Slider Docker源码分析

**Input Data**

metainfo.json,appConfig.json和resources.json

其中metainfo.json和appConfig.json决定了docker container的所有信息，resoucres.json的作用是仅决定了YARN Containers的启动数目和设置内存及cpu。具体的定义方式在其他文档中介绍

**流程描述**

Agent main.py程序启动之前的流程是相同的，Agent和SliderAppMaster联通后进行心跳交互。

1. 判断是否是docker

AgentProviderService

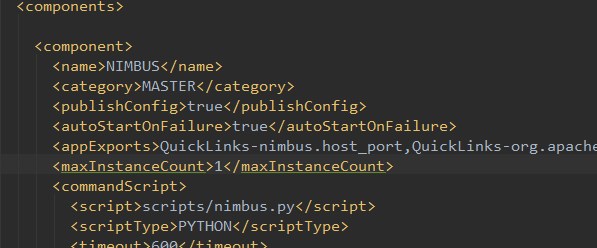
#isDockerContainer //判断component为docker类型

#isYarnDockerContainer //判断 component为yarn\_docker类型

从metainfo.xml中获取组件的type，如下所示：



对于非Docker的组件type为STANDARD：



type默认为TYPE\_STANDARD [STANDARD]，可设置类型：STANDARD|DOCKER|

PYTHON及YARN\_DOCKER

1. AgentProviderServier向Agent发送INSTALL命令

Slider AM根据Agent发过来的心跳信息，生成Install CommandAgentProviderService

*#addInstallDockerCommand*

cmd初始化信息如下：

#taskId，和command ID <= taskID+"-1"

#clusterName

#roleCommand <= INSTALL

#serviceName

#componentName

#roleName

#hostLevelParams

package\_list <= osSpecifics，[tarball,xxx.tar.gz]

containerId

#configurations

在appConfig.json中的global配置

#componentCommand

name <= INSTALL

exec <= DEFAULT

cmdParams: service\_package\_folder,cmd,schema\_version,command\_timeout,

script\_type,record\_config

#hostName

#addContainerDetails //docker执行相关信息

image <= metaContainer.getImage

network

networkScripts

name

options

additionalParams

commandPath

statusCommand

startCommand

mounts

ports

inputFile

configFiles

# dockerConfig 放入configurations中

yarnDocker，//do Nothing

docker，command\_path <= docker bin //来自于appConfig.json中的配置

docker.image\_name <= image

注意：cmd以HeartbeatResponse的形式放到给Slider Agent，

response.setRestartEanbled <= autoStartOnFailure是否为true

1. Agent端接收Response（先Register然后再Heatbeat）

Controller.py#heatbeatWithServer

response <= sendRequest(heartbeatUrl,data)

addToQueue <= executionCommands { response['executionCommands']}

updateStateBasedOnCommand ||在不同的状态会有不同的statusCommand

*将executionCommands传入Queue中等待调度*

ActionQueue#process\_command

#execute\_command [ExecutionCommands]

