(1).wayne中创建命名空间

(2).wayne创建apollo项目

(3).wayne中容器化apollo

(1).wayne中创建命名空间

点击“创建命名空间”:



选中“自动创建”会在K8S集群中创建对应的命名空间。名称指的是在wayne中的逻辑名称，方便wayne管理，两者尽量保持一致。

可用机房：0.1表示cpu核数，1表示内存是1G；分别表示这个namespace中cpu和内存可以达到的上限。

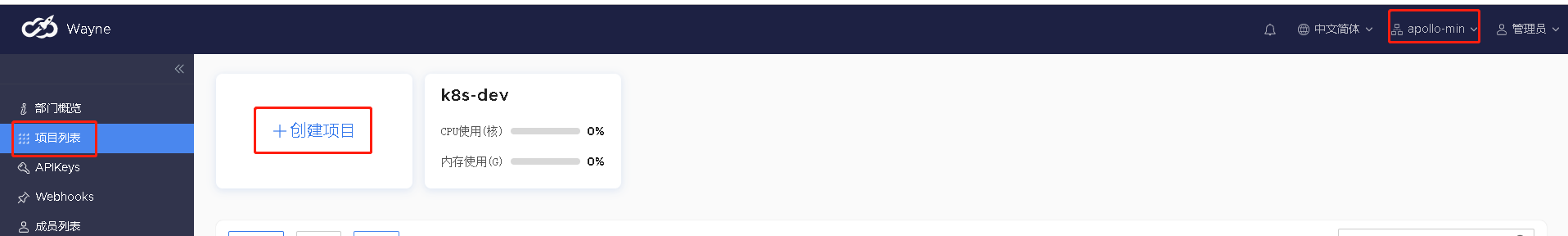


(2).wayne创建apollo项目

wayne中项目的概念：

一个namespace(wayne与K8S共有)中可以部署多个项目，比如说用户中心这个部门(对应user-namespace)下有多个项目，passport, account, user等项目；而每个项目又对应多个服务，比如passport项目对应passport-rpc, passport-web等。可以如此类比理解wayne中的管理结构。

返回前台创建项目：



后边我们要容器化的apollo的各项服务都会放在下图中的apollo-min项目中：



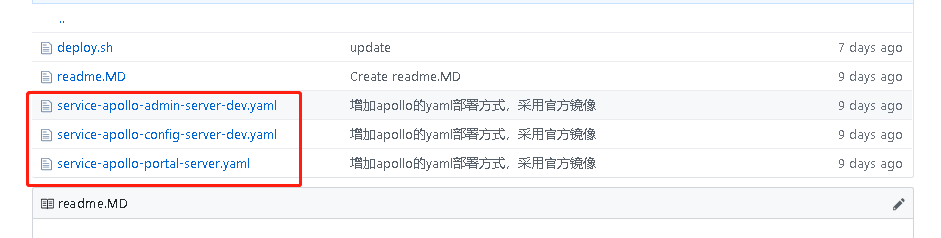
(3).wayne中容器化apollo

在前台的项目列表页中进入项目apollo-min：



笔者提供了dev环境下的最小apollo集群容器化的配置文件，位于：

<https://github.com/hepyu/k8s-app-config/tree/master/yaml/min-cluster-allinone/apollo-min>

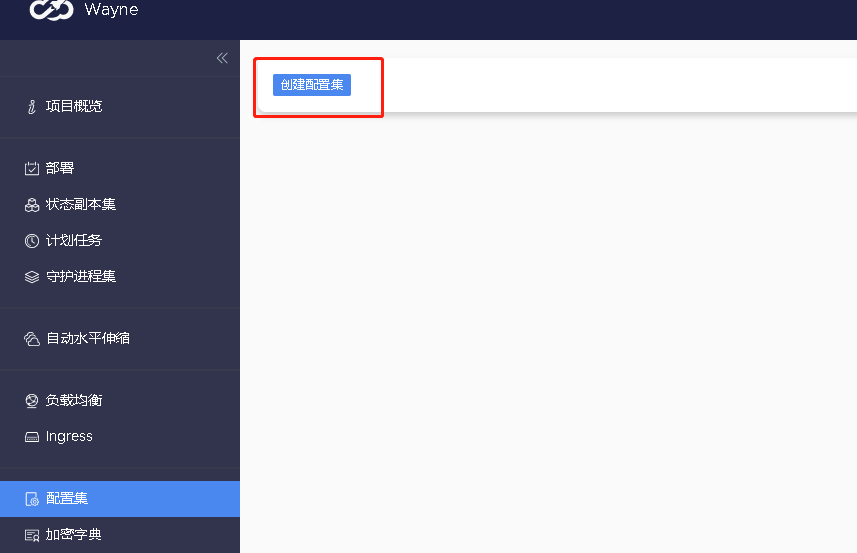


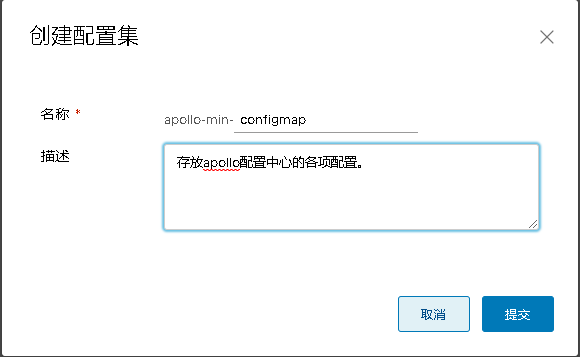
以wayne对apollo-config-server容器化举例，admin-server与portal-server类似：

apollo-config-server有4个组件：1个Configmap, 2个Service(其中1个是nodePort暴露apollo配置服务到容器外部)，1个StatefulSet。

