Install 目录的获取

流程图如下：



执行流程：

SliderClient #actionUploadFile (string clusterName,ActionUploadArgs uploadArgs)

——> 添加类ActionUploadArgs，参数:containerID,APP路径

——> 方法处理流程：

1) 获取参数中的ContainerID，或者根据clusterName获取liveContainers

2) 根据clusterName创建RPC连接

3）调用SliderIPCService#upload(request)

SliderIPCService#uploadFile -> SliderClusterProtocol#upload

——> ActionUploadFile extends AyncAction

#execute(SliderAppMaster,QueueAccess,AppState)

——> 方法处理流程：

1）upload方法，#schedule(AsyncAction action)

2）QueueService implements QueueAccess #schedule ->

将action放入scheduledActions，

3）QueueService：按照FIFO原则，从scheduledActions取出action后，

放入actionQueue

——> 修改proto文件，增加upload方法

QueueExecutor#run

——> 从QueueService中获取action，QueueService.actionQueue.take并执行

ActionUpload.execute(SliderAppMaster,QueueService,AppState)

SliderAppMaster#onUploadFile(ActionUpload)

——>根据containers及components，获取所有需要操作的containers（将component对应的containerID放入containers列表）

——>AgentProviderServier#setExecuteCMD

——> AgentProviderServier#addExecuteContainers

AgentProvidersService#handleHeartbeat

——> 根据传入的参数设置Response Commond，当前流程如下所示：

当前command的类型包括：

|  |  |
| --- | --- |
| 参数名 | 含义 |
| NOP | 空操作 |
| INSTALL | 安装Component |
| INSTALL\_ADDON | 安装其他的Package（如果有） |
| START | 启动Component |
| STOP | 停止Component |
| UPGRADE | 升级 |
| TERMINATE | 停止agent |

AgentCommand的类别有以下几种：

|  |  |
| --- | --- |
| 参数名 | 含义 |
| EXECUTION\_COMMAND | INSTALL/STOP/UPGRADE |
| STATUS\_COMMAND | GET\_CONFIG/STATUS |
| REGISTRATION\_COMMAND | 没有类别的命令 |

CommandResult ，命令执行结果状态：

|  |  |
| --- | --- |
| 参数名 | 含义 |
| IN\_PROGRESS | 执行过程中 |
| COMPLETED | 执行完成 |
| FAILED | 执行失败 |

Component instance的状态包括：

|  |  |
| --- | --- |
| 参数名 | 含义 |
| INIT | 初始状态，没有安装 |
| INSTALLING | 正在安装 |
| INSTALLED | 安装其他的Package（如果有） |
| STARTING | 启动状态 |
| STARTED | 已经启动状态 |
| INSTALL\_FAILED | 安装失败（安装后，启动失败） |
| UPGRADING | 更新状态，执行更新前需要进行一系列操作 |
| UPGRADED | 更新完成 |
| STOPPING | 启动停止命令 |
| STOPPED | 停止Agent |
| TERMINATING | 给Agent发送停止信号，执行停止操作 |

在handleHeartbeat中，根据Agent的状态及触发操作，Command发送流程的算法如下所示：



Slider Agent: Controller.py # registerAndHeartbeat

——>Controller#heartbeatWithServer

——>处理流程：

1）sendRequest，获取response，REST（heartbeat URL），调用AgentProviderService#

handleHeartBeat

2）解析response(json数据)，根据response.keys执行命令，例如executionCommands in reponse.keys

3）执行命令：更新agent状态：updateStateBasedOnCommand，命令包括 START,STOP,INSTALL等；componetExpectedState,componentActualState，同时配置 command的一些参数，例如command['configurations']['global']['app\_root'] ，[app\_version]

将命令传送到agent#actionQueue，保存了command的Queue，FIFO的方式进行执 行

CustomServiceOrchestrator#runCommand

核心的类是HeartBeatResponse，下图描述了相关类之间的关系：



AgentProviders#addUploadFileCommnd

——>handleHeartbeat中根据参数进行设置executeCommands的参数