

函数 workflow

Sandbox 调试工具使用指南

文档版本

01

发布日期

2020-05-13



版权所有 © 华为技术有限公司 2020。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目录

1 Sandbox 介绍	1
2 安装与配置	3
2.1 准备.....	3
2.2 Linux/Mac 安装与配置.....	4
2.3 Windows 安装与配置.....	6
2.4 配置文件说明.....	7
3 快速入门	9
3.1 获取帮助.....	9
3.2 验证 HSAM 模板.....	10
3.3 生成本地事件.....	11
3.4 使用华为云事件源.....	19
3.5 Sandbox 远程部署.....	20
3.6 调用函数.....	21
3.7 本地 HTTP API.....	23
3.8 自动生成函数.....	24
4 Sandbox 应用示例	26
4.1 HSAM 模板参考.....	26
4.2 本地 HTTP API 示例.....	36
4.3 本地日志.....	37
4.4 使用环境变量.....	37
4.5 调试.....	38

1 Sandbox 介绍

在部署到华为云之前，可以使用华为云 Serverless Sandbox（HSS）在本地开发和测试 Serverless 应用。

Serverless 应用是由可以被多种事件触发的函数构成，能够触发华为云函数执行的事件包括：上传对象至对象存储服务（OBS）、从分布式队列服务（DMS）拉取的消息、从消息通知服务获取的消息等。

华为云 Serverless Sandbox（HSS）可以用来在本地测试函数功能，验证华为无服务器应用模型（HSAM），并为各种事件源本地生成样本有效载荷。

HSS 提供了 cloudEvent 命令，可以将来自华为云服务的事件直接路由到本地环境来调试本地函数功能。

特性概述

使用华为云 Serverless Sandbox（HSS）可以实现以下功能：

- 在本地主机上开发、测试函数功能。
- 验证华为无服务器应用模型（HSAM）。
- 使用本地事件文件调用函数。
- 调试功能函数。
- 提供交互式命令。
- 支持本地 HTTP API。
- 支持 Mac 系统。
- 支持远程部署。
- 支持 Windows 系统。

定义和缩写

词表如 [表 1-1](#) 所示。

表 1-1 词表

英文缩写	英文	中文
AK/SK	Access Key/Secret Key	访问密钥
AOS	Application Orchestration Service	应用编排服务
CES	Cloud Eye Service	云监控服务
CTS	Cloud Trace Service	云审计服务
DES	Data Express Service	数据快递服务
DMS	Distributed Message Service	分布式消息服务
HSAM	Huawei Serverless Application Model	华为无服务器应用模型
HSS	Huawei Serverless Sandbox	华为无服务器沙箱
OBS	Object Storage Service	对象存储服务
SMN	Simple Message Notification	消息通知服务
DIS	Data Ingestion Service	数据接入服务
APIG	API Gateway	API网关
LTS	Log Tank Service	云日志服务
YAML	YAML Ain' t Markup Language	一种以数据作为中心的标记语言

参考

更多相关资料参考：

[YAMLog](#)

[YAML Validator](#)

[JSON Validator](#)

2 安装与配置

2.1 准备

前提条件

运行HSS开发功能函数前，需要在本地主机上安装并运行Docker。HSS使用DOCKER_HOST环境变量联系Docker守护进程。

要验证Docker是否正常工作，可以在命令行界面输入Docker命令，例如：`docker ps`。

类似回显信息如下：

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS	PORTS	NAMES	

支持的操作系统

支持Linux/Mac、Windows系统。

支持的 Runtime

支持的运行时如下。

- Python 2.7
- Python 3.6
- Go 1.8.3 (Windows暂不支持)
- Java 8
- Node.js 6.10
- Node.js 8.10
- csharp2.0 (Windows暂不支持)
- csharp2.1 (Windows暂不支持)

产品下载

根据不同系统下载Sandbox工具包，如[表2-1](#)所示。

表 2-1 Sandbox 工具包

系统	下载	MD5校验码
Linux	Linux系统下载Sandbox工具包	0b538743657d2c966b22 aae1d1b09d3a
Mac	Mac系统下载Sandbox工具包	ff3744ad295a6a122d11f 9e64897a5bb
Windows	Windows系统下载Sandbox工具包	57f4739efe10aafc122dfc d845d1ce3a

2.2 Linux/Mac 安装与配置

这里以Linux为例

执行以下命令解压缩“[hss-1.0.10.linux-amd64.tgz](#)”文件，得到如表2-2所示文件。

```
$ tar zxvf hss-1.0.10.linux-amd64.tgz
```

表 2-2 文件说明

程序包	说明
fss-sandbox-csharp2.0-runtime.tar.gz	csharp 2.0运行时软件包
fss-sandbox-csharp2.1-runtime.tar.gz	csharp 2.1运行时软件包
fss-sandbox-go1.8-runtime.tar.gz	Golang 1.8运行时软件包
fss-sandbox-java8.0-runtime.tar.gz	Java 8.0运行时软件包
fss-sandbox-nodejs6.10-runtime.tar.gz	Node.js 6.10运行时软件包
fss-sandbox-nodejs8.10-runtime.tar.gz	Node.js 8.10运行时软件包
fss-sandbox-python2.7-runtime.tar.gz	Python 2.7运行时软件包
fss-sandbox-python3.6-runtime.tar.gz	Python 3.6运行时软件包
fss-sandbox-image.tar.gz	初始化镜像包
hss-1.0.10-package.tar.gz	HSS软件包
install.sh	可执行文件

