# Kubesphere使用手册

## 前言

官方定义：一个以 Kubernetes 为内核的云原生分布式操作系统，支持云原生应用在多云与多集群的统一分发和运维管理。

对于我们企业级用户来说，在大规模容器化编排的今天，Kubernetes已经成为大规模容器编排的事实性标准。但在企业级应用上，我们需要更加直观，简洁。面向“用户”和“管理”的产品，在对比了，Dashboard，Kuboard，Rancher等开源系统后，最终选择了kubesphere做为环境运行服务平台。

以下，从一个例子出发，介绍kubesphere具体使用中需要说明的点

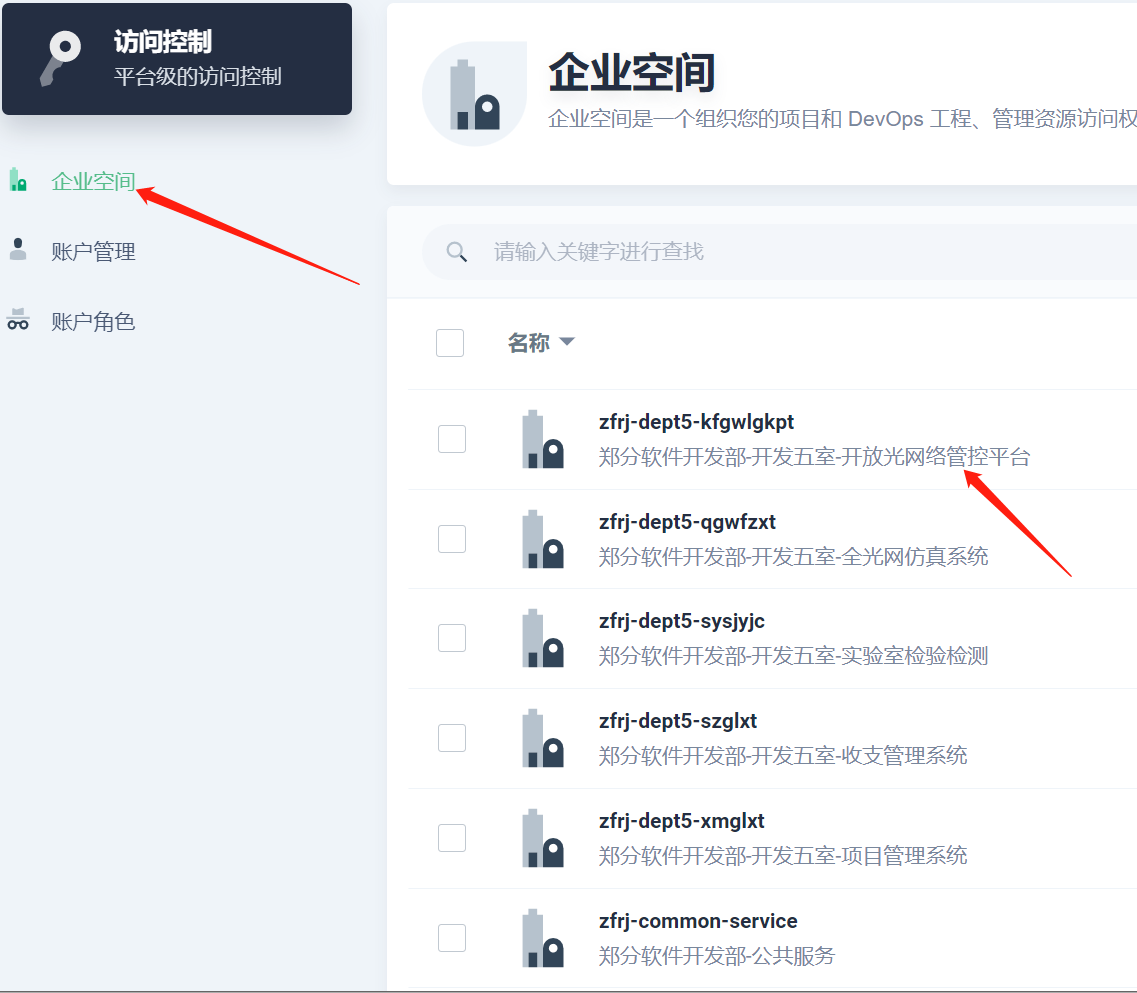
## 关于企业空间

Kubesphere中的企业空间，在实际使用中就相当于我们的一个个项目，不同的企业空间对应着不同的项目名字。

如下图，zfrj-dept5-kfgwlgkpt ，这个名字的企业空间代表：

“郑分软件开发部-开发五室-开放光网络管控平台” 这个项目

在用户的分配上，企业空间也是基础条件之一

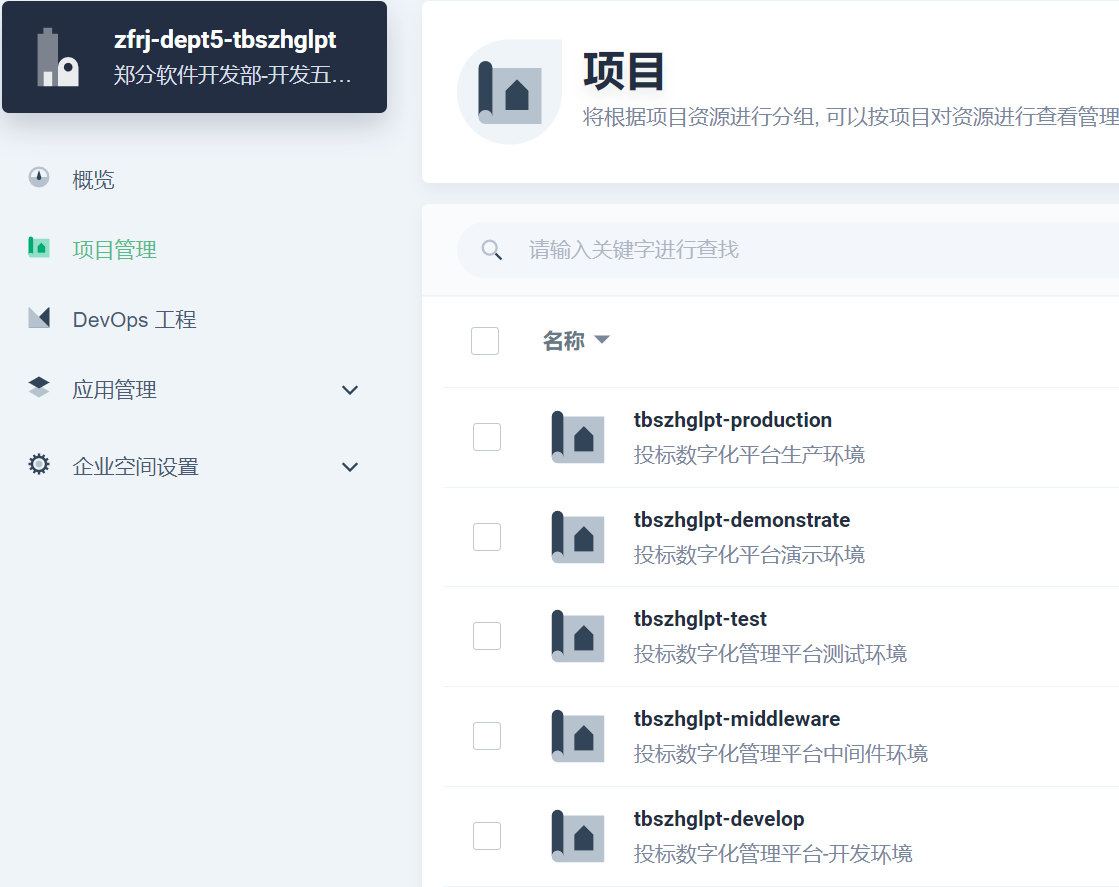


进入企业空间之后，用到最多的两块是 “项目管理” 和 “DevOps工程”

