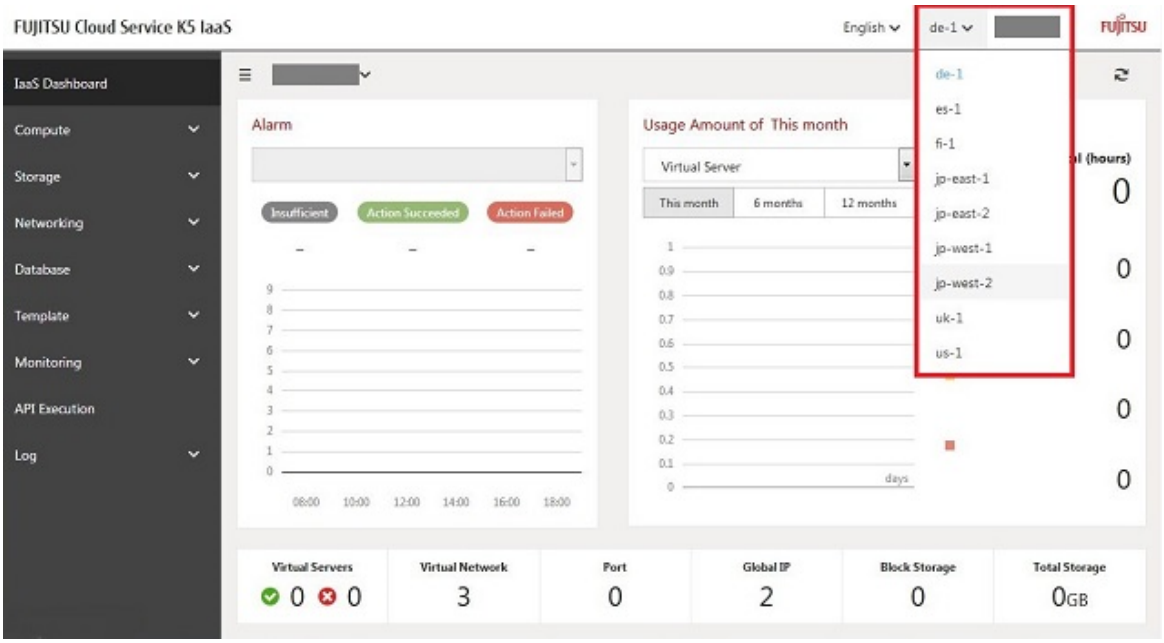
The procedure is as follows.

Procedure

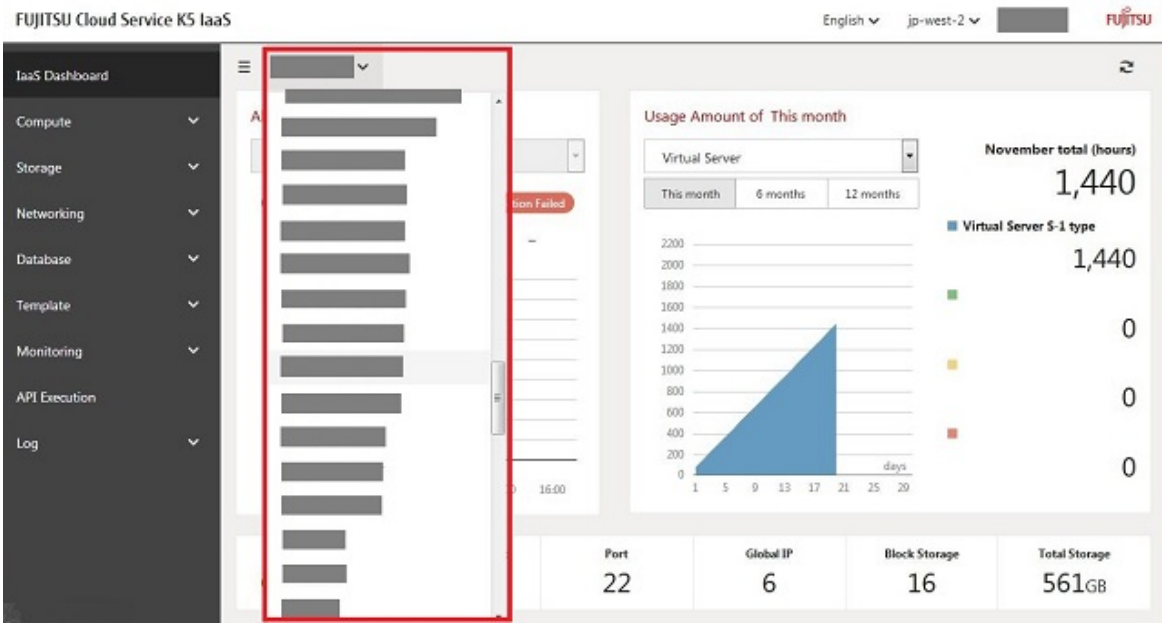
1. After logging in to the Portal, selecting "IaaS Portal" from the "Management" tab displays the "IaaS Dashboard" as below.



