1. 接口
2. 功能

规则执行器的主要功能是执行调度算法产生的规则。其主要功能如下：

1. 新建一个pod，将其绑定到特定node上
2. 删除一个pod

2.接口设计

|  |  |
| --- | --- |
| 接口编号 | I\_1\_1 |
| 接口功能 | 创建一个pod，将其绑定到特定node上 |
| 接口参数 | 1. 只传一个string类型的json列表，json中包含下面几条属性 2. Pod属主类型 3. Pod属主名称 4. Node名称或IP 5. 创建pod的个数 6. 时间戳 7. 命名空间 |
| 接口返回值 | 成功与否 |
| 接口url | POST /pods?JsonArray=? |

|  |  |
| --- | --- |
| 接口编号 | I\_1\_2 |
| 接口功能 | 删除一个pod |
| 接口参数 | 1. 只传一个string类型的json列表，json中包含下面几条属性 2. Pod属主类型 3. Pod属主名称 4. 删除pod个数 5. 命名空间 6. 时间戳 |
| 接口返回值 | 成功与否 |
| 接口url | DELETE /pods?JsonArray=? |

1. 功能点
2. scheduler的工作流程
   1. 首先从调度算法获取到数据（scheduler的接口）
   2. 然后规则执行器在kubernetes集群中创建规则的承载体——rules（自定义资源，权且命名为rules）
   3. 再然后规则执行器执行创建或删除pod的指令，删除kubernetes集群中的pod
   4. 在kube-scheduler执行过程中，需要干涉pod的修改
   5. Pod绑定后，需要修改规则的承载体rules
3. Scheduler的内部功能点
   1. 接口编写
   2. 规则承载体rule的创建（不进行删除和修改）
   3. 访问kubernetes的API，对pod进行操作
   4. 优选策略的编写
   5. 自定义资源对应的controller的编写，修改和删除rules
4. 开发任务
   1. 接口编写
   2. Rule的创建以及对pod的操作
   3. 优选策略编写
   4. Controller编写