Application Orchestration Service

User Guide

Issue	01
Date	2020-08-05

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: https://www.huawei.com

Email: support@huawei.com

Contents

1 Introduction

This chapter introduces how to use Application Orchestration Service (AOS).

With AOS, you can deploy applications in the cloud by writing templates (declarations of resources that make up stacks) and creating stacks from the templates, as shown in **Figure 1-1**. AOS also provides application lifecycle management features, such as starting, changing, and deleting.

Figure 1-1 How AOS works

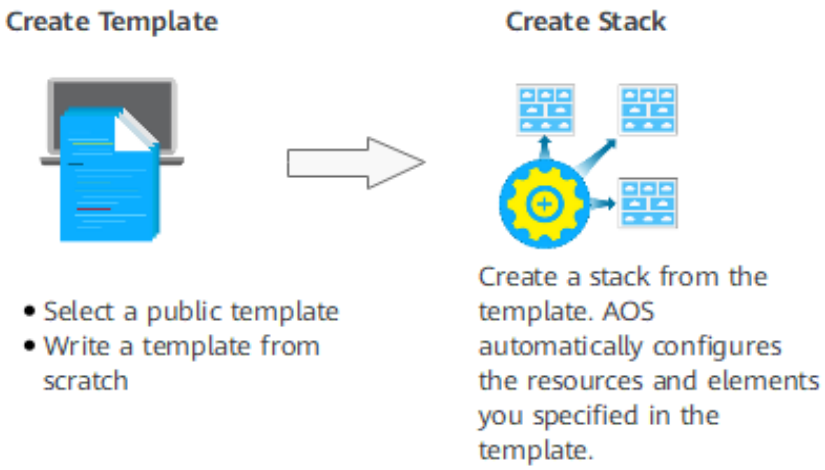


Table 1-1 How AOS works

Step	Description
1. Create a template	A template is a text file that uses AOS syntax to describe application attributes, cloud service configurations, and dependencies between applications and cloud services. How to obtain a template: • Select a public template: AOS provides multiple public templates from which you can quickly create stacks. • Write a template from scratch: You can write a template from scratch in JSON or YAML format. Before writing a template, gain a basic understanding of AOS templates from **Templates (Cloud-based Automation Scripts)**. In addition to writing a template on the web UI, you can also write it on a local host and then upload it to AOS. For details on how to orchestrate and deploy resources by writing a template, see **Writing a Template to Create an ECS**. **NOTICE** The YAML syntax does not support the Tab key. The hierarchical relationship must be aligned with an even number of **spaces**, such as 2, 4, 6, or 8 spaces. For more information about templates, see **Templates (Cloud-Based Automation Scripts)**.
2. Create a stack	A stack is a collection of applications and cloud service resources. The applications or cloud services in a stack are treated as a unit when being created or deleted. You can create stacks from templates. After you select a template and specify stack parameters, AOS automatically sets up the resources and elements you specify in the template. For more information about stacks, see **Stack Management**.

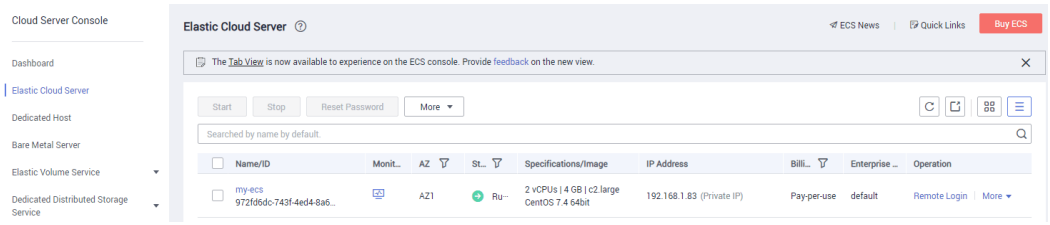
2 Getting Started

2.1 Writing a Template to Create an ECS

This section describes how to create an Elastic Cloud Server (ECS), including a Virtual Private Cloud (VPC) and subnet by writing a template. An ECS is a computing server equipped with CPUs, memory, images, and Elastic Volume Service (EVS) disks. ECSs can be created on demand and supports auto scaling. A VPC provides logically isolated, configurable, and manageable virtual networks for your ECSs. One or more subnets are automatically created when you create a VPC.