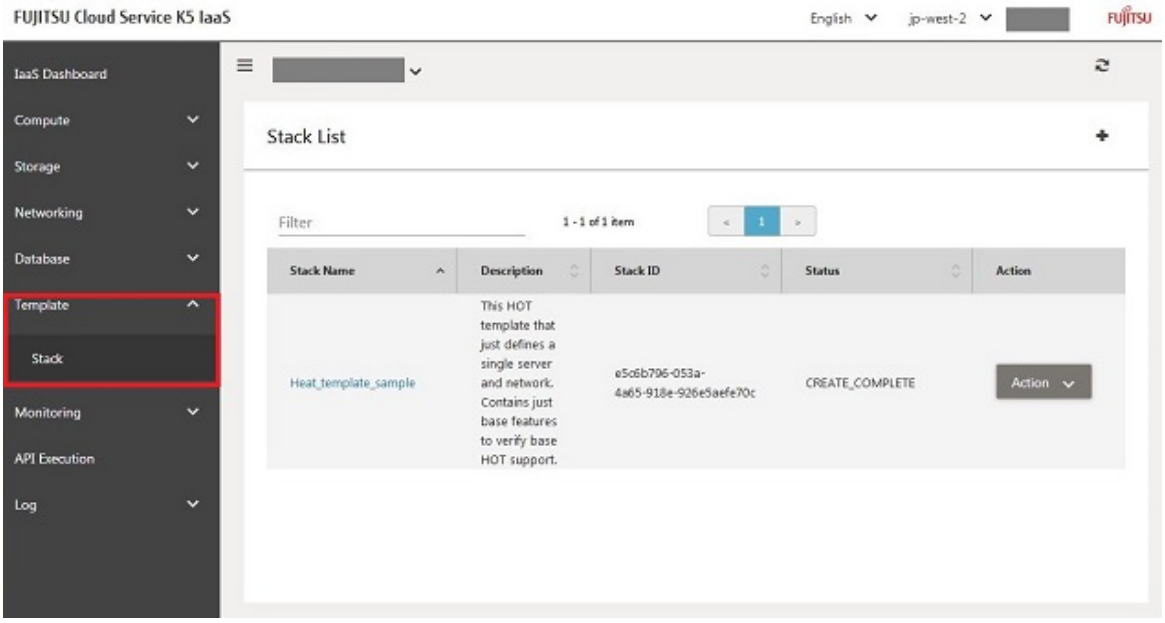
2. Select the region you want to create the stack in.