* 项目管理

如下图，点击项目管理，可以看到我们这个项目所对应的不同环境，基本会有，开发，测试，演示和生产环境

项目管理这里，具体指的就是项目的具体环境。开发所申请的具体权限，就是跟这个不同的环境有关的



* DevOps工程

代表的是CI/CD部分，用于项目的持续集成，持续部署。本kubesphere承担的是CD部分。

这里我们为了解耦，不过度在kubesphere上跑大量的CI工作，占用过多的集群资源，具体的CI工作由gitlab的CI来具体完成

DevOps里也分不同的流水线，用来区分分别对应的环境

如下： tbszhglpt-pipeline-develop-backend 代表投标数字化管理平台开发环境的后端服务流水线，点击运行，会部署到响应的开发环境中去，也就是前面的 tbszhglpt-develop



## 关于权限

目前，对于开发人员开放了两种权限，

* 可发起项目更新的权限
* 可查看权限

可更新权限包含可查看权限，具体使用中，可由每个项目的具体负责人，灵活安排人员。运维具体配置项目负责所需要的

不同的角色，在使用可担任起不同的任务，

## 常见FAQ

1. 我的服务怎么访问

## IP + Port + [ /URI ]

在kubesphere上部署的服务，外网访问是可选的，外网端口一般固定为**10.10.48.31**，如果没有说明则不添加外网访问，服务间以service名称进行访问

