CustomServiceOrchestrator.py <= allocated\_ports

#runCommand <= command

dump\_command\_to\_json

#install 传递的参数是app\_config.json中的参数

componentConfigurations

#start //该命令是启动命令

#finalize\_command //完成参数替换

${AGENT\_WORK\_ROOT} => AgentConfig.getWorkRootPath

${AGENT\_LOG\_ROOT} => AgentConfig.getLogPath

${{{0}.ALLOCATED\_PORT}}

//其中{0} -> component\_name

allowed\_ports <= get\_allowed\_ports

slider.allowed.ports <- global\_config.get('slider.allowed.ports')

#allocate\_ports

//${component\_name.ALLOCATED\_PORT}||{DEFAULT||{PER\_CONTAINER}

主要是使用端口替换这几个字符串

#allocate\_port

在分配过程会查看该port是否可用

sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

sock.settimeout(0.2)

sock.connect(('127.0.0.1', port))

sock.close()

如果slider中配置了slider.allowed.ports，则使用这些端口，否则：

port\_list = [0] \* MAX\_ATTEMPTS

itor = iter(port\_list)

port = sock.getsockname()[1]

#json\_path

<= self.dump\_command\_to\_json(command, allocated\_ports, store\_command)

script\_params

<= [command\_name,json\_path,current\_base\_dir]

#python\_executor.run\_file

py\_file

script\_params

environment\_vars