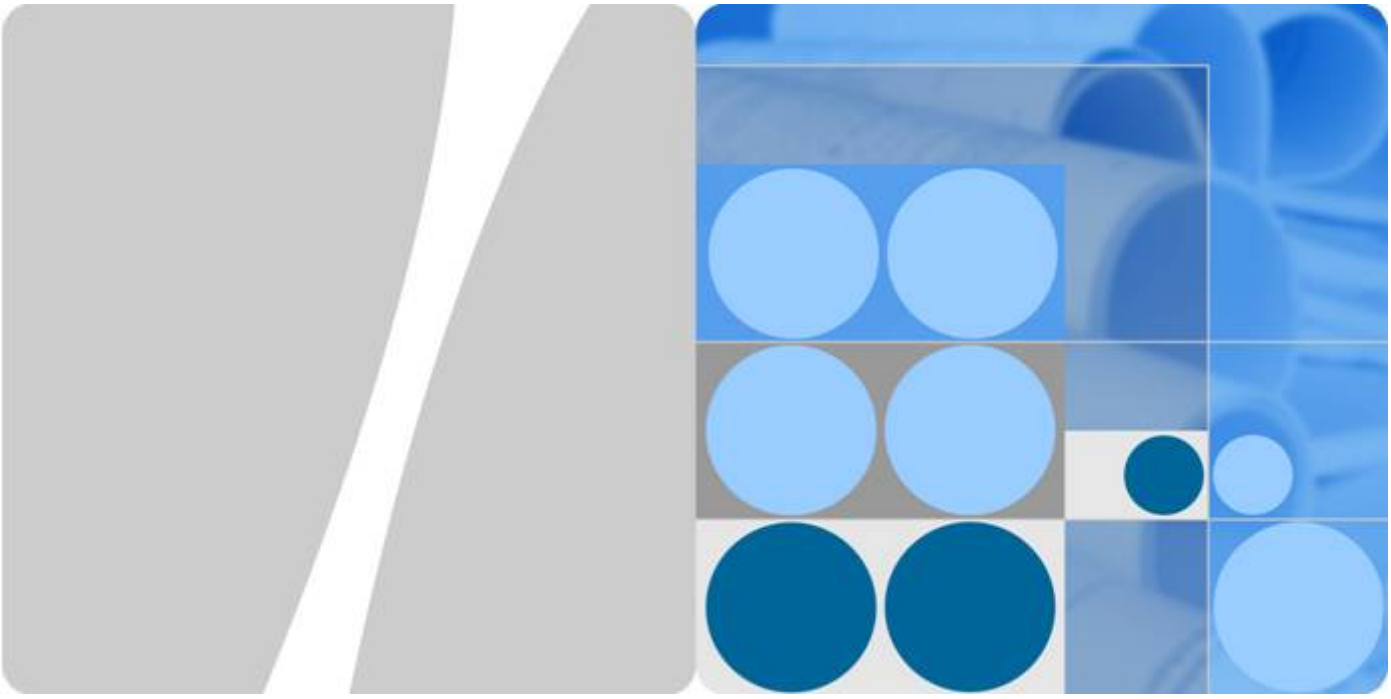

Resource Template Service

Product Description

Issue 02

Date 2018-05-30

Contents

1 Resource Template Service

Resource Template Service (RTS) makes it easier for you to simplify your cloud computing resource management and automatic O&M. Based on the template specifications defined in the RTS service, you can compile a template file and define a collection of cloud computing resources, dependencies between resources, and resource configurations. Then you can automatically create and configure all resources in the template by using the orchestration engine so as to implement automatic deployment and simplify O&M. The RTS service supports most native OpenStack Heat APIs and templates in the Heat Orchestration Template (HOT) format.

You can use RTS by calling RTS Application Programming Interface (API) or using the RTS console. The RTS console is a visualized user interface provided for RTS. With the RTS console, you can use templates to automatically deploy resources.

Concepts

- Template

  A cloud orchestration template is a user-readable, easy-to-compile file that describes how to assemble a set of resources and install required software. This template specifies resources to be used, attributes to be set, and parameters indispensible to the success of automation for a specific application deployment.

- Resource

  A resource is a template that specifies the component of your desired architecture, including Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Cloud Eye, Relational Database Service (RDS), Scalable File Service (SFS), Auto Scaling (AS), and Virtual Private Cloud (VPC) .

- Stack

  A stack is a collection of useful resources, including servers and databases. With a template, Heat orchestration engine helps create an instantiated set of resources, that is, a stack, to accommodate the specified application framework or component (contained in a template). A stack is actually a running instance of a template. Creating a stack is the deployment of an application framework or component.

- Regions and AZs

  A region is a geographic area where resources used by your RTSs are located.

  An availability zone (AZ) is a physical location where resources use independent power supply and networks within a region. An AZ is insulated from failures in other AZs and provides inexpensive, low-latency network connectivity to other AZs in the same region.

A region can have more than one AZ. AZs are physically isolated but interconnected through an internal network.