//根据命令的字段，决定使用哪个脚本执行命令

--------------- yarnDockerMode ---------------------

YarnDockerManager.py#exectue\_command

#获取config

#执行程序

INSTALL => install\_container

#configFiles 生成配置文件，=> workroot, InlineTemplate

START => start\_container

#outFile,errFile

#execute\_command\_on\_linux

startCommand <= extract\_config\_from\_command

与DockerManager的不同在于执行命令的组合方式不同，而且其中有获取container的执行Ip和Hostname命令

------------------- dockerMode -------------------------

DockerManager.py#execute\_command

1）获取config

2）获取命令名

INSTALL => pull\_image docker pull {imageName}

START => start\_container

#port

#mount

#inputFile

#memory

#image

#param

------------------------ regular ------------------

CustomServiceOrchestrator.runCommand

#获取脚本，并执行

1. Agent返回心跳（命令执行结果）

#commandResult <= 命令执行结束

#commandStatuses <= command,roleResult,reportResult ActionQueue

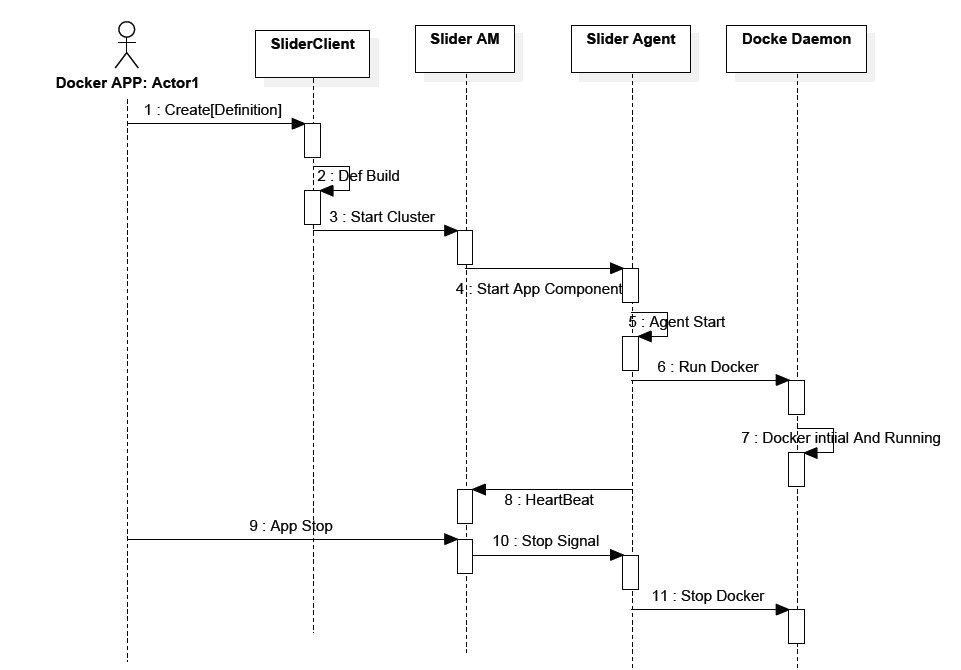
#Controller data<= Heatbeat.build() ActionQueue#commandStatus

#Controller.sendRequest[url,data]

总体框架：



流程图如下：



**附录:addInstallCommand的command json文件**

{

"componentConfigurations": {

"slider-appmaster": {

"env.MALLOC\_ARENA\_MAX": "4",

"zookeeper.path": "/services/slider/users/dev/demo-tomcat",

"site.global.xmx\_val": "1g",

"zookeeper.hosts": "sgserver001,sgserver002,sgserver003",

"jvm.heapsize": "128M",

"site.fs.default.name": "hdfs://test122",

"site.server-xml.http.port": "${TOMCAT.ALLOCATED\_PORT}",

"site.global.memory\_val": "200M",

"site.global.xms\_val": "128m",

"application.def": "/slider/appzip/slider-tomcat-app-package-0.80.0-bc1.3.1.zip",

"site.server-xml.connection.timeout": "20000",

"zookeeper.quorum": "sgserver001:2181,sgserver002:2181,sgserver003:2181",

"site.global.hdfs\_app\_dir": "/slider/.slider/cluster/demo-tomcat/localstore",

"site.fs.defaultFS": "hdfs://test122",

"java\_home": "/usr/jdk64/jdk1.7.0\_67",

"site.global.security\_enabled": "false"

}

},

"clusterName": "demo-tomcat",

"componentName": "TOMCAT",

"hostname": "sgserver005",

"hostLevelParams": {

"java\_home": "/usr/jdk64/jdk1.7.0\_67",

"container\_id": "container\_1469588862638\_0068\_02\_000002",

"package\_list": "[{\"type\":\"tarball\",\"name\":\"files/tomcat-7.0.62.tar.gz\"}]"

},

"commandType": "EXECUTION\_COMMAND",

"roleCommand": "INSTALL",

"serviceName": "demo-tomcat",

"role": "TOMCAT",

"commandParams": {"record\_config": "false",

"foreground\_start": "false",

"script": "scripts/tomcat.py",

"schema\_version": "2.0",

"service\_package\_folder": "${AGENT\_WORK\_ROOT}/work/app/definition/package",

"command\_timeout": "600",

"script\_type": "PYTHON"

},

"taskId": 1,

"public\_hostname": "sgserver005",

"appmaster\_hostname": "sgserver005",

"commandId": "1-1",

"containers": [],

"configurations": {

"global": {

"security\_enabled": "false",

"app\_container\_id": "container\_1469588862638\_0068\_02\_000002",

"xms\_val": "128m",

"app\_root": "/hadoop/yarn/local/usercache/dev/appcache/application\_1469588862638\_0068/container\_1469588862638\_0068\_02\_000002/app/install",

"app\_log\_dir": "/hadoop/yarn/log/application\_1469588862638\_0068/container\_1469588862638\_0068\_02\_000002",

"xmx\_val": "1g",

"app\_pid\_dir": "/hadoop/yarn/local/usercache/dev/appcache/application\_1469588862638\_0068/container\_1469588862638\_0068\_02\_000002/app/run",