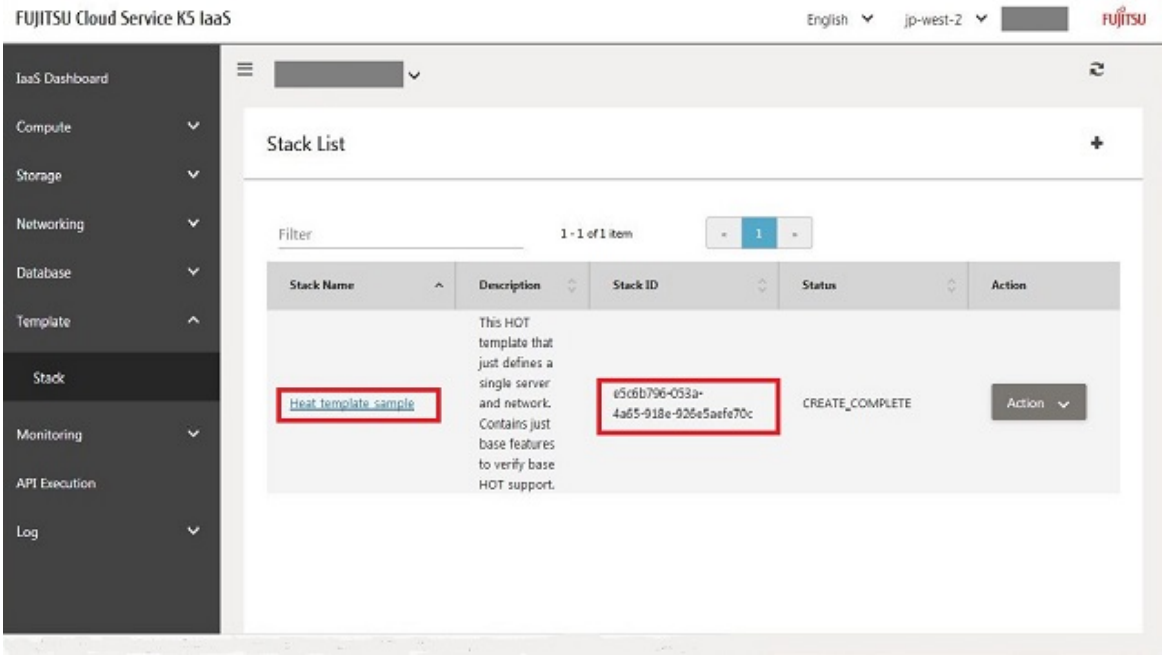
3. Select the project you want to create the stack in.



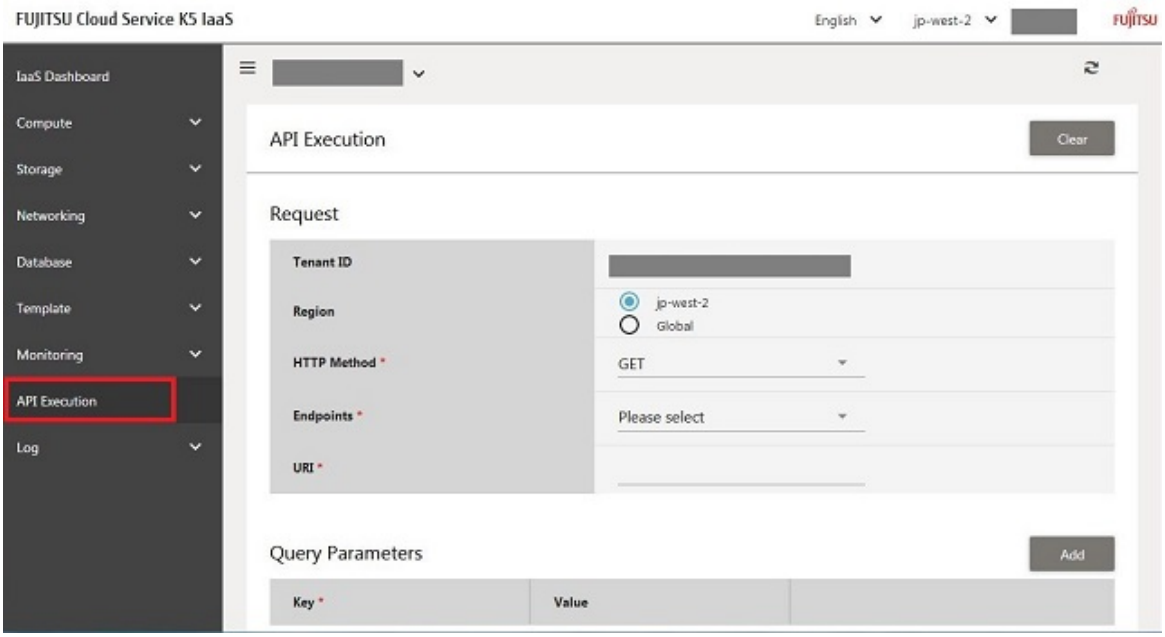
4. Clicking "Template" > "Stack" in the menu on the left displays the "Stack list" window.