- Project

  Projects are used to group and isolate OpenStack resources, including computing, storage, and network resources. Multiple projects can be created for one account. A project can be a department or a project team.

Product Functions

- Stack management: A stack is cloud application that is created based on a template. You can create, update, and delete a stack as well as manage resources, events, and templates in a stack.
- Resource orchestration: You only need to create one resource configuration template in the JSON or YAML format. RTS analyzes resource dependencies based on the template and orchestrates resources in the public cloud environment in sequence.
- Auto scaling: If services, such as web servers, need to be deployed on more than one ECSs, users require that auto scaling to be implemented for the services based on ECS workloads. RTS provides the auto scaling function to meet users' requirements. The auto scaling function provided by RTS is different from the Auto Scaling service provided in the public cloud system. The auto scaling function is available only to RTS, and no APIs are exposed to users.
- Software configuration: This function allows you to configure software on your servers.
- Software deployment: This function is used to manage the life cycle of software.

2 Product Advantages

The RTS can help users to uniformly model and configure the resources on the cloud. You only need to create a template file that describes the required resources and the dependencies between resources. The RTS will automatically create and configure all the resources in the template through the orchestration engine, so that you can manage cloud resources more simply and easily. The RTS has the following advantages:

Providing Sample Templates to Make Creating Templates Easy

The **Examples Template** of the RTS console provides some common templates for you to choose. You only need to set the values of several simple parameters (such as the region, stack name, etc.).

Flexible Combination of Multiple Cloud Services

Currently, the RTS service supports a variety of core cloud services, such as ECS, EVS, VPC, AS, CES, SFS, and RDS. You can flexibly combine cloud services through templates to meet the automation operation and maintenance requirements of different service scenarios.

Batch Operation to Simplify Resource Management

By defining multiple resource types and resource dependencies in the template, all the services in the template can be deployed in one batch, which greatly simplifies the deployment complexity and saves your time cost. You can also dynamically adjust the stack template based on your business needs; if you no longer need the group resources, you can delete all resources in one click.

3 Supported Resources

RTS supports 33 resource types, including alarms and infrastructure resources, such as computing, network, and storage resources. **Table 3-1** lists all supported resources. You can also view the currently supported resources through the **Resource Type** page of the management console.

For details about descriptions of templates, see chapter **Resource Template Reference**.

NOTE

OSE means OpenStack Extension, which indicates resource types that do not belong to the default resource types defined by OpenStack Heat or the types that can be defined by you.

Table 3-1 Resource types supported by RTS

Resource Type	Description	RTS support	Heat Support	Dependency
OS::Cinder::Volume	A resource that provides Cinder volumes.	Y	Y	EVS
OS::Cinder::VolumeAttachment	A resource for associating volumes with instances.	Y	Y	EVS
OS::Heat::AutoScalingGroup	An auto scaling group that can scale arbitrary resources.	Y	Y	RTS
OS::Heat::CloudConfig	A configuration resource for representing Cloud-init cloud-config.	Y	Y	RTS
OS::Heat::MultipartMime	Assembles a collection of software configurations as a multi-part mime.	Y	Y	RTS
OS::Heat::RandomString	A resource which generates a random string.	Y	Y	RTS

Resource Type	Description	RTS support	Heat Support	Dependency
OS::Heat::ResourceGroup	Creates one or more identically configured, nested resources.	Y	Y	RTS
OS::Heat::ScalingPolicy	A resource to manage scaling of OS::Heat::AutoScalingGroup.	Y	Y	RTS
OS::Heat::SoftwareComponent	A resource for describing and storing a software component.	Y	Y	RTS
OS::Heat::SoftwareConfig	A resource for describing and storing software configuration.	Y	Y	RTS
OS::Heat::StructuredConfig	A resource which has the same logic with OS::Heat::SoftwareConfig.	Y	Y	RTS
OS::Heat::WaitCondition	A resource for handling signals received by WaitConditionHandle.	Y	Y	RTS
OS::Heat::WaitConditionHandle	A resource for managing instance signals.	Y	Y	RTS
OS::Neutron::FloatingIP	A resource for managing Neutron floating IP addresses.	Y	Y	VPC
OS::Neutron::FloatingIPAssociation	A resource for associating floating IP addresses and ports.	Y	Y	VPC
OS::Neutron::Net	A resource for managing Neutron networks.	Y	Y	VPC
OS::Neutron::Port	A resource for managing Neutron ports.	Y	Y	VPC
OS::Neutron::Router	A resource that implements Neutron routers.	Y	Y	VPC
OS::Neutron::RouterInterface	A resource for managing Neutron router interfaces.	Y	Y	VPC
OS::Neutron::SecurityGroup	A resource for managing Neutron security groups.	Y	Y	VPC
OS::Neutron::Subnet	A resource for managing Neutron subnets.	Y	Y	VPC