At the end of this walkthrough, you will see the newly created ECS on the **Cloud Server Console**, as shown in **Figure 2-1**.

Figure 2-1 Created ECS



NOTE

In addition to writing templates from scratch, you can also select public templates to create templates. For more information about templates, see **Templates (Cloud-Based Automation Scripts)**.

In this section, you will complete the following steps:

1. **Step 1: Write a Template**: Use the YAML language to write a template for creating an ECS, VPC, and subnet.
2. **Step 2: Create an ECS**: Use the template to create an ECS, VPC, and subnet.
3. **Step 3: Delete Unnecessary Resources**: Delete unnecessary stack to avoid unwanted charges.

Step 1: Write a Template

Step 1 Write a simple template to create a VPC.

```
tosca_definitions_version: huaweicloud_tosca_version_1_0   #Template version information
node_templates:                #Element object definition
  myvpc:                      #VPC
    type: HuaweiCloud.VPC.VPC
    properties:
      name: my-vpc             #Name of the VPC
      cidr: '192.168.0.0/16'   #VPC CIDR
```

This template includes:

1. **tosca_definitions_version**: specifies the version of a template. Currently, only **huaweicloud_tosca_version_1_0** is supported by AOS.
2. **node_templates**: defines the set of objects to be orchestrated in a template. In AOS, objects are used interchangeably with elements. An object can be an application or cloud service resource. In the preceding template, **node_templates** defines the **myvpc** VPC.
3. **type**: specifies the type of an orchestration object. The value comes from the element type list and can be set to **HuaweiCloud.******* (***** indicates the element name in the **Resource Indexes**). In the preceding template, the **myvpc** VPC type is **HuaweiCloud.VPC.VPC**.
4. **properties**: defines element properties, which vary with element types. In the preceding template, the **myvpc** VPC has the **names** and **cidr** properties, which indicate the name and network segment of the VPC, respectively. For more information, see **VPC.VPC**.

Step 2 Define a subnet in the VPC. A VPC is a large network segment and is usually divided into several subnets. Define a subnet in the created VPC based on the preceding template.

```
tosca_definitions_version: huaweicloud_tosca_version_1_0   #Template version information
node_templates:                #Element object definition
  myvpc:                      #VPC
    type: HuaweiCloud.VPC.VPC
    properties:
      name: my-vpc             #Name of the VPC
      cidr: '192.168.0.0/16'   #VPC CIDR
  mysubnet:                   #Subnet
    type: HuaweiCloud.VPC.Subnet
    properties:
      name: my-subnet          #Name of the subnet
      cidr: '192.168.1.0/24'   #Subnet CIDR
      gateway: 192.168.1.1     #Gateway of the subnet
      vpcId:                  #ID of the VPC to which the subnet belongs
        get_reference: myvpc
      dhcpEnable: true         #Determines whether to enable the DHCP function for the subnet in the VPC.
    requirements:             #Dependency between the subnet and VPC.
      - vpcId:
          node: myvpc
```

The **requirements** parameter specifies the element that has a dependency relationship with the current element. For example, define **myvpc** as a dependent node in the **requirements** of the subnet because a subnet depends on a VPC.

Step 3 Define an ECS in the template.

```
tosca_definitions_version: huaweicloud_tosca_version_1_0   #Template version information
node_templates:                #Element object definition
  myvpc:                      #VPC
    type: HuaweiCloud.VPC.VPC
    properties:
```



```
    name: my-vpc              #Name of the VPC
    cidr: '192.168.0.0/16'    #VPC CIDR
 mysubnet:                    #Subnet
  type: HuaweiCloud.VPC.Subnet
  properties:
    name: my-subnet           #Name of the subnet
    cidr: '192.168.1.0/24'    #Subnet CIDR
    gateway: 192.168.1.1      #Gateway of the subnet
    vpcId:                    #ID of the VPC to which the subnet belongs
      get_reference: myvpc
    dhcpEnable: true          #Determines whether to enable the DHCP function for the subnet in the VPC.
  requirements:               #Dependency between the subnet and VPC.
    - vpcId:
        node: myvpc
 myecs:                       #ECS
  type: HuaweiCloud.ECS.CloudServer
  properties:
    name: my-ecs              #Name of the ECS
    instances: 1              #Number of created ECSs
    imageId: 7be2e72e-0679-4a1b-8faf-0c1865708b20    #Image ID used by the ECS. In this template, the image ID is the ID of the system disk based on 64-bit CentOS 7.4.
    flavor: c2.large          #Specifications of the ECS
    vpcId:                    #ID of the VPC to which the ECS belongs. Either a new or an existing VPC ID can be used.
      get_reference: myvpc    #Obtains the dynamic attribute value of the associated element.
    availabilityZone: cn-south-1a      #AZ to which the ECS belongs. This template uses an AZ in the CN South-Guangzhou region.
    nics:                     #NIC of the ECS
      - subnetId:
          get_reference: mysubnet
    rootVolume:               #System disk configuration of the ECS
      volumeType: SATA        #Common I/O disk type
      size: 40                #System disk size (unit: GB)
  requirements:               #Dependency among the ECS, VPC, and subnet.
    - vpcId:
        node: myvpc
    - nics.subnetId:
        node: mysubnet
```