安装 Sandbox

进入程序包存放目录，执行以下命令。

```
$ ./install.sh
```

显示如下类似回显信息时表示安装成功。

```
Please input the install directory[default: ./hss/]: hss will be installed into the default directory[/home/
paas/wny/hss] ./
./config.yaml
./hss
./test/
./test/sample-code/
./test/sample-code/python2.7/
./test/sample-code/python2.7/remote_pdb.py
./test/sample-code/python2.7/template.yml
./test/sample-code/python2.7/main.py
./test/sample-code/python2.7/evt.json
./test/sample-code/java8/
./test/sample-code/java8/mytest.jar
./test/sample-code/java8/template.yml
./test/sample-code/java8/evt.json
./test/sample-code/go1.8/
./test/sample-code/go1.8/function.so
./test/sample-code/go1.8/function.go
./test/sample-code/go1.8/template.yml
./test/sample-code/go1.8/evt.json
./test/sample-code/nodejs8.10/
./test/sample-code/nodejs8.10/index.js
./test/sample-code/nodejs8.10/template.yml
./test/sample-code/nodejs8.10/evt.json
./test/sample-code/python3.6/
./test/sample-code/python3.6/remote_pdb-1.2.0.dist-info/
./test/sample-code/python3.6/remote_pdb-1.2.0.dist-info/metadata.json
./test/sample-code/python3.6/remote_pdb-1.2.0.dist-info/METADATA
./test/sample-code/python3.6/remote_pdb-1.2.0.dist-info/entry_points.txt
./test/sample-code/python3.6/remote_pdb-1.2.0.dist-info/RECORD
./test/sample-code/python3.6/remote_pdb-1.2.0.dist-info/INSTALLER
./test/sample-code/python3.6/remote_pdb-1.2.0.dist-info/DESCRIPTION.rst
./test/sample-code/python3.6/remote_pdb-1.2.0.dist-info/top_level.txt
./test/sample-code/python3.6/remote_pdb-1.2.0.dist-info/WHEEL
./test/sample-code/python3.6/remote_pdb.py
./test/sample-code/python3.6/template.yml
./test/sample-code/python3.6/main.py
./test/sample-code/python3.6/evt.json
./test/sample-code/nodejs6.10/
./test/sample-code/nodejs6.10/index.js
./test/sample-code/nodejs6.10/template.yml
./test/sample-code/nodejs6.10/evt.json
./test/sample-code/csharp2.0/
./test/sample-code/csharp2.0/HC.Serverless.Function.Common.dll
./test/sample-code/csharp2.0/Newtonsoft.Json.dll
./test/sample-code/csharp2.0/TestExample.deps.json
./test/sample-code/csharp2.0/TestExample.pdb
./test/sample-code/csharp2.0/CSharpRuntime.Example.deps.json
./test/sample-code/csharp2.0/HC.Serverless.Function.Common.JsonSerializer.dll
./test/sample-code/csharp2.0/HC.Serverless.Function.Common.JsonSerializer.pdb
./test/sample-code/csharp2.0/template.yml
./test/sample-code/csharp2.0/CSharpRuntime.Example.dll
./test/sample-code/csharp2.0/CSharpRuntime.Example.runtimeconfig.json
./test/sample-code/csharp2.0/TestExample.dll
./test/sample-code/csharp2.0/TestExample.runtimeconfig.json
./test/sample-code/csharp2.1/
./test/sample-code/csharp2.1/HC.Serverless.Function.Common.dll
./test/sample-code/csharp2.1/Newtonsoft.Json.dll
./test/sample-code/csharp2.1/TestExample.deps.json
./test/sample-code/csharp2.1/HC.Serverless.Function.Common.JsonSerializer.dll
./test/sample-code/csharp2.1/template.yml
```



```
./test/sample-code/csharp2.1/TestExample.dll
./test/sample-code/csharp2.1/TestExample.runtimeconfig.json
./test/examples/
./test/examples/template.yml
./template/
./template/python2.7/
./template/python2.7/index.py
./template/java8/
./template/java8/mytest.jar
./template/go1.8/
./template/go1.8/index.so
./template/go1.8/index.go
./template/template.yml
./template/nodejs8.10/
./template/nodejs8.10/index.js
./template/python3.6/
./template/python3.6/index.py
./template/nodejs6.10/
./template/nodejs6.10/index.js
./README.md
fss-sandbox-image.tar
Load initial image successfully.
Install successfully.
```

设置环境变量

执行以下命令，将安装目录加到path变量，方便使用。

```
$ export PATH=$PATH:<安装目录>
```

2.3 Windows 安装与配置

前提条件

- 已部署安装并配置好Java环境，建议使用Java 1.8版本。
- 已部署安装并配置好Nodejs环境，建议使用Nodejs 6.10或8.10版本。
- 已部署安装并配置好Python环境，建议使用Python 2.7或3.6版本。

安装 Sandbox

执行以下命令解压缩“[hss-1.0.10.windows-amd64.tgz](#)”文件，得到如[表2-3](#)。

表 2-3 文件说明

程序包	说明
fss-sandbox-java-runtime.windows-amd64.tar.gz	Java运行时软件包
fss-sandbox-nodejs-runtime.windows-amd64.tar.gz	Node.js运行时软件包
fss-sandbox-python-runtime.windows-amd64.tar.gz	Python运行时软件包
hss-1.0.10-package.tar.gz	HSS软件包
install.bat	部署脚本

进入解压目录，双击执行“install.bat”，设置安装目录，完成文件解压安装。如下示例中安装目录为当前D:\temp\hss。

```
C:\Windows\system32\cmd.exe
"Please input the install directory[default: ./hss]: "
"hss will be installed into the default directory[D:\temp\hss]"
fss-sandbox-java-runtime.windows-amd64.tar.gz D:\temp\hss
fss-sandbox-nodejs-runtime.windows-amd64.tar.gz D:\temp\hss
fss-sandbox-python-runtime.windows-amd64.tar.gz D:\temp\hss
hss-1.0.0-package.tar.gz D:\temp\hss
"Install successfully."
请按任意键继续. . .
```

设置环境变量（可选）

- 步骤1** 安装完成后，点击右键选择“我的电脑”，点击“属性”，选择“高级系统设置”。
- 步骤2** 选择“高级”选项卡，点击“环境变量”。
- 步骤3** 在“系统变量”中设置2项属性：HSS_HOME，PATH。若已存在则点击“编辑”，不存在则点击“新建”。

变量名：HSS_HOME

变量值：D:\temp\hss\bin

在PATH末尾增加“%HSS_HOME%”，变量值需根据自己的实际路径配置。

- 步骤4** 打开命令提示符，输入“hss -h”查看相关命令详情，若能正常显示，则安装配置成功。

----结束

2.4 配置文件说明

HSS使用配置文件config.yaml，它指定了以下内容：

- Access Key and Secret Key
- Region, Project Name and Identifier
- IAM and DMS service endpoints

参数说明如下表2-4所示。

表 2-4 配置文件参数说明

参数	说明
AccessKey	访问密钥AK，获取方式请参考 管理访问密钥 。

参数	说明
SecretKey	访问密钥SK，获取方式请参考 管理访问密钥 。
Timeout	CLI命令超时时间，默认值30秒，最大值90秒 根据需要设置。
CurrentRegion	CLI当前对接服务所属默认Region 请参考 可用区 。
ProjectName	Regions成员，对接服务所属项目名称 请参考 查看项目ID 。
DomainName	Regions成员，对接服务所属项目域名 默认为myhuaweicloud.com

更改配置文命令

使用命令"update-config"可以更改config.yaml文件中的配置信息。命令格式如下：

```
update-config --rn <RegionName>
```

RegionName为当前区域名称

示例命令

```
update-config --rn southchina --ct 15
```

执行此命令将config.yaml文件中的超时间隔修改为15秒。

查看当前配置命令

使用命令"info"可以查看当前配置。命令格式如下：

```
info
```

更换区域命令

使用命令"switch"可以进行区域更换。命令格式如下：

```
switch <RegionName>
```

RegionName为要更换的区域名称。

示例命令

```
switch southchina
```

3 快速入门

3.1 获取帮助

获取帮助，命令如下。

```
$ hss --help
```

或执行hss进入交互模式，示例命令如下。

```
$ hss
```

输入help命令获取相应帮助。

类似回显信息如下：

```
NAME:
  hss - Use this app to parse, validate and deploy Huawei SAM (HSAM) files

USAGE:
  hss [global options] command [command options] [arguments...]

VERSION:
  1.0.10

COMMANDS:
  validate  Validate your HSAM YAML template file
  invoke    Invoke FunctionGraph function from template file.
  localEvent Generate FunctionGraph events (e.g. OBS, DMS, SMN etc.) locally. The events can be piped to
hss invoke command.
  cloudEvent Generate FunctionGraph events (e.g. OBS, DMS, SMN etc.) from Huawei Cloud. The events
can be pipe to hss invoke command.
  apigw     Allows you to run your functions locally for quick development and testing. When run in a
directory that contains your functions and HSAM template, it will start a local HTTP server hosting all of
your functions
  deploy    Deploy the specified template by creating a template and stack in AOS and excutes CREATE
lifecycle.
  create    new your fursion template file and template file.

GLOBAL OPTIONS:
  --help, -h      show help
  --version, -v   print the version
```

获取特定命令的帮助，命令如下。

```
$ hss <command> --help
```

或执行hss进入交互模式，命令如下：

\$ hss

获取特定命令的帮助，命令如下：

```
command> --help
```

类似回显信息如下：

```
NAME:
  hss validate - Validate your HSAM YAML template file

USAGE:
  hss validate [command options] [arguments...]

OPTIONS:
  --templateFile value, -t value      HSAM template file (default: "template.[yaml|yml]")
[$HSAM_TEMPLATE_FILE]
  --verbose, -v                       Optional: Provide verbose information
  --logFile value, -l value           Optional. Log file to redirect logs
```

