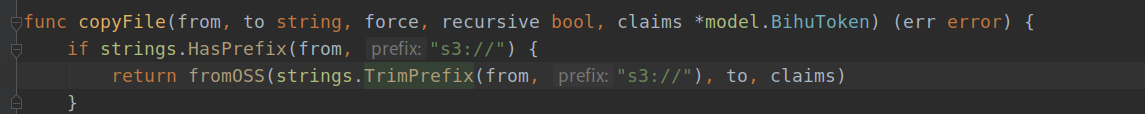
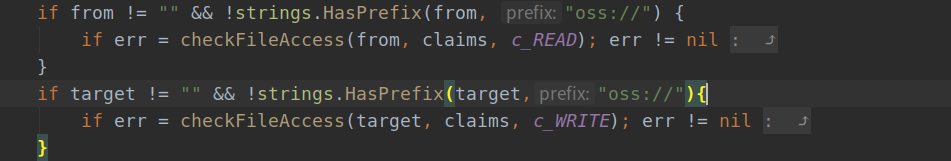
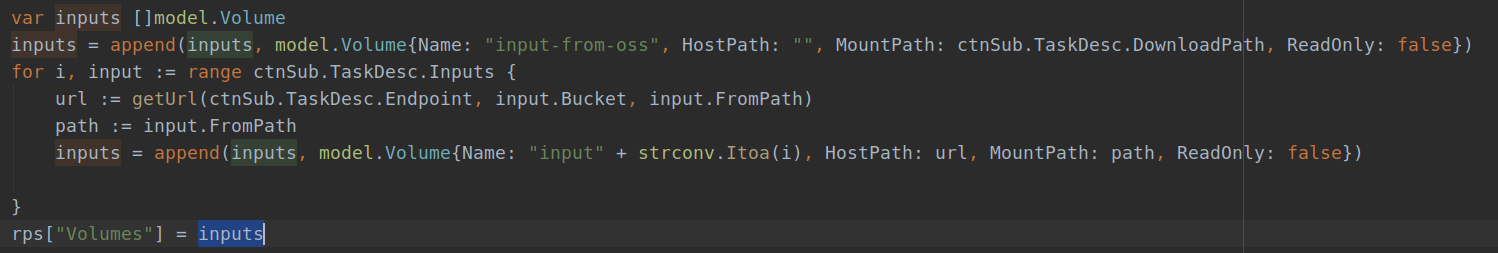
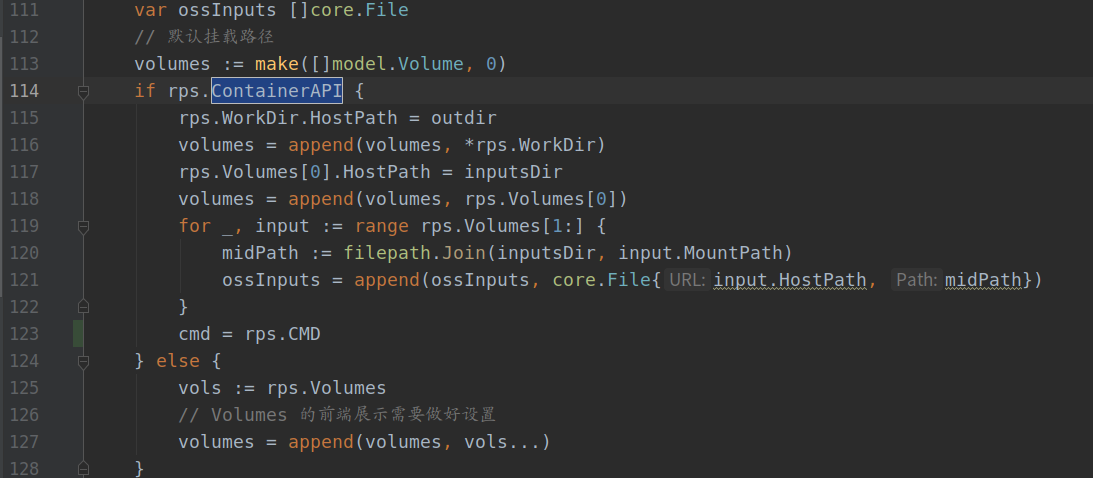
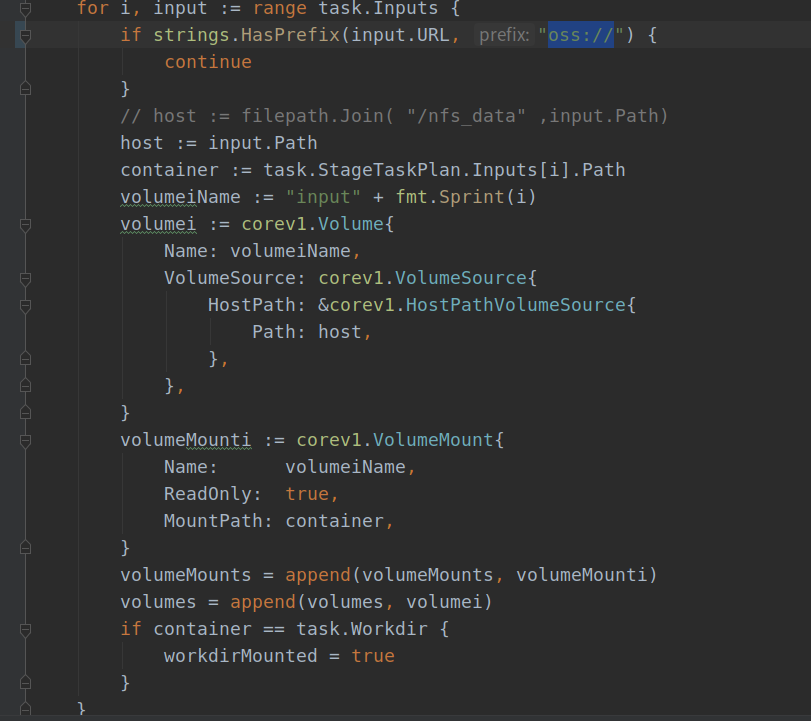
## Submit