Step 4 Save the template as a local file **myecs.yaml**.

Step 5 Log in to the AOS console.

NOTE

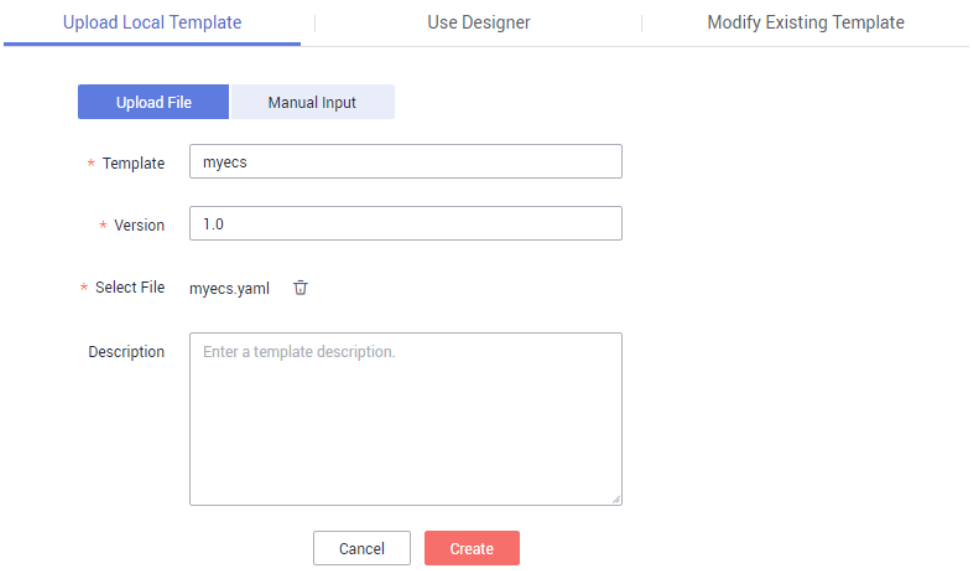
In this template, the image ID is the ID of the **CentOS 6.3 64bit** image, and the AZ is **cn-south-1a** in the **CN South-Guangzhou** region. Therefore, select the **CN South-Guangzhou** region after logging in to the console to avoid stack creation failure.

Step 6 In the navigation pane, choose **My Templates**, and then click **Create Template**.

Step 7 On the **Upload Local Template** tab page, specify the following parameters, upload a local YAML file, and then click **Create**. The template details page is then displayed, showing the template information.

- **Template**: Enter a template name. Each template name must be globally unique. For example, set this parameter to **myecs**.
- **Version**: Set this parameter to 1.0.
- **Select File**: Upload the **myecs.yaml** file.

Figure 2-2 Uploading a template from the local host



----End

Step 2: Create an ECS

Step 1 Log in to the AOS console.

Step 2 In the navigation pane, choose **My Templates**. The **myecs** template is displayed in the template list.

Step 3 Click **Create Stack** in the **Operation** column of the **myecs** template.

Step 4 Set the stack information.