3.2 验证 HSAM 模板

使用以下命令来验证函数HSAM模板文件。模板文件的名称使用-t或--templateFile选项指定。如果在该命令中未指定模板文件，则将验证当前工作目录中的默认模板文件template.yaml。此命令语法如下所示：

\$ hss validate

```
$ hss validate [-t (--template) <template-file>]
```

示例命令如下。

在当前工作目录中验证template.yaml

```
$ hss validate
```

类似回显信息如下：

```
2018/03/15 20:10:54 SUCCESS - Template: template.yml successfully parsed.
```

模板路径： /root/hss/test/sample-code/python2.7/template.yml

```
$ hss validate -t /root/hss/test/sample-code/python2.7/template.yml
```

类似回显信息如下：

```
2018/03/15 20:14:02 Configuring Default Values
2018/03/15 20:14:02 SUCCESS - Template: /root/hss/test/sample-code/python2.7/template.yml
successfully parsed.
```

或执行hss进入交互模式，命令如下：

\$ hss

在当前工作目录中验证template.yaml，命令如下：

```
> validate
```

类似回显信息如下：

```
2018/08/06 20:22:47 INFO: Template: template.yml, successfully parsed.
```

模板路径： /root/hss/test/sample-code/python2.7/template.yml

```
$ validate -t /root/hss/test/sample-code/python2.7/template.yml
```

类似回显信息如下：

```
2018/08/06 20:25:20 INFO: Template:: /root/hss/test/sample-code/python2.7/template.yml, successfully parsed.
```

3.3 生成本地事件

使用以下命令生成模拟JSON事件记录。这些记录可以在调用函数时使用。

```
$ hss localEvent <service> [service-options]
```

HSS本地事件命令的输出可以保存在本地文件中，然后可以在HSS调用命令中使用该文件，如下所示。

```
$ hss localEvent obs --bucket bucket1 > events.json  
$ hss invoke -e events.json <FunctionStage>
```

HSS本地事件命令的输出可以直接传送给HSS调用命令，如下所示。

```
$ hss localEvent obs --bucket bucket2 | hss invoke <FunctionStage>
```

如果使用来自非stdin输入指定事件文件，它将优先于来自stdin的输入。

由于交互模式hss相关命令不支持|，所以该命令可替换成以下。

使用localEvent -o参数生成本地文件。

```
> localEvent obs --bucket bucket2 -o <object file>
```

使用invoke -e参数执行本地文件。

```
> Invoke -e <object file> <FunctionStage>
```

选项说明

该命令可以用于从华为云服务（如OBS，SMN和DMS）中生成典型事件。<service>可以是以下之一。

- obs：生成华为云OBS事件
- smn：生成华为云SMN事件
- dms：生成华为云DMS事件
- dis：生成华为云DIS事件
- apig：生成华为云APIG事件
- lts：生成华为云LTS事件
- cts：生成华为云CTS事件

表 3-1 OBS options

options	说明	是否必选
--region, -r	OBS region	可选 缺省值：cn-north-1
--bucket, -b	OBS bucket name	可选 缺省值： example-bucket

options	说明	是否必选
--key, -k	OBS key name	可选 缺省值: example/key
--objectFile value, -o	logged to the objectFile	可选

表 3-2 SMN options

options	说明	是否必选
--message, -m	SMN message	可选 缺省值: <default message>
--subject, -s	SMN subject	可选 缺省值:<default subject>
--topic, -t	SMN topic	可选 缺省值:<default topic>
--objectFile value, -o	logged to the objectFile	可选

表 3-3 DMS options

options	说明	是否必选
--region, -r	DMS region	可选 缺省值: cn-north-1
--queueId, -qid	DMS queue id	可选 缺省值: <default QID>
--consumerGroupId, -gid	DMS consumer group id	可选 缺省值: <default CGID>
--messages, -m	DMS message	可选 缺省值: <default messages>
--objectFile value, -o	logged to the objectFile	可选

表 3-4 DIS options

options	说明	是否必选
--stream value, -s	DIS stream	可选 缺省值: mystream
--tag value, -t	DIS position of reading the data	可选 缺省值: LATEST
--records value, -r	DIS data in JSON	可选 缺省值: [{"partition_key": "\", "shardId_0000000000\", \"data\": \"d2VsY29tZQ==\", \"sequence_number\": \"0\"}"]
--nextPartitionCursor value, --npc	The next partition cursor	可选 缺省值如“说明”中所示。
--verbose, -v	Provide verbose information	可选
--objectFile value, -o	logged to the objectFile	可选

 说明

--nextPartitionCursor value, --npc缺省值如下:

```
eyJnZXRjdGVyYXRvclBhcmFtIjp7InN0cmVhbS1uYW1lIjoizGZlLXN3dGVzdCIsInBhcnRpdGlvbi1pZCI6InNoYXJkSWQtMDAwMDAwMDAwMCIsmN1cnNvci10eXBlljoifVFJTUV9IT1JJWk90Iiwic3Rhcnc2VxdWVud2UtbmVtYmVyljoicj9LCjZlcmF0ZVRpbWVzdGFtcCI6MTUwOTYwNmM5MjE5MX0
```

表 3-5 APIG options

options	说明	是否必选
--method value, -m	The http request method	可选 缺省值: GET
--path value, -p	The http request path	可选 缺省值: /mytest
--body value, -b	The http request body	可选
--resptype value, -t	The http response type	可选 缺省值: html

options	说明	是否必选
--stage value, -s	The release stage	可选 缺省值: RELEASE
--host value, -a	The release address of host	可选 缺省值: 50eedf92-c9ad-4ac0-827e-d7c11415d4f1.apigw.cn-north-1.huaweicloud.com
--objectFile value, -o	logged to the objectFile	可选

表 3-6 LTS options

options	说明	是否必选
--data value, -d	data of logs encoded by base64	可选 缺省值如“说明”中所示。
--objectFile value, -o	logged to the objectFile	可选

options	说明	是否必选
--request value, --rq	Event request content in JSON	可选 缺省值: {}
--response value, --rs	.Event response content in JSON	可选 缺省值: {}
--code value, -c	Event response code	可选 缺省值: 204
--serviceType value, --st	Sender service	可选 缺省值: FunctionGraph
--resourceType value, --rty	Sender resource type	可选 缺省值: graph
--resourceName value, --rn	Sender resource name	可选 缺省值: workflow-2be1
--resourceId value, --ri	Sender resource id	可选 缺省值: <default resource id>
--traceName value, --tn	Track record name	可选 缺省值: deleteGraph
--traceType value, --tt	Track record type	可选 缺省值: ConsoleAction
--recordTime value, --rtm	Time when tracking records	可选 缺省值: 2018/08/01 08:54:07 GMT+08:00
--traceId value, --ti	Track record id	可选 缺省值: <default traceId>
--traceStatus value, --ts	Track record status	可选 缺省值: normal
--objectFile value, -o	logged to the objectFile	可选

事件详情

- DMS Record


```

{
  "QueueID": "39907254-5321-4b90-bad2-f82455ff3ead",
  "Region": "cn-north-1",
  "Messages": [
    {
      "Body": "this is body message",
      "Attributes": {

```

```
        "weight": "80",
        "size": "100"
      }
    ],
    "EventType": "MessageCreated",
    "ConsumerGroupID": "g-06c2bff5-67db-4055-ba4e-a31bb09c3485"
  }
}
```