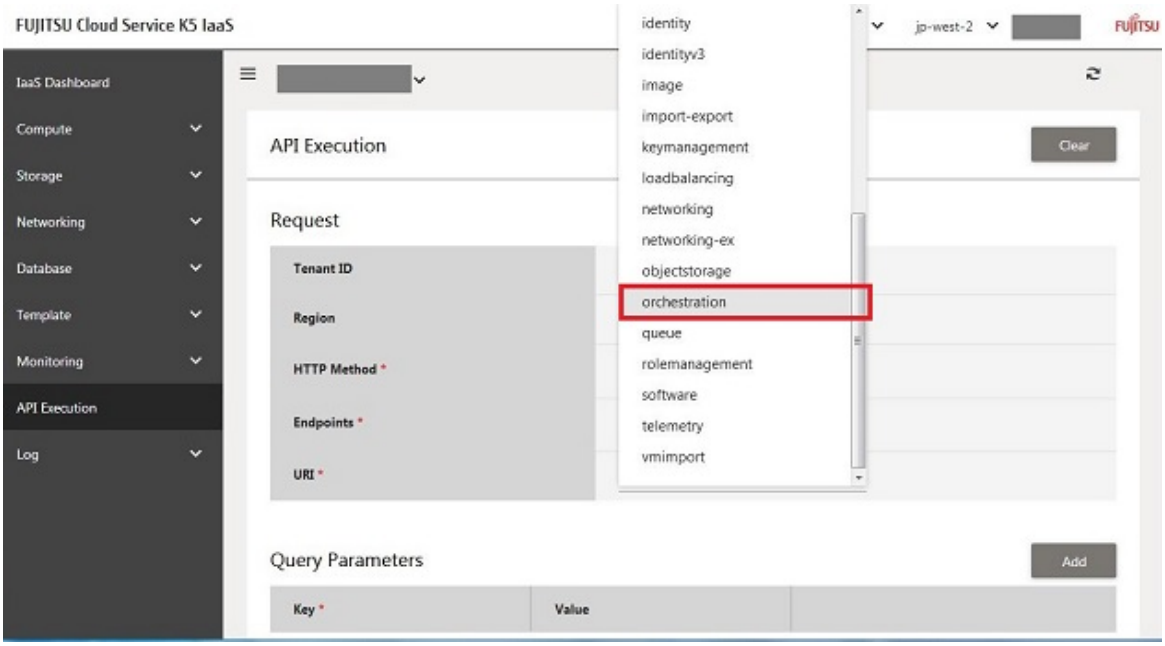
5. Take a note of the "Stack Name" and "Stack ID" of the stack you want to view the outputs of.



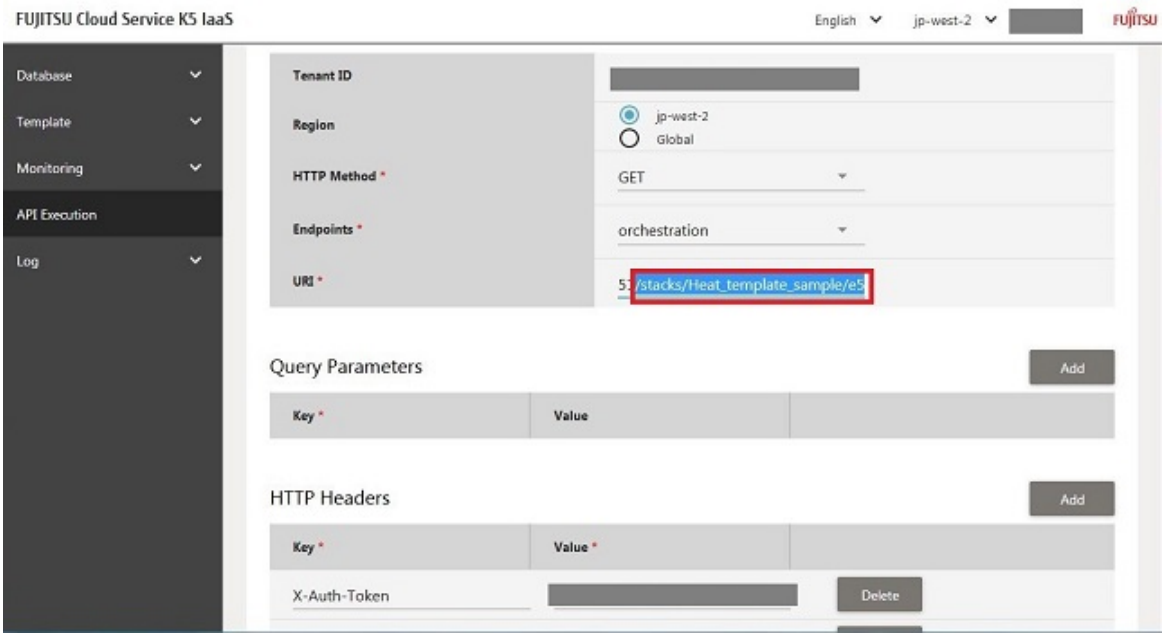
6. Clicking "API Execution" in the menu on the left displays the "API Execution" window.