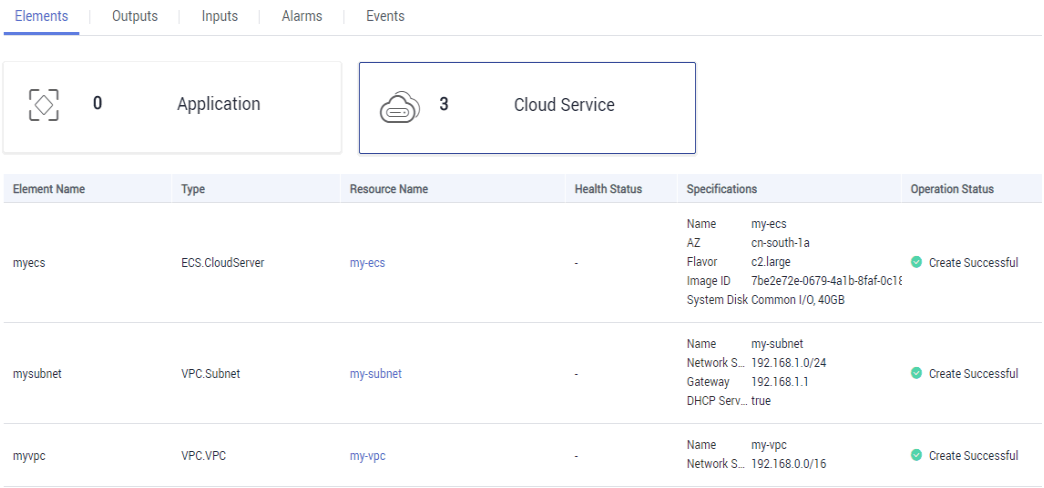
- **Stack Name**: Enter a unique stack name, for example, **aos-ecs**.
- **Description**: The description can be left blank.

Step 5 Click **Next** and check the stack information. If the stack information is correct, click **Create Stack**.

The stack details page is displayed, showing that the stack is being created. The stack includes a VPC, a subnet, and an ECS. It will take about 6 minutes to create the stack.

Step 6 Wait until the stack status becomes **Normal**. The VPC, subnet, and ECS are created and displayed in the stack element list.

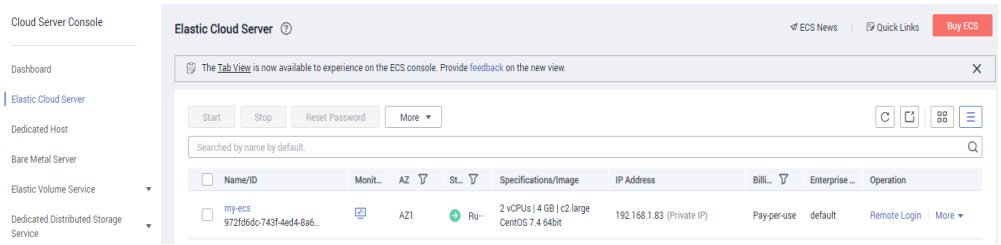
Figure 2-3 Created stack



Step 7 View the created cloud services.

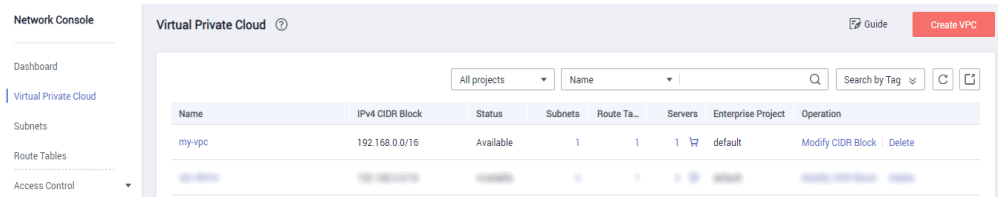
1. Log in to the management console.
2. Choose **Service List** > **Computing** > **Elastic Cloud Server**. You will see the newly created ECS on the ECS list.

Figure 2-4 Created ECS



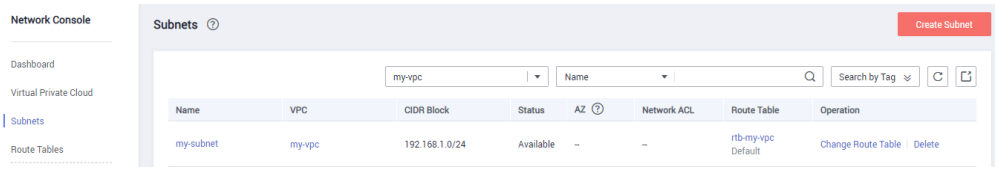
3. Choose **Service List** > **Network** > **Virtual Private Cloud**. You will see the newly created VPC on the VPC list.