Resource Type	Description	RTS support	Heat Support	Dependency
OS::Nova::KeyPair	A resource for creating Nova key pairs.	Y	Y	ECS
OS::Nova::Server	A resource for managing Nova instances.	Y	Y	ECS
OS::Nova::ServerGroup	A resource for managing Nova server groups.	Y	Y	ECS
OSE::AS::ScalingConfig	A resource for managing autoscaling configuration.	Y	N	AS
OSE::AS::ScalingGroup	A resource for managing autoscaling group.	Y	N	AS
OSE::AS::ScalingPolicy	A resource for managing autoscaling policy.	Y	N	AS
OSE::CES::Alarm	A resource for managing Cloud Eye service's alarms.	Y	N	CES
OSE::RDS::Instance	A resource for managing RDS instances.	Y	N	RDS
OSE::VPC::Vpc	A resource for managing VPCs.	Y	N	VPC
OSE::VPC::Subnet	A resource for managing VPC subnets.	Y	N	VPC
OSE::SFS::Share	A resource for creating SFS shared files.	Y	N	SFS
OSE::SFS::ShareAccessRule	A resource for creating policies for accessing SFS shared files.	Y	N	SFS

4 Supported APIs

Table 4-1 Supported APIs

NO.	API	Description	RTS Support	Heat Support
1	GET /	List versions	Y	Y
2	POST /v1/{tenant_id}/stacks	Create stack	Y	Y
3	GET /v1/{tenant_id}/stacks	List stack	Y	Y
4	POST /v1/{tenant_id}/stacks/preview	Preview stack	Y	Y
5	GET /v1/{tenant_id}/stacks/{stack_name}	Find stack	Y	Y
6	GET /v1/{tenant_id}/stacks/{stack_name}/resources	Find stack resources	Y	Y
7	GET /v1/{tenant_id}/stacks/{stack_name}/{stack_id}	Show stack details	Y	Y
8	PUT /v1/{tenant_id}/stacks/{stack_name}/{stack_id}	Update stack	Y	Y
9	DELETE /v1/{tenant_id}/stacks/{stack_name}/{stack_id}	Delete stack	Y	Y
10	POST /v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Cancel a stack update **NOTE** If this API is called during a stack update, the related resources may in the rollback state. As a result, the resources involved become abnormal.	Y	Y

NO.	API	Description	RTS Support	Heat Support
11	POST /v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Check stack resources	Y	Y
12	GET /v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources	List resources	Y	Y
13	GET /v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}	Show resource data	Y	Y
14	GET /v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/metadata	Show resource metadata	Y	Y
15	POST /v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/signal	Send a signal to a resource	Y	Y
16	GET /v1/{tenant_id}/stacks/{stack_name}/events	Find stack events	Y	Y
17	GET /v1/{tenant_id}/stacks/{stack_name}/{stack_id}/events	List stack events	Y	Y
18	GET /v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/events	List resource events	Y	Y
19	GET /v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/events/{event_id}	Show event details	Y	Y
20	GET /v1/{tenant_id}/stacks/{stack_name}/{stack_id}/template	Get stack template	Y	Y
21	POST /v1/{tenant_id}/validate	Validate template	Y	Y
22	GET /v1/{tenant_id}/resource_types/{type_name}/template	Show resource template	Y	Y
23	GET /v1/{tenant_id}/resource_types/{type_name}	Show resource schema	Y	Y

NO.	API	Description	RTS Support	Heat Support
24	GET /v1/{tenant_id}/resource_types	List resource types	Y	Y
25	GET /v1/{tenant_id}/build_info	Show build information	Y	Y
26	POST /v1/{tenant_id}/software_configs	Create configuration	Y	Y
27	GET /v1/{tenant_id}/software_configs/{config_id}	Show configuration details	Y	Y
28	DELETE /v1/{tenant_id}/software_configs/{config_id}	Delete configuration	Y	Y
29	PATCH /v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name_or_physical_id}	Mark a resource as unhealthy	Y	Y

NOTE

For details, see *Resource Template Service API Reference*.

5 Constraints and Limitations

Constraints

- RTS does not support snapshot-related APIs and the following operations: suspend, restore, abandon, and adopt.
- The AZ attribute in a template cannot be updated during a stack life cycle.
- Names of different resources must be different in the same level.

Specification Limitations

- Each tenant can create a maximum of 100 stacks.
- Each stack can contain a maximum of 1000 resources.
- Each stack can contain a maximum of 1000 operation logs.
- A maximum of seven stacks can be nested.
- The maximum size of the template file after extracting the **zip** package is 512 KB.

6 Heat Client Version

The supported Heat client version is 1.5.1.

You can obtain the client software at:

https://github.com/openstack/python-heatclient/releases/tag/1.5.1

7 User Permissions

The public cloud system provides two types of permissions by default: user management and resource management.

- User management refers to the management of users, user groups, and user group rights.
- Resource management refers to the control operations that can be performed by users on cloud service resources.

For further details, see **Permission Description**.

8 Change History

Release Date	Description
2018-05-30	This issue is the second official release. Added five resource types supported by RTS. For details, see **Supported Resources**.
2018-03-15	This issue is the first official release.