- SMN Record

```
{
  "record": [
    {
      "event_version": "1.0",
      "smn": {
        "topic_urn": "urn:smn:cn-north-1:0162c0f220284698b77a3d264376343a:test3",
        "timestamp": "2018-01-09T07:11:40Z",
        "message_attributes": null,
        "message": "this is smn message content",
        "type": "notification",
        "message_id": "a51671f77d4a479cacb09e2cd591a983",
        "subject": "this is smn message subject"
      },
      "event_subscription_urn":
        "urn:fss:cn-north-1:0162c0f220284698b77a3d264376343a:function:default:
          read-smn-message:latest",
      "event_source": "smn"
    }
  ]
}
```

- DIS Record

```
{
  "ShardID": "shardId-0000000000",
  "Message": {
    "next_partition_cursor":
      "eyJnZXRRJdGVyYXRvclBhcmFtljpw7InN0cmVhbS1uYW1lIjoiZGZlLXN3dGVzdClInBhcnRpdGlvbi
      1pZCI6InNoYXJkSWQtMDAwMDAwMDAwMCIImN1cnNvci10eXBlljoiVFJTV9IT1JJWk9OIwi
      c3RhcncRpbmctc2VxdWVvY2UtbnVtYmVyljoiNCJ9LCJnZW5lcmF0ZVRpbWVzdGFtcCI6MTUw
      OTYwNjM5MjE5MX0",
    "records": [{"partition_key": "shardId_0000000000", "data": "d2VsY29tZQ=="},
    "sequence_number": "0"},
    "millis_behind_latest": 0
  },
  "Tag": "LATEST",
  "StreamName": "mystream"
}
```

- APIG Record

```
{
  "body": "",
  "requestContext": {
    "apiId": "bc1dcffd-aa35-474d-897c-d53425a4c08e",
    "requestId": "11cdcdf33949dc6d722640a13091c77",
    "stage": "RELEASE"
  },
  "queryStringParameters": {
    "responseType": "html"
  },
  "httpMethod": "GET",
  "pathParameters": {},
  "headers": {
    "accept-language": "zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2",

```

```
"accept-encoding": "gzip, deflate, br",
"x-forwarded-port": "443",
"x-forwarded-for": "127.0.0.1",
"accept": "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8",
"upgrade-insecure-requests": "1",
"host": "50eedf92-c9ad-4ac0-827e-d7c11415d4f1.apigw.cn-
north-1.huaweicloud.com",
"x-forwarded-proto": "",
"pragma": "no-cache",
"cache-control": "no-cache",
"x-real-ip": "127.0.0.1",
"user-agent": "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:57.0) Gecko/20100101
Firefox/57.0"
},
"path": "/mytest",
"isBase64Encoded": true
}
```

- LTS Record

```
{
  "lts": {
    "data": {
      "logs": [{
        "message": "2018-06-26/18:40:53 [INF] [config.go:72] Successfully loaded
general configuration file\\r\\n",
        "time": "1530009653059",
        "host_name": "ecs-testagent.novalocal",
        "ip": "192.168.1.98",
        "path": "usr/local/telescope/log/common.log",
        "log_uid": "663d6930-792d-11e8-8b08-286ed488ce70",
        "line_no": 615
      },
    ],
    "owner": "6280e170bd934f60a4d851cf5ca05129",
    "log_group_id": "97a9d284-4448-11e8-8fa4-286ed488ce70",
    "log_topic_id": "1a9675a7-784d-11e8-9f70-286ed488ce70"
  }
}
```

- CTS Record

```
{
  "cts": {
    "time": "2018/08/01 08:54:07 GMT+08:00",
    "user": {
      "name": "userName",
      "id": "5b726c4bfd84821ba866bafaaf56aax",
      "domain": {
        "name": "domainName",
        "id": "b2b3853af40448fcb9e40dxj89505ba"
      }
    },
    "request": {},
    "response": {},
    "code": 204,
    "service_type": "FunctionGraph",
    "resource_type": "graph",
    "resource_name": "workflow-2be1",
    "resource_id": "urn:fgs:cn-
north-1:2d1d891d93054bbaa69b9e866c0971ac:graph:workflow-2be1",
    "trace_name": "deleteGraph",
    "trace_type": "ConsoleAction",
    "record_time": "2018/08/01 08:54:07 GMT+08:00",
  }
}
```

```

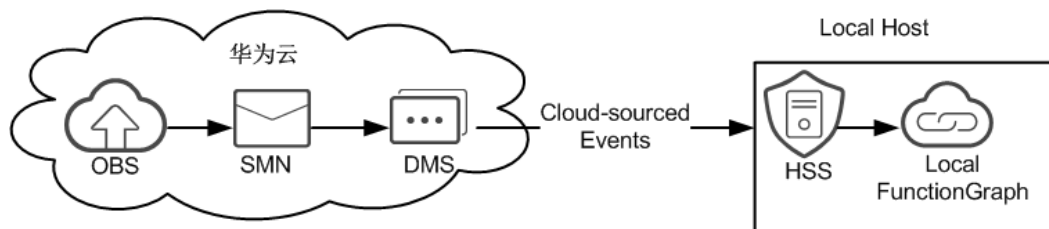
"trace_id": "69be64a7-0233-11e8-82e4-e5d37911193e",
"trace_status": "normal"
}
}

```

3.4 使用华为云事件源

HSS云事件命令将从华为云服务（例如OBS）获取事件。这是使用DMS队列完成的。必须配置华为云服务才能将事件发送到DMS队列。HSS将轮询来自DMS队列的事件，然后将事件发送到标准输出，如图3-1所示。

图 3-1 华为云事件



该命令的语法如下。

\$ hss cloudEvent [options]

可以将HSS cloudEvent命令的stdout事件输出到本地文件，然后可以在HSS invoke命令中使用该文件，如下以dms为例。

```

$ hss cloudEvent dms -qid <qid> -gid <gid> > event.json
$ hss invoke -e event.json <FunctionStage>

```

或执行hss进入交互模式，命令如下：

```

cloudEvent dms -qid <qid> -gid <gid> -o event.json

```

非交互模式下HSS cloudEvent事件命令的输出可以直接传送给HSS调用命令，如下所示。

```

$ hss cloudEvent dms -qid <qid> -gid <gid> | hss invoke <FunctionStage>

```

表 3-8 Options

options	说明	是否必选
--interval, -i	Time interval to poll DMS queue. Range: 1-10 seconds. Default = 2 seconds.	可选
--queueId, -qid	DMS queue id	必选
--consumerGroupId, -gid	DMS consumer group id	必选

必须配置华为云认证，并配置华为云服务才能将事件路由到DMS队列。

通过华为云认证后，获取相应的ak/sk、ProjectId等，配置到config.yaml文件中。

当HSS cloudEvent命令被调用时，HSS每隔一秒钟开始轮询华为云中的DMS队列，间隔时间为1到10秒。当在DMS队列中找到事件时，它将被检索并发送到stdout。

3.5 Sandbox 远程部署

命令介绍

Sandbox远程部署功能依赖应用编排服务AOS，在使用sandbox的远程部署功能前，需开通[AOS服务](#)。使用以下命令将函数HSAM模板定义的云资源通过应用编排服务AOS远程部署至云平台。

```
$ hss deploy [options]
```