提示：一些用到的中间件一般需要外网访问，比如数据库，minio存储

1. 在哪里查看服务具体端口

进入你的项目环境，比如下图，找到应用负载，点击，然后点击服务



既可以在右侧看到具体的对外端口有哪些，在“外网访问” 列，可以看到对应的5位数的端口，即比如上图30095既是

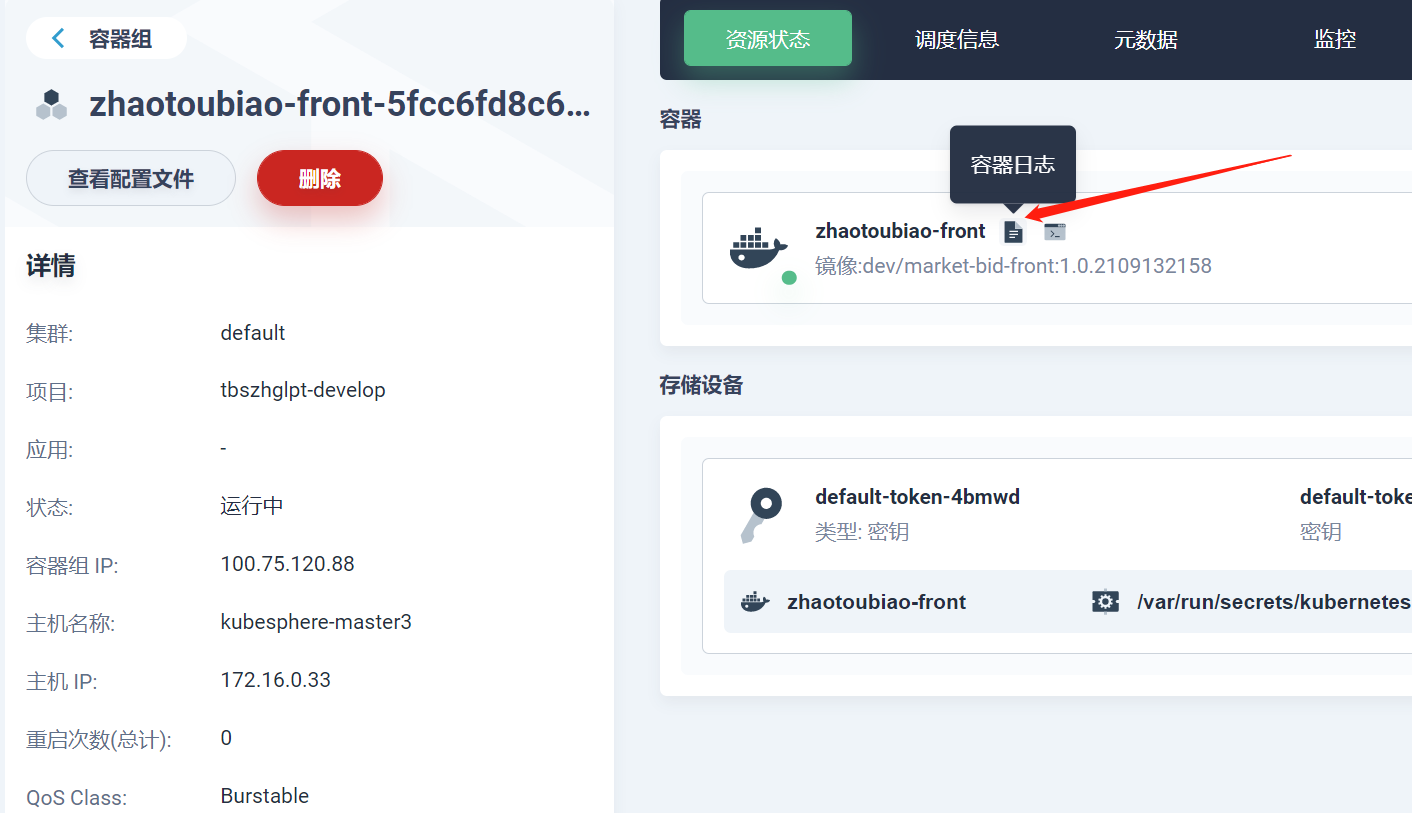
1. 如何查看日志

如下图，服务下方的工作负载，点击进入，即可看到对应服务



一般服务会在 “部署” 里看到，或者 “有状态副本集” 里看到，而且服务的名称即可自己理解，一般直接是工程名称，或者服务模块较多的话就是模块名称或者gitlab里的工程名称