"memory\_val": "200M",

"app\_container\_tag": "1",

"pid\_file": "/hadoop/yarn/local/usercache/dev/appcache/application\_1469588862638\_0068/container\_1469588862638\_0068\_02\_000002/app/run/component.pid",

"app\_install\_dir": "/hadoop/yarn/local/usercache/dev/appcache/application\_1469588862638\_0068/container\_1469588862638\_0068\_02\_000002/app/install",

"hdfs\_app\_dir": "/slider/.slider/cluster/demo-tomcat/localstore",

"app\_input\_conf\_dir": "/hadoop/yarn/local/usercache/dev/appcache/application\_1469588862638\_0068/container\_1469588862638\_0068\_02\_000002/propagatedconf"

},

"server-xml": {

"connection.timeout": "20000",

"http.port": "48694"

}

**附录2：addStartCommand**

{

"componentConfigurations": {

"slider-appmaster": {

"env.MALLOC\_ARENA\_MAX": "4",

"zookeeper.path": "/services/slider/users/dev/demo-tomcat",

"site.global.xmx\_val": "1g",

"zookeeper.hosts": "sgserver001,sgserver002,sgserver003",

"jvm.heapsize": "128M",

"site.fs.default.name": "hdfs://test122",

"site.server-xml.http.port": "${TOMCAT.ALLOCATED\_PORT}",

"site.global.memory\_val": "200M",

"site.global.xms\_val": "128m",

"application.def": "/slider/appzip/slider-tomcat-app-package-0.80.0-bc1.3.1.zip",

"site.server-xml.connection.timeout": "20000",

"zookeeper.quorum": "sgserver001:2181,sgserver002:2181,sgserver003:2181",

"site.global.hdfs\_app\_dir": "/slider/.slider/cluster/demo-tomcat/localstore",

"site.fs.defaultFS": "hdfs://test122",

"java\_home": "/usr/jdk64/jdk1.7.0\_67",

"site.global.security\_enabled": "false"

}

},

"roleCommand": "START",

"clusterName": "demo-tomcat",

"componentName": "TOMCAT",

"hostname": "sgserver005",

"hostLevelParams": {

"java\_home": "/usr/jdk64/jdk1.7.0\_67",

"container\_id": "container\_1469588862638\_0068\_02\_000002"

},

"commandType": "EXECUTION\_COMMAND",

"roleParams": {

"auto\_restart": "false"

},

"serviceName": "demo-tomcat",

"role": "TOMCAT",

"commandParams": {

"record\_config": "true",

"foreground\_start": "false",

"script": "scripts/tomcat.py",

"schema\_version": "2.0",

"service\_package\_folder": "${AGENT\_WORK\_ROOT}/work/app/definition/package",

"command\_timeout": "600",

"script\_type": "PYTHON"

},

"taskId": 2,

"public\_hostname": "sgserver005",

"appmaster\_hostname": "sgserver005",

"commandId": "2-1",

"containers": [],

"configurations": {

"global": {

"security\_enabled": "false",

"app\_container\_id": "container\_1469588862638\_0068\_02\_000002",

"xms\_val": "128m",

"app\_root": "/hadoop/yarn/local/usercache/dev/appcache/application\_1469588862638\_0068/container\_1469588862638\_0068\_02\_000002/app/install",

"app\_log\_dir": "/hadoop/yarn/log/application\_1469588862638\_0068/container\_1469588862638\_0068\_02\_000002",

"xmx\_val": "1g",

"app\_pid\_dir": "/hadoop/yarn/local/usercache/dev/appcache/application\_1469588862638\_0068/container\_1469588862638\_0068\_02\_000002/app/run",

"memory\_val": "200M",

"app\_container\_tag": "1",

"pid\_file": "/hadoop/yarn/local/usercache/dev/appcache/application\_1469588862638\_0068/container\_1469588862638\_0068\_02\_000002/app/run/component.pid",

"app\_install\_dir": "/hadoop/yarn/local/usercache/dev/appcache/application\_1469588862638\_0068/container\_1469588862638\_0068\_02\_000002/app/install",

"hdfs\_app\_dir": "/slider/.slider/cluster/demo-tomcat/localstore",

"app\_input\_conf\_dir": "/hadoop/yarn/local/usercache/dev/appcache/application\_1469588862638\_0068/container\_1469588862638\_0068\_02\_000002/propagatedconf"

},

"server-xml": {

"connection.timeout": "20000",

"http.port": "48694"

}

}

}