表 3-9 hss deploy 命令 Options 说明

Options	说明	是否必选
-- templateFile, -t	HSAM模板文件的路径	必选 缺省值: template.yml
-- obsBucket, -b	若代码直接在HSAM模板中编辑，该参数无需配置。若代码上传方式为“obs”时，需将本地代码压缩包上传至OBS桶。若未配置该参数，将创建一个默认名称的OBS桶；若指定的OBS桶不存在，需用户确认是否预先创建该OBS桶。	可选 缺省值: sandbox-{region}-{projectid}
-- obsPrefix, -p	上传至OBS桶的代码压缩包名称的前缀	可选 缺省值: HSAM模板中的函数名称
--stackName	上传至AOS平台的堆栈名称	必选
--package, -g	分步操作标志，package转换成AOS模板，打包zip并上传到OBS服务	可选
--aosTemplateFile, -o	转换成AOS模板名称	可选 缺省值: 本地AOS模板名称aosTemplateFile.yaml
-- templateName, --tn	上传至AOS平台的模板名称	可选 缺省值: 本地AOS模板名称
-- templateDescription, --td	上传至AOS平台的模板描述信息	可选 缺省值: 空

Options	说明	是否必选
-- stackDescription, --sd	待部署的堆栈描述信息	可选 缺省值：空
--verbose, -v	Provide verbose information	可选

Sandbox远程部署包括：

1. 使用**hss deploy -g deploy**命令将函数代码打包上传至OBS桶，并将函数HSAM模板转化为AOS模板。
2. 使用**hss deploy -g deploy**命令将本地AOS模板上传至AOS服务平台，根据上传的AOS模板部署堆栈。
3. 使用**hss deploy**命令将函数代码打包上传至OBS桶，函数HSAM模板转化为AOS模板，并将本地AOS模板上传至AOS服务平台，根据上传的AOS模板部署堆栈。

Sandbox与AOS服务结合实现本地调试和远程部署资源流程如下：

1. 本地编写函数代码和函数HSAM模板。
2. 验证HSAM模板。
3. 生成本地事件（云事件）。
4. 本地调试函数。
5. 上传代码压缩包至OBS桶，将HSAM模板转化为AOS模板。
6. 上传本地AOS模板，部署堆栈。

示例命令

使用上述命令将代码压缩包上传至指定的OBS桶以及本地生成AOS模板后，将本地AOS模板上传至AOS平台，并根据上传的AOS模板部署堆栈。命令如下：

```
$ hss deploy -t template.yml -s sandbox --tn blueprint-sandbox --tv 1.0
```

上述命令将当前目录下的AOS模板上传至AOS平台，并根据上传的AOS模板部署堆栈。

3.6 调用函数

命令介绍

使用以下命令调用函数

```
$ hss invoke [options] [<FunctionStage>]
```

Options说明如[表3-10](#)所示。

表 3-10 Options 说明

Options	说明	是否必选
--templateFile, -t	模板文件的路径	必选 缺省值: template.yml
--debugPort, -d	调试端口号	可选
--eventFile, -e	用于模拟事件的JSON事件文件	可选
--logFile, -l	日志文件名	可选
--envFile, -n	环境变量文件名	可选
--verbose, -v	提供详细的调试信息	可选

参数说明如表3-11所示。

表 3-11 参数说明

参数	说明	是否必选
FunctionStage	函数资源名称	必选

📖 说明

如果模板文件包含多个FunctionStage，则必须指定函数的名称，否则命令将终止并显示错误。

示例命令

如果未指定模板文件，HSS将在当前工作目录中使用默认的HSAM模板文件 template.yml。如果模板文件只有一个FunctionGraph函数资源，将调用该FunctionGraph下的函数。可以使用事件文件调用函数，并将输出发送到stdout。命令如下。

```
$ hss invoke -e evt.json
```

非交互模式下可以通过从stdin输入事件来调用函数，交互模式不支持hss命令带逻辑运算符，命令如下。

```
$ echo '{"keyword1": "value2"}' | hss invoke MyFuncStage1
```

如果HSS调用没有事件文件并且没有stdin值，那么它将等待你输入一个事件字符串。可以单击'Enter'来调用没有输入的函数，也可以使用显式模板文件调用函数。命令如下。

```
$ hss invoke -e event.json -t template.yml MyFuncStage1
```

3.7 本地 HTTP API

命令介绍

使用以下命令调用函数

```
$ hss invoke [options] [<FunctionStage>]
```

Options说明如表3-12所示。

表 3-12 Options 说明

Options	说明	是否必选
--templateFile, -t	模板文件的路径	必选 缺省值: template.yml
--eventFile, -e	用于模拟事件的JSON事件文件	可选
--host -H	监听主机ip	可选
--port -P	监听端口号	可选
--logFile, -l	日志文件名	可选
--verbose, -v	提供详细的调试信息	可选

参数说明如表3-13所示。

表 3-13 参数说明

参数	说明	是否必选
FunctionStage	函数资源名称	可选

示例命令

如果未指定模板文件，HSS将在当前工作目录中使用默认的HSAM模板文件 template.yml。如果模板文件只有一个FunctionGraph函数资源，将调用该FunctionGraph下的函数。可以使用事件文件调用函数，并将输出发送到stdout。命令如下。

```
$ hss apigw -e evt.json
```

类似回显信息如下：

```
2018/08/23 16:19:49 INFO: Template: template.yml, successfully parsed
2018/08/23 16:19:49 INFO: Please curl http://0.0.0.0:9999/MyFuncStage2 to execution function MyFuncStage2
2018/08/23 16:19:49 INFO: Please curl http://0.0.0.0:9999/MyFuncStage1 to execution function
```

```
MyFuncStage1  
2018/08/23 16:19:49 INFO: Please provide a FunctionStage name shown above as the first agument.
```

在新的shell窗口执行curl http://0.0.0.0:9999/MyFuncStage1调用函数MyFuncStage1，0.0.0.0请替换成当前可用的主机ip。

执行成功返回结果如下：

```
2018/08/23 16:19:49 INFO: Function Return: Hello message: {"name": "TONY"}, code: 200  
2018/08/23 16:19:49 http: multiple response.WriteHeader calls
```

3.8 自动生成函数

命令介绍

使用以下命令可以自动生成函数

```
$ hss create [options]
```

Options说明如表3-14所示。

表 3-14 Options 说明

Options	说明	是否必选	参数类型
Functionname, --fn	函数名称	是	String
description, -d	描述 默认值:Automatically generated with HSS	否	String
runtime, --rt	运行时语言 默认Node.js6.10 [Python2.7,Python3.6,No de.js6.10,Node.js8.10,Java 8,Go1.8]	是	String
handler, --hd	函数执行入口 默认值:index.handler	否	String

📖 说明

Java和Go函数的执行入口为默认值，自定义创建不生效。Java函数默认值为：
com.huawei.cff.MyTest.handler，Go函数默认值为：index.Handler。

示例命令

租户输入函数生成命令，hss根据命令要求生成对应的函数文件以及模板，以创建“HelloWorld”函数为例，命令如下：

```
$hss> create --fn HelloWorld
```