Figure 2-5 Created VPC



4. Click the VPC name to show more details about the VPC. On the VPC details page, you will see that the subnet has been created in the VPC.

Figure 2-6 Created subnet



----End

Step 3: Delete Unnecessary Resources

Delete unnecessary stack resources to avoid unwanted charges.

Step 1 Log in to the AOS console.

Step 2 In the navigation pane, click **My Stacks**.

Step 3 Select the stack that will no longer be used, and click **Delete** to delete the stack.

----End

3 Stack Management

Stack management consists of two aspects. One is lifecycle management of created stacks, including deleting and changing. The other is viewing stack details to obtain stack running status.

Table 3-1 describes stack lifecycle status.

Table 3-1 Status description

Status	Description
Normal	Both the stack and its instances run properly.
Abnormal	The stack runs abnormally. Some or all stack instances run abnormally and cannot provide functions.
Initializing	Stack instances have not been installed or have been uninstalled. The stack does not provide functions.
Processing	A stack lifecycle action is being performed. The status of stack instances is unknown.
Unknown error	An unknown stack error occurs.

Changing a Stack

After a stack is created successfully (that is, in the normal status), you can change input parameters based on your service requirements.

Step 1 Log in to the Application Orchestration Service (AOS) console.

Step 2 In the navigation pane, click **My Stacks**.

Step 3 In the stack list, click the stack to be changed.

Step 4 On the stack details page, click **Change**.

Step 5 Change the template version or input parameters, and click **Next**.

Step 6 Confirm the configurations and then click **Change**.

On the **Events** tab page, view the detailed operation events related to stack change.

----End

The change rules are as follows:

1. Template change rules:
   a. During template change, only the following elements are allowed to be added or deleted.
      - AOS.Agent, AOS.AppComponent, AOS.AppComponent.Container, AOS.AppComponent.Process, AOS.AppGroup, AOS.Application.Stateful, AOS.Application.Stateless, AOS.Job, AOS.Service, AOS.StatelessApplication, and AOS.StatefulApplication
      - CCE.Addon.AutoScaler, CCE.ConfigMap, CCE.DaemonSet, CCE.Deployment, CCE.HelmRelease, CCE.Ingress, CCE.Job, CCE.NodePool, CCE.Secret, CCE.Service, CCE.StatefulSet, CCE.Storage.OBS, and CCE.Storage.SFS
      - ECS.CloudServer
      - ServiceStage.Agent, ServiceStage.AppGroup, ServiceStage.ContainerComponent, ServiceStage.Job, ServiceStage.ProcessComponent, ServiceStage.Service, ServiceStage.SoftwareComponent, ServiceStage.StatefulApplication, and ServiceStage.StatelessApplication
      - nodes.Agent, nodes.AppGroup, nodes.ContainerComponent, nodes.HelmApplication, nodes.Job, nodes.ProcessComponent, nodes.Service, nodes.SoftwareComponent, nodes.StatefulApplication, and nodes.StatelessApplication
   b. Do not modify the policies of the template.
   c. Do not modify the association relationship between existing elements.
   d. Do not delete the association relationship between existing elements alone. If necessary, delete both the elements and their relationship.
   e. Do not add a relationship between a new element and an existing element.
2. Parameter change rules:
   a. Do not change the input parameters of the following elements:
      - AOS.Job and AOS.Service
      - APM.AutoScaler
      - ServiceStage.Job and ServiceStage.Service
      - nodes.Agent, nodes.AutoScaler, nodes.DynamicNodePool, nodes.DynamicVolume, nodes.FlexVolume, nodes.Job, nodes.NodePool, nodes.Service, and nodes.StatefulApplication
   b. Do not change the input parameters of the following elements unless in cluster 1.7 or later:

- AOS.Agent and AOS.StatefulApplication
- ServiceStage.Agent and ServiceStage.StatefulApplication

Deleting a Stack

Deleted stacks cannot be restored. Exercise caution when deleting a stack.

Step 1 Log in to the AOS console.

Step 2 In the navigation pane, click **My Stacks**.