如上front点击，即可在右侧看到该日志标志

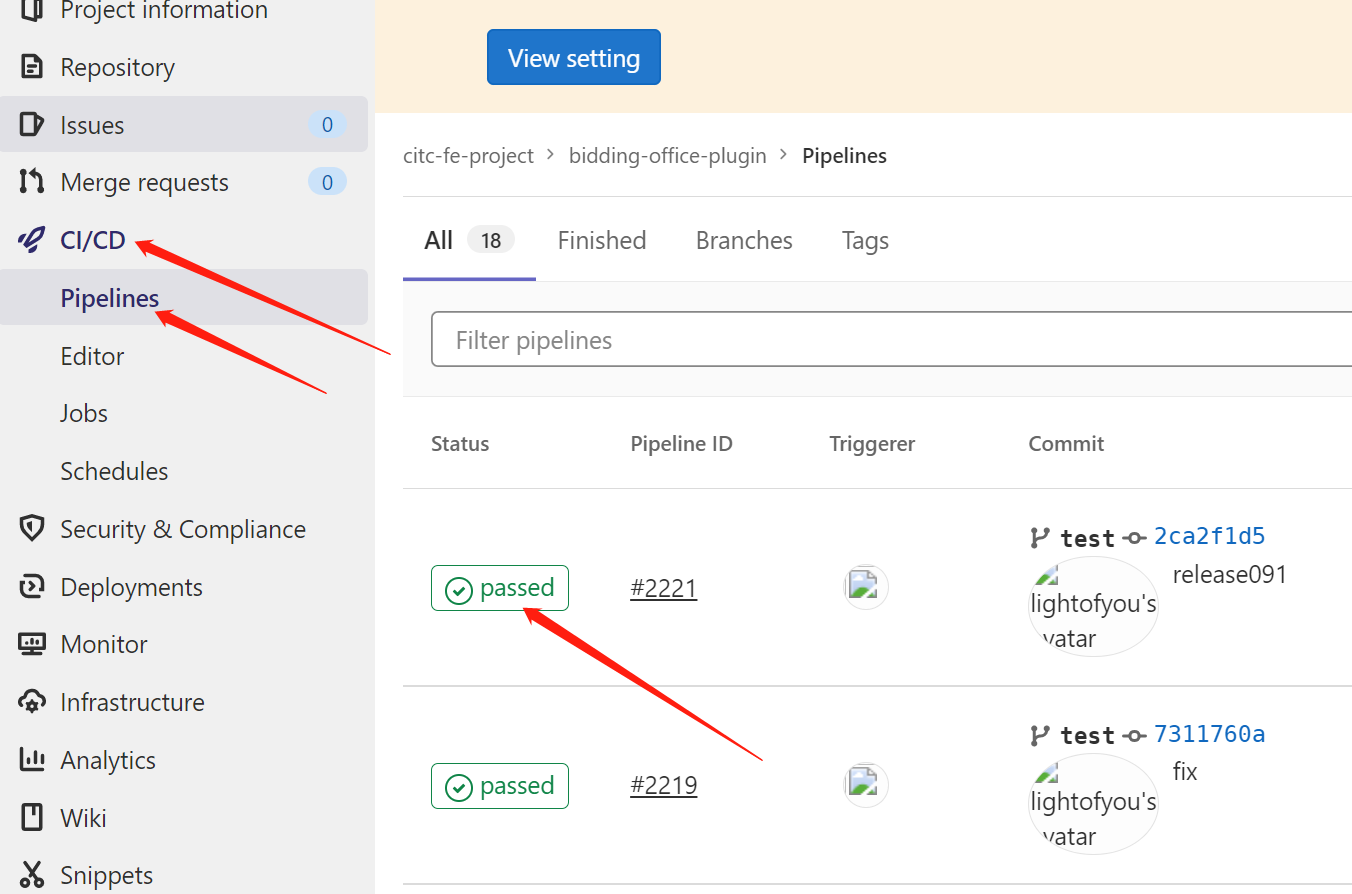


点击容器日志，即可查看对应的服务日志

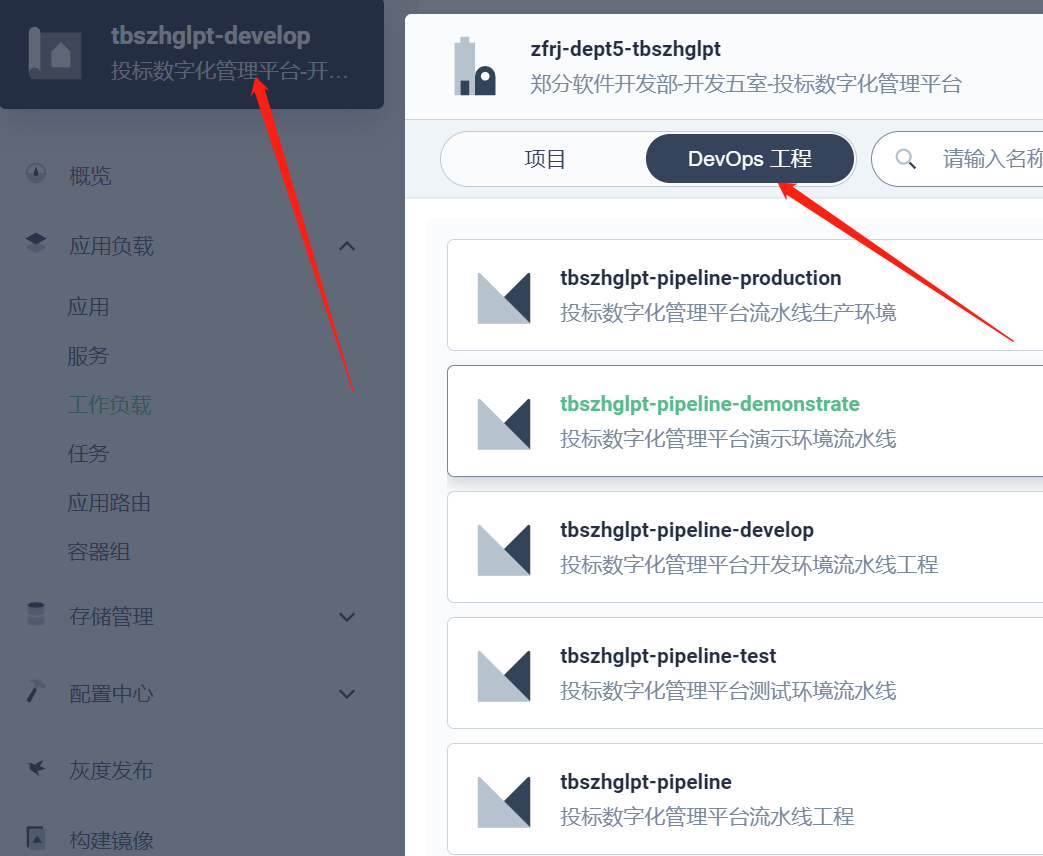
1. 如何更新升级服务

*在使用自动升级服务（CD）之前，一般要先在自己的git仓库中配置好gitlab-ci的工作，做自动触发，完成构建和push镜像工作，详情请参见****《软件开发部DevOpt&TestOps体系规范》***