类似回显信息如下：

```
$hss> create --fn HelloWorld
index.js
2018/11/24 15:12:43 INFO: the code path is :/root/hwz/src/sandbox/build/release/code/
HelloWorld/
$hss> cd /root/hwz/src/sandbox/build/release/code/HelloWorld
/root/hwz/src/sandbox/build/release/code/HelloWorld
$hss> invoke -t HelloWorld.yml
2018/11/24 15:14:58 INFO: Reading events from stdin since eventFile not provided:
{"test":"sandbox create"}
2018/11/24 15:16:36 INFO: Template: HelloWorld.yml, successfully parsed
2018/11/24 15:16:36 INFO: EnvVars: map[RUNTIME_FUNC_NAME:handler
RUNTIME_FUNC_VERSION: RUNTIME_HANDLER:index.handler RUNTIME_CPU: SecretKey:
RegionName: RUNTIME_PROJECT_ID: ProjectName: RUNTIME_PACKAGE: DEBUG_PORT:
AccessKey: RUNTIME_MEMORY:128 RUNTIME_TIMEOUT:3 RUNTIME_USERDATA:{}]
2018/11/24 15:16:36 INFO: Container created, cout:
[487048bf00f4e73add732efb78031d0f154e58bae448c903b64173ec2246e033]
2018/11/24 15:16:36 INFO: Results: {"test":"sandbox
create","functionname":"test","timestamp":"Sat Nov 24 2018 07:16:36 GMT+0000 (UTC)"} code:
200
$hss> ls
HelloWorld.yml
index.js
$hss> cat index.js
exports.handler = function (event, context, callback) {
  event.functionname = 'test';
  event.requestId = context.requestId;
  event.timestamp = (new Date()).toString();

  console.log('event:', JSON.stringify(event));
  console.log('ak:', context.getAccessKey());
  console.log('sk:', context.getSecretKey());
  callback(null, JSON.stringify(event));
}
```

4 Sandbox 应用示例

4.1 HSAM 模板参考

介绍

HSAM是用于定义华为云serverless应用的模型。

Serverless应用是基于事件驱动的全托管应用。一个典型的serverless应用由一个或多个华为云函数组成。这些云函数可以独立使用，也可以结合其他华为云资源产生触发事件，触发云函数执行。如向华为云OBS上传对象，在华为云SMN上生产消息以及使用华为云API网关调用API等产生触发事件触发函数执行。

HSAM为应用程序管理员提供一种描述Serverless应用业务逻辑的描述语言模型。结合华为云应用编排服务AOS，在华为云中编排和部署Serverless应用所需的资源。

函数 workflow 本地调试工具HSS（Huawei Serverless Sandbox）使用HSAM在本地开发和测试用户自定义的serverless应用。利用本地调试工具HSS的远程部署功能，应用编排服务AOS可以根据HSAM自动将serverless应用部署到华为云。

规范

格式

华为云HSAM采用YAML格式的模板文件来描述serverless应用。

HSAM定义的serverless应用资源包括：Events、Functions、FunctionStages

其中：

- Event和Function资源可定义成独立的资源，FunctionStage资源关联一个Function资源和若干Events资源。
- HSAM支持的Events包括：OBS、DMS、SMN、DIS、LTS、CTS、APIG、TIMER。

模板文件必须在文档根节点中包含SAMversion: 'v1'部分，模板文件包含下面两种类型：Resource类型和事件源类型

示例：华为云HSAM模板

```
``yaml
SAMversion: 'v1'
```

```
Resources:
MyFunc1: # Function resource name
Type: 'HC::Serverless::Function'
Properties:
  FuncName: testFunc1
  Description: "This is my Func1"
  Handler: index.handler
  Runtime: Python2.7
  MemorySize: 128
  Timeout: 180
  CodeUri: ""
  Code: |
    # -*- coding:utf-8 -*-
    import json
    def handler(event, context):
        output = 'Hello message: ' + json.dumps(event)
        return output
Xrole: allService
Environment:
  Variables:
    key1: value1
    key2: value2
ObsEvt1: # OBS event resource name
Type: OBS
Properties:
  Bucket: my-photo-bucket
  EventType:
    - ObjectCreated:*
    - ObjectRemoved:*
  Prefix: fss
  Suffix: jpg
SmnEvt: # SMN event resource name
Type: SMN
Properties:
  Topic: topic1
DmsEvt: # DMS event resource name
Type: DMS
Properties:
  ConsumerGroupId: g-92777f57-a42a-401e-829d-4151417c639f
  QueueId: a99e1c82-808a-4d3b-b03a-0e161e514c2c
LtsEvt: # LTS event resource name
Type: LTS
Properties:
  GroupId: fd537073-7ffb-11e8-b63a-286ed488ce71
  GroupName: test-log-group-1
  TopicId: 51acbef6-aa92-11e8-810a-286ed488ce71
  TopicName: lts-topic-aos
CtsEvt: # CTS event resource name
Type: CTS
Properties:
  Name: ctstest
Operations:
  AOS:
    stack:
      - createStack
      - deleteStack
    quota:
      - updateQuota
CCE(1.0):
  app:
    - createApp
    - deleteApp
```

```
    image:
      - deleteImage
  DisEvt: # DIS event resource name
  Type: DIS
  Properties:
    Stream: dis-test
    BatchSize: 100
    ShardIteratorType: TRIM_HORIZON
    PollingInterval: 30
  ApiEvt: # APIG event resource name
  Type: APIG
  Properties:
    ApiName: testApi
    ApiGroup: my_apigroup
    Auth: IAM
    Protocol: HTTPS
    Timeout: 5000
  TimerEvt: # TIMER event resource name
  Type: TIMER
  Properties:
    Name: my_timer
    Schedule: 3m
    ScheduleType: Rate
    UserEvent: "user data"
  MyFuncStage1: # FunctionStage resource name
  Type: 'HC::Serverless::FunctionStage'
  Properties:
    Function: MyFunc1
  Events:
    TimerEvt
    LtsEvt
    CtsEvt
    ApiEvt
  ...
```

📖 说明

HSAM中的所有属性名称都区分大小写。

Resource 类型

- HC::Serverless::Function
- HC::Serverless::FunctionStage

HC::Serverless::Function

创建serverless应用的云函数资源。

表 4-1 属性

属性名称	类型	描述
FuncName	string	必填 函数的名称

属性名称	类型	描述
Handler	string	必填 函数的执行入口。规则：xx.xx，若代码采用在线编辑方式，执行入口可自定义；若采用OBS上传文件方式，函数的执行入口由本地代码文件名称+入口函数名称组成。举例：本地代码文件main.py，入口函数名称为handler，则Handler：main.handler。
Runtime	string	必填 运行时环境。可选值为：Node.js6.10、Node.js8.10、Python2.7、Python3.6、Java8，Go1.8，csharp2.0、csharp2.1。
CodeUri	string	OBS上传文件方式时必填 本地代码文件位置
Code	string	在线编辑方式时必填 函数源码
Description	string	可选 函数的描述
Role	string	可选 为函数配置 委托 。
MemorySize	integer	可选 每次函数执行分配的内存大小，单位是MB，有效值为128、256、512、768、1024、1280、1536、1792、2048、2560、3072、3584、4096，默认为128（MB）。
Timeout	integer	可选 函数在被终止之前可以运行的最长时间，单位是秒，范围3~900秒，默认为3秒。
Environment	环境变量对象	可选 为函数配置 环境变量 。

示例：HC::Serverless::Function

```

`yaml
MyFunc1: # Function resource name
  Type: 'HC::Serverless::Function'
  Properties:
    FuncName: testFunc1
    Description: "This is my Func1"
    Handler: index.handler
    Runtime: Python2.7
    MemorySize: 128
    Timeout: 180
    Code: |
      # -*- coding:utf-8 -*-
      import json
  
```



```
def handler (event, context):
    output = 'Hello message: ' + json.dumps(event)
    return output
Xrole: allService
Environment:
Variables:
    key1: value1
    key2: value2
...
```

HC::Serverless::FunctionStage

创建FunctionStage资源。FunctionStage资源包含一个函数和若干触发器，也可只包含一个函数。该资源描述函数与触发器的编排关系。

表 4-2 属性

属性名称	类型	描述
Function	string	必填 Function resource名称，关联一个函数。
Events	Events	可选 Event resource列表，关联该函数下的一组触发器。如果未配置Events属性，则表示函数下没有配置触发器。