submit功能除了封装甲方的请求，转换为星光的格式以进行作业提交，目前主要是需要添加对对象存储的支持。用户的请求如下

{
  
 "request\_id":"123",
  
 "tags":"codeWell",
  
 "task\_desc":{
  
 "image":"192.168.182.136:5000/fm",
  
 "task\_id":"",
  
 "cmd":["/root/fm"],
  
 "cpu\_limit":"1",
  
 "mem\_limit":"50Mi",
  
 "repo\_path":"",
  
 "env":{},
  
 "inputs":[
  
 {
  
 "from\_path":"succ.file.txt",
  
 "bucket":"nzc"
  
 }
  
 ],
  
 "outputs":[],
  
 "download\_path":"/abc",
  
 "upload\_path":"/cde",
  
 "inputEndpoint":"http://oss-cn-shenzhen.aliyuncs.com"
  
 }
  
}

目前只做了aliyun文件下载功能，上传功能只添加在storage服务，job回收时上传文件的逻辑还没写

### 代码改动

* 在storage服务下添加了下载和上传功能，但当前的对应提供商的账户和认证信息都写死，日后需要改进：
* 
* 在函数fileOperationNewXX下，当from或target地址拥有前缀oss://时，不需要检查相应地址是否存在  
  
* 在RuntimeParamsStandard中添加了属性ContainerAPI,将其属性Volumes用于存储用户上传文件信息。  
  （submit.go）  
  
* Volumes[0]为挂载信息，即共享存储路径和容器内下载路径的映射，Volumes[1:]为下载文件信息
* (submitCwlJob.go)  
  
* 当前主要是通过bool变量ContainerAPI以及判断路径是否包含前缀oss://来将api逻辑嵌入到星光中（感觉这么做不太好）
* 例如，在runner.go中，task.Input中的内容的会进行挂载，但如果其中某个元素带有oss://前缀，则无需重复挂载  
  

## 其他功能

1. 获取用户所有容器信息，但没有做获取单个容器信息，需要先完成容器监控
2. 获取日志：用户需求时获取一段时间内的日志（给出了起止时间）目前没有方便测试的用例，同时k8s获取日志似乎只有获取某个时间点以来的所有日志，大概需要查询后再筛选
3. 停止任务，删除容器：对用户的请求进行了封装，直接调用现有api