Step 3 In the stack list, select the stack to be deleted and click **Delete**.

Step 4 In the dialog box that is displayed, click **OK**.

Check the stack name carefully. The deletion cannot be revoked.

On the **Events** tab page, view the detailed operation events related to stack deletion.

NOTE

If the stack status remains **Deleting** until a timeout message is displayed and the stack status becomes **Abnormal**, try to forcibly delete the stack.

----End

Viewing Stack Details

After a stack is created, view its data and resources on the stack details page.

- Stack elements

  The elements of a stack, such as applications and cloud services are displayed.

  Element health status:

  – Healthy: The resource is running properly.
  – Unknown: The AOS fails to obtain the resource status because an error occurs during the health check.
  – Abnormal: The AOS successfully calls the health check API of the resource, but the resource status is abnormal.
- Output parameters

  Output parameters and their values in the stack template are displayed.
- Input parameters

  Input parameters and their values in the stack template are displayed.
- Alarms

  Alarm information of the stack is displayed.
- Events

  View stack events to monitor stack operation progress. For example, when you create a stack, all important steps during the stack creation are displayed on the **Events** tab page. The events are sorted in chronological order with the latest event being displayed at the top.

4 CTS

4.1 AOS Operations Supported by CTS

Cloud Trace Service (CTS) records all operations performed on cloud services, providing data support for customers in fault locating, resource management, and security auditing. When you enable CTS, it begins to record operations performed on Application Orchestration Service (AOS) resources. CTS stores operation records from the last seven days.

Table 4-1 AOS operations supported by CTS

Operation	Description
CreateTemplate	Creating a template
DeleteTemplate	Deleting a template
UpdateTemplate	Updating a stack
PreviewStack	Previewing a stack
CreateStack	Creating a stack
DeleteStack	Deleting a stack
UpdateStack	Updating a stack
ExecuteStackAction	Executing a stack lifecycle action
CleanupResources	Cleaning a resource
UpdateTenantState	Freezing or unfreezing an account
GetBillingData	Generating billing data

4.2 Viewing Logs in CTS

When you enable CTS, operations performed on Application Orchestration Service (AOS) resources begin to be recorded. On the CTS console, you can query operation records from the last 7 days by performing the following operations.

Procedure

Step 1 Log in to the CTS console.

Step 2 In the left navigation pane, click **Trace List**.

Step 3 Filter the desired operation events.

The trace list supports four filter types:

- **Trace Source**, **Resource Type**, and **Search By**

  Select the search criteria from the drop-down lists. For example, select **AOS** from the **Trace Source** drop-down list box.

  From the **Search By** drop-down list, specify a trace name. From the **Search By** drop-down list, select or enter a specific resource ID. From the **Search By** drop-down list, select or enter a specific resource name.
- **Trace Status**: Select one of **All trace statuses**, **Normal**, **Warning**, and **Incident**.
- **Operator**: Select a specific operator (at the user level rather than the account level).
- **Start Date** and **End Date**: You can specify a time period to query traces.

Step 4 On the left of the to-be-queried record, click to view details.

Step 5 Click **View Trace** in the **Operation** column. On the displayed **View Trace** dialog box, the trace structure details are displayed.


```
{
    "service_type": "AOS",
    "user": {
        "domain": {
            "name": "***",
            "id": "6c389820d2fd46489c8987e5eb2675cc"
        },
        "id": "19652d0b0ff1407a9432b85b9e12f9eb",
        "name": "***"
    },
    "time": "2018/04/26 16:16:53 GMT+08:00",
    "code": 200,
    "resource_type": "AOS",
    "resource_name": "Stack",
    "resource_id": "19652d0b0ff1407a9432b85b9e12f9eb",
    "source_ip": "192.168.12.22",
    "trace_name": "PreviewStack",
    "trace_type": "ApiCall",
    "request": {},
    "api_version": "3.0.0",
    "message": "Preview stack successfully. Project id: 1e19d41bb1f24b5da4a98107607aac0f, stack name:
jhgdjh, template id: cea9ee29-3b39-f7be-d093-aff126b250e8, cluster id: . ",
    "record_time": "2018/04/26 16:16:53 GMT+08:00",
    "trace_id": "2da40c60-492a-11e8-a065-286ed488cbe3",
```



```
    "trace_status": "warning"
}
```

----End