7. Select "orchestration" from the endpoints.



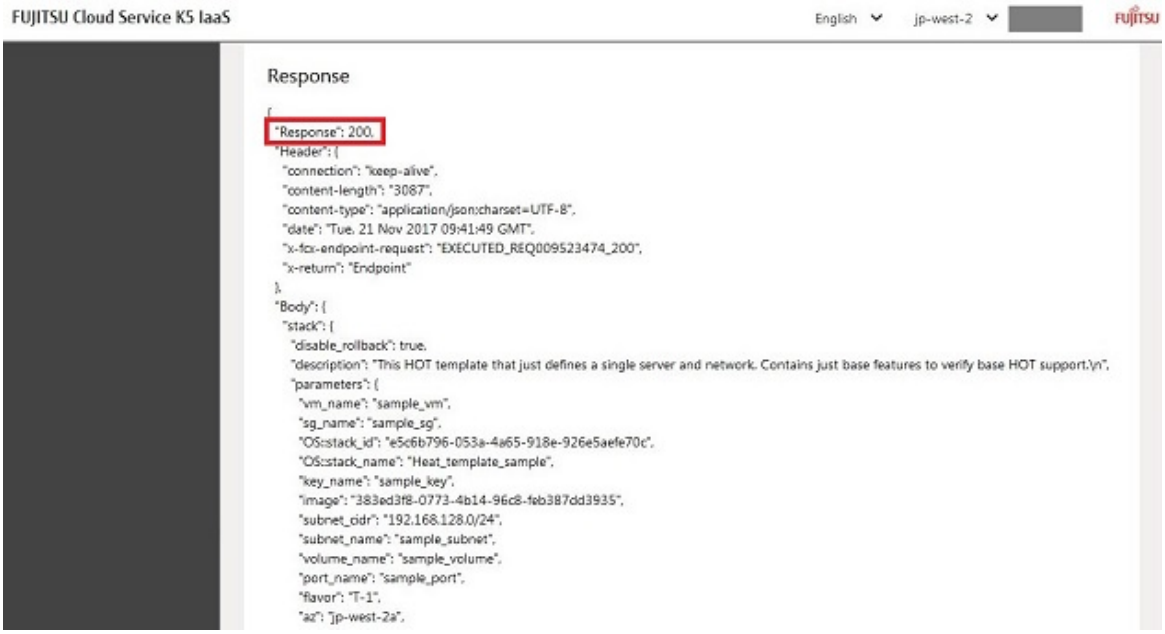
8. Add "/stacks/<name of noted stack>/<noted stack ID>" to the end of the displayed URI.



*It is only partially shown in the image, but input all of the stack ID.

9. Click "Execute API".

   When "Response": 200 is returned in the Response field, the information was successfully retrieved.



10. Check Body > stack > outputs field in the Response field, and check whether the value defined for outputs is being displayed.

In the above figure, the private key information of the key pair is displayed in outputs.

FUJITSU Cloud Service for OSS

IaaS Heat Template Specifications

Version 2.7

Published Date July 2018