示例：HC::Serverless::FunctionStage

```
```yaml
MyFunc1: # Function resource name
 Type: 'HC::Serverless::Function'
 Properties:
 FuncName: testFunc1
 Description: "This is my Func1"
 Handler: index.handler
 Runtime: Python2.7
 MemorySize: 128
 Timeout: 180
 Code: |
 # -*- coding:utf-8 -*-
 import json
 def handler (event, context):
 output = 'Hello message: ' + json.dumps(event)
 return output
 Xrole: allService
 Environment:
 Variables:
 key1: value1
 key2: value2
SmnEvt: # SMN event resource name
 Type: SMN
 Properties:
 Topic: topic1
MyFuncStage1: # FunctionStage resource name
 Type: 'HC::Serverless::FunctionStage'
 Properties:
 Function: MyFunc1
 Events:
```

```
SmnEvt
```

## 事件源类型

- **TIMER**  
描述类型为定时触发器的对象。

表 4-3 属性

属性名称	类型	描述
Name	string	必填 定时触发器的名称。支持英文，数字，下划线，且只能以英文字母开头，长度不能超过64个字符。
Schedule	string	必填 定时触发规则。触发规则详情请参考： <a href="#">使用定时触发器</a> 。
Schedule Type	string	必填 定时触发规则类型，支持Rate、Cron两种触发规则类型。
Status	Status	可选 是否启用定时触发器。有效值为DISABLED，ACTIVE，默认启动定时触发器。
UserEvent	string	可选 附加信息，用户可以自定义。当输入附加信息时，Timer触发器触发函数执行时，执行事件中将包含附加的信息，附加信息小于2KB

示例：TIMER事件源对象

```
```yaml
TimerEvt:
  Type: TIMER
  Properties:
    Name: my_timer
    Schedule: 3m
    ScheduleType: Rate
    UserEvent: "user data"
```
```

- **APIG**  
描述类型为APIG触发器的对象。

表 4-4 属性

| 属性名称     | 类型      | 描述                                                      |
|----------|---------|---------------------------------------------------------|
| ApiName  | string  | 必填<br>API的名称                                            |
| APIGroup | string  | 必填<br>API分组的名称                                          |
| Auth     | string  | 可选<br>安全认证类型，支持IAM、APP和NONE三种安全认证类型，若不配置该属性默认采用IAM安全认证。 |
| Protocol | string  | 可选<br>请求协议，支持HTTP、HTTPS两种请求协议。若不配置该属性默认采用HTTPS请求协议。     |
| Timeout  | integer | 可选<br>后端超时(毫秒)，超时时间范围为1~60000毫秒。若不配置该属性默认设置超时时间为5000毫秒。 |

示例：APIG事件源对象

```
``yaml
ApiEvt:
 Type: APIG # trigger type
 Properties:
 ApiName: testApi
 ApiGroup: my_apigroup
 Auth: IAM
 Protocol: HTTPS
 Timeout: 5000
````
```

- **OBS**
描述类型为OBS触发器的对象。

表 4-5 属性

| 属性名称 | 类型 | 描述 |
|-----------|----------|---|
| Bucket | string | 必填
OBS桶的名称。 |
| EventType | string列表 | 必填
OBS触发事件列表。有效值为["ObjectCreated:", "ObjectCreated:Put", "ObjectCreated:Post", "ObjectCreated:Copy", "ObjectCreated:CompleteMultipartUpload", "ObjectRemoved:", "ObjectRemoved:Delete", "ObjectRemoved:DeleteMarkerCreated"]'. |

| 属性名称 | 类型 | 描述 |
|--------|--------|----------------------------------|
| Prefix | string | 必填
前缀过滤配置。用来限制对以此关键字开头的对象的通知。 |
| Suffix | string | 必填
后缀过滤配置。用来限制对以此关键字结尾的对象的通知。 |

示例：OBS事件源对象

```
```yaml
ObsEvt:
 Type: OBS
 Properties:
 Bucket: my-photo-bucket
 EventType:
 - ObjectCreated:*
 - ObjectRemoved:*
 Prefix: fss
 Suffix: jpg
```
```

- **SMN**

描述类型为SMN触发器的对象。

表 4-6 属性

| 属性名称 | 类型 | 描述 |
|--------|--------|------------------------------------|
| Topic | string | 必填
订阅的SMN主题名称。该属性从已创建的SMN主题中获取。 |
| Remark | string | 可选
订阅备注。当该属性未配置时，采用默认值APItest。 |

示例：SMN事件源对象

```
```yaml
SmnEvt:
 Type: SMN
 Properties:
 Topic: topic1
```
```

- **DMS**

描述类型为DMS触发器的对象。

表 4-7 属性

| 属性名称 | 类型 | 描述 |
|-----------------|-----------------|--|
| ConsumerGroupId | ConsumerGroupId | 必填
ConsumerGroupId |
| QueueId | string | 必填
DMS队列ID。该属性从已创建的DMS队列中获取。 |
| PollingInterval | integer | 可选
拉取周期为DMS触发器轮询消息的间隔时间。当该属性未配置时，采用默认值30。 |

示例：DMS事件源对象

```
```yaml
DmsEvt:
 Type: DMS
 Properties:
 ConsumerGroupId: g-92777f57-a42a-401e-829d-4151417c639f
 QueueId: a99e1c82-808a-4d3b-b03a-0e161e514c2c
 PollingInterval: 30
```
```

- **DIS**
描述类型为DIS触发器的对象。

表 4-8 属性

| 属性名称 | 类型 | 描述 |
|-------------------|---------|--|
| Stream | string | 必填
通道名称。该属性从已创建的DIS通道中获取。 |
| BatchSize | integer | 可选
批处理大小，有效范围为1到10000。当该属性未配置时，采用默认值100。 |
| ShardIteratorType | string | 可选
在流中开始读取数据的位置，支持TRIM_HORIZON，LATEST。当该属性未配置时，采用默认值TRIM_HORIZON。 |
| PollingInterval | integer | 可选
拉取流数据的周期(秒)，有效范围为1到60秒。当该属性未配置时，采用默认值30。 |

示例：DIS事件源对象

```
```yaml
DisEvt:
 Type: DIS
```
```

```
Properties:
  Stream: dis-test
  BatchSize: 100
  ShardIteratorType: TRIM_HORIZON
  PollingInterval: 30
...`
```

- **LTS**
描述类型为LTS触发器的对象。

表 4-9 属性

| 属性名称 | 类型 | 描述 |
|-----------|--------|-------------------------------------|
| GroupId | string | 必填
LTS日志组ID。该属性从已创建的LTS日志组中获取。 |
| GroupName | string | 必填
LTS日志组名称。该属性从已创建的LTS日志组中获取。 |
| TopicId | string | 必填
LTS日志主题ID。该属性从已创建的LTS日志主题中获取。 |
| TopicName | string | 必填
LTS日志主题名称。该属性从已创建的LTS日志主题中获取。 |

示例：LTS事件源对象

```
`yaml
LtsEvt:
  Type: LTS
  Properties:
    GroupId: fd537073-7ffb-11e8-b63a-286ed488ce71
    GroupName: test-log-group-1
    TopicId: 51acbef6-aa92-11e8-810a-286ed488ce71
    TopicName: lts-topic-aos
...`
```

- **CTS**
描述类型为CTS触发器的对象。

表 4-10 属性

| 属性名称 | 类型 | 描述 |
|------------|--------|---|
| Name | string | 必填
CTS触发器的名称。支持中文、英文、数字和下划线，长度不能超过64个字符。 |
| Operations | object | 必填
支持审计的服务操作。详情请参考 支持审计的服务操作 。 |