在push或者merge代码到特定的分支后，可在gitlab的CI/CD页面查看具体任务情况，如下，看到passed状态后，即可进行自动部署的操作

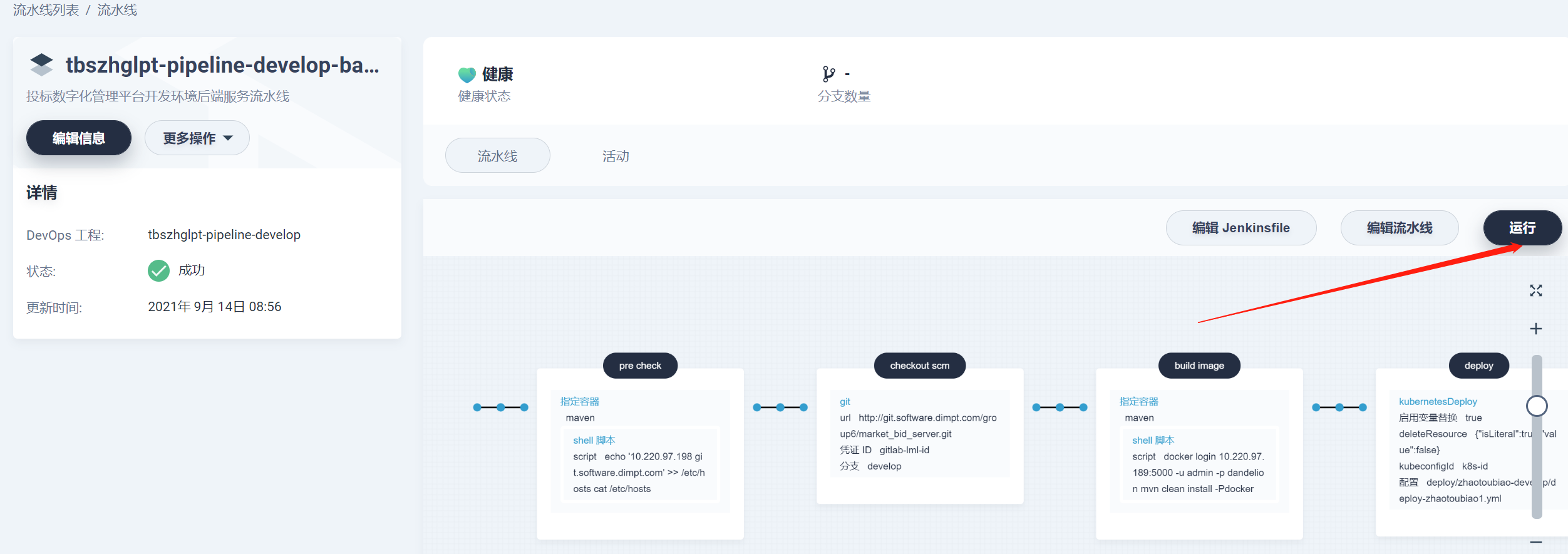


在左上角黑色方块区域点击可切换项目环境和DevOps工程，进入对应环境的DevOps，找到对应的后端或前端流水线，点击进去点运行即可





点击 “ 运行”



然后关注该任务的具体状态即可。

1. 查看工程服务配置

在具体项目的具体环境下点击 “配置管理” 🡪 “配置”



具体的名称与之前运维与开发之间的约定有关，比如上图中application-test带表开发环境

1. 服务之间访问怎么配置

服务之间的访问是以 service 名称来进行访问的，kubesphere中跨项目，跨环境都可以访问，具体访问的service名称可向运维咨询。

另外，**不要在服务中配置诸如 10.10.48.31:3xxxx 这样的访问方式，**内部调用时这个是不支持通信的