示例：CTS事件源对象

```
``yaml
CtsEvt:
  Type: CTS
  Properties:
    Name: ctstest
  Operations:
    AOS:
      stack:
        - createStack
        - deleteStack
      quota:
        - updateQuota
    CCE(1.0):
      app:
        - createApp
        - deleteApp
      image:
        - deleteImage
````
```

## 4.2 本地 HTTP API 示例

命令HSS apigw支持在本地通过curl调试函数。

要在通过curl调用函数，需要启动本地监控。此处以Python 2.7为例。

切换到test的Python 2.7目录，此处启用8888端口进行监听，命令

```
$ hss apigw -e evt.json -P 8888
```

或在交互模式下，直接输入如下命令执行：

```
apigw -e evt.json -P 8888
```

类似回显信息如下：

```
2018/08/27 15:41:46 INFO: Template: template.yml, successfully parsed
2018/08/27 15:41:46 INFO: Please curl http://0.0.0.0:8888/MyFuncStage2 to execution function
MyFuncStage2
2018/08/27 15:41:46 INFO: Please curl http://0.0.0.0:8888/MyFuncStage1 to execution function
MyFuncStage1
2018/08/27 15:41:46 INFO: Please provide a FunctionStage name shown above as the first agument.
```

打开新是shell窗口或postman，执行以下命令调用指定函数MyFuncStage1

```
curl -X GET http://127.0.0.1:8888/MyFuncStage1
```

类似回显信息如下：

```
Hello message: {"name": "TONY"}
```

也可以通过POST更改执行内容。命令如下：

```
curl -X POST http://127.0.0.1:8888/MyFuncStage1 -d '{"test":"message"}
```

类似回显信息如下：

```
Hello message: {"test": "message"}
```

## 4.3 本地日志

invoke命令允许将日志从调用函数的过程导入到文件中。如果正在运行针对HSS的自动测试并希望捕获日志以进行分析，这将非常有用。例如：

```
$ hss invoke --logFile ./output.log
```

## 4.4 使用环境变量

如果函数使用环境变量，环境变量的值可以从三种方式传递给Docker容器。按以下优先顺序从这三个源中选择每个环境变量：

1. 使用--envFile选项指定的环境变量文件。
2. Shell环境变量。
3. 在模板文件的功能属性中定义。

在定义环境变量时，请选择优先级较高的定义方式。

## HSAM 模板

考虑具有两个函数MyFunc1和MyFunc2的HSAM模板示例：

```
SAMversion: 'v1'
Resources:
 MyFunc1:
 Type: HEC::Serverless::Function
 Properties:
 Handler: gotest.handler
 Runtime: go1.8.3
 MemorySize: 1024
 Timeout: 30
 Environment:
 Variables:
 TABLE_NAME: my-table
 MyFunc2:
 Type: HEC::Serverless::Function
 Properties:
 Handler: pytest.handler
 Runtime: python2.7
 Environment:
 Variables:
 STAGE: prod
 TABLE_NAME: my-table
```

## 环境变量文件

使用invoke命令的--envFile参数来提供一个YAML文件，其中包含您的函数中定义的环境变量的值。

例如文件env.yaml内容如下。

```
NAME: "John"
LAST_NAME: "Smith"
```

示例：



```
$ hss invoke --envFile env.yaml
```

或在交互模式下，直接输入如下命令执行：

```
invoke --envFile env.yaml
```

## Shell 环境

如果你的shell环境中定义的环境变量映射到你的FunctionGraph函数中的环境变量，它们将被传递给Docker容器。Shell变量全局适用于所有函数。例如，如果两个函数使用一个名为TABLE\_NAME的环境变量，则通过shell环境提供的TABLE\_NAME值将可用于这两个函数。

以下命令将使TABLE\_NAME的值可用于MyFunction1和MyFunction2。

```
$ TABLE_NAME=mytable hss invoke
```

## Shell 和环境变量文件的组合

为了更好的控制，你可以使用shell环境变量和环境变量文件的组合。如果两个地方都定义了一个变量，则文件中的变量将覆盖shell中定义的环境变量。

## 4.5 调试

命令HSS invoke支持在本地调试函数。要在启用调试支持的情况下运行HSS，请在命令行上指定--debugPort或-d。

要在调试模式下在端口5000上本地调用函数，示例命令如下。

```
$ hss invoke -d 5000 <FunctionStage>
```

或在交互模式下，直接输入如下命令执行：

```
invoke -d 5000 <FunctionStage>
```

## 调试 Python 函数

Python要求在函数代码中启用远程调试，如果使用--debugPort或-d为使用其中一个Python运行时的FunctionGraph启用调试，则HSS将通过该端口从主机映射到FunctionGraph函数运行时容器。如果需要在函数代码中启用远程调试，请使用Python软件包，例如remote-pdb：<https://pypi.python.org/pypi/remote-pdb>

举例说明调试方法：

1. 在代码中import remote\_pdb，通过set\_trace设置断点，并监听5000端口。

```
import json
import remote_pdb
def handler(event, context):
 remote_pdb.RemotePdb(host='127.0.0.1', port=5000).set_trace()
 logger = context.getLogger()
 logger.info("start...")
 output = 'Hello message: ' + json.dumps(event)
 return output
```
2. 执行函数，添加-d 5000参数，将容器的5000端口映射到主机的5000端口。示例命令如下：

```
$ hss invoke -d 5000 MyFuncStage1
```

类似回显信息如下：

```
2018/08/23 16:59:07 INFO: Reading events from stdin since eventFile not provided:
{"key": "value" }
2018/08/23 16:59:13 INFO: Template: template.yml, successfully parsed
2018/08/23 16:59:13 INFO: Container created, cout:
[8ae754ed07ebbb70713710b5ba433e019d8aba44bc0817741716a67b1fb0d707]
2018/08/23 16:59:14 INFO: Executing the function successfully in the Container, code: 200
root@SZX1000371099:~/data/1.0.4/hss/test/sample-code/python2.7# vi main.py
root@SZX1000371099:~/data/1.0.4/hss/test/sample-code/python2.7# echo '{"key": "value" }' | hss
invoke -d 5000 MyFuncStage1
2018/08/23 17:01:14 INFO: Reading events from stdin since eventFile not provided:
2018/08/23 17:01:14 INFO: Template: template.yml, successfully parsed
2018/08/23 17:01:14 INFO: Container created, cout:
[1ac4aee102d04b38fb447adc7cb53739be47426faf6dd837abfb3b836b9cdd3f]
2018/08/23 17:01:15 CRITICAL:root:RemotePdb session open at 0.0.0.0:5000, waiting for connection ...
2018/08/23 17:01:15 RemotePdb session open at 0.0.0.0:5000, waiting for connection ...
2018/08/23 17:02:23 CRITICAL:root:RemotePdb accepted connection from ('172.17.0.1', 40888).
2018/08/23 17:02:23 RemotePdb accepted connection from ('172.17.0.1', 40888).
2018/08/23 17:03:50 INFO: Executing the function successfully in the Container, code: 200
```

3. 在新的shell窗口执行telnet 127.0.0.1 5000打开pdb调试命令行，具体调试命令可查看上述链接。

类似回显信息如下：

```
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^'.
> /var/fssrt/function/main.py(6)handler()
(Pdb)
```

## 调试 Go 函数

调试功能暂不可用。

## 调试 Java 函数

调试功能暂不可用。

## 调试 Node.js 函数

调试功能暂不可用。

## 调试 C#函数

调试功能暂不可用。

## 调试 PHP 函数

调试功能暂不可用。