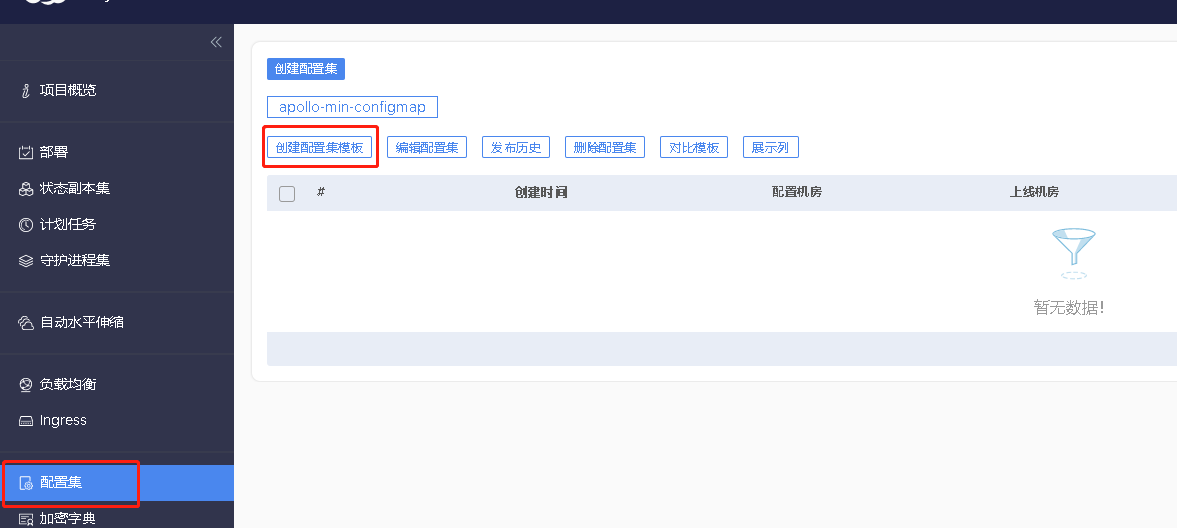
1.创建configmap

configmap对应wayne中的配置集中的每个配置项，创建配置集：



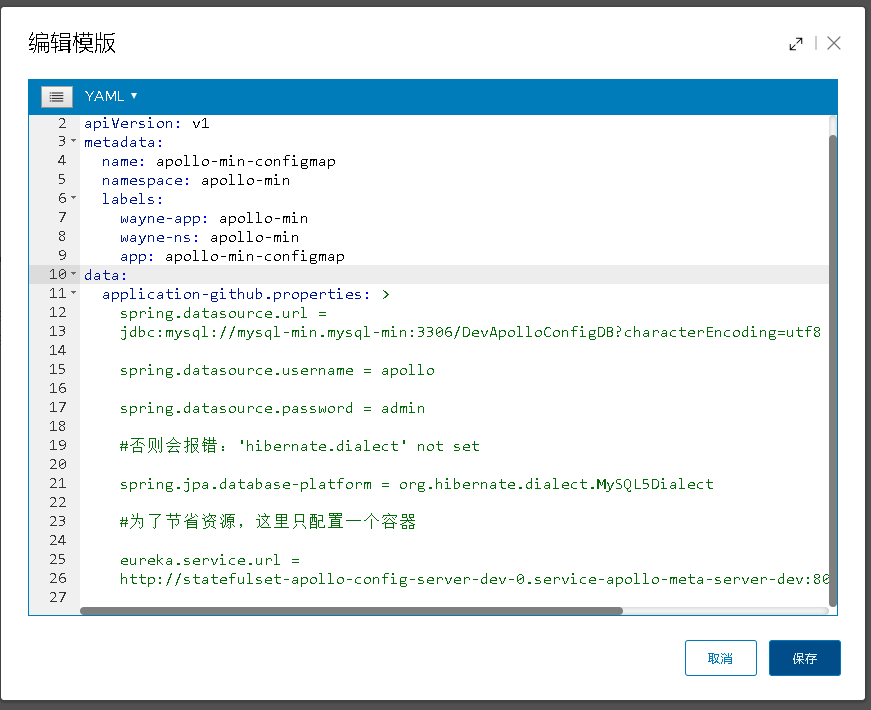


创建configmap，对应wayne中的“创建配置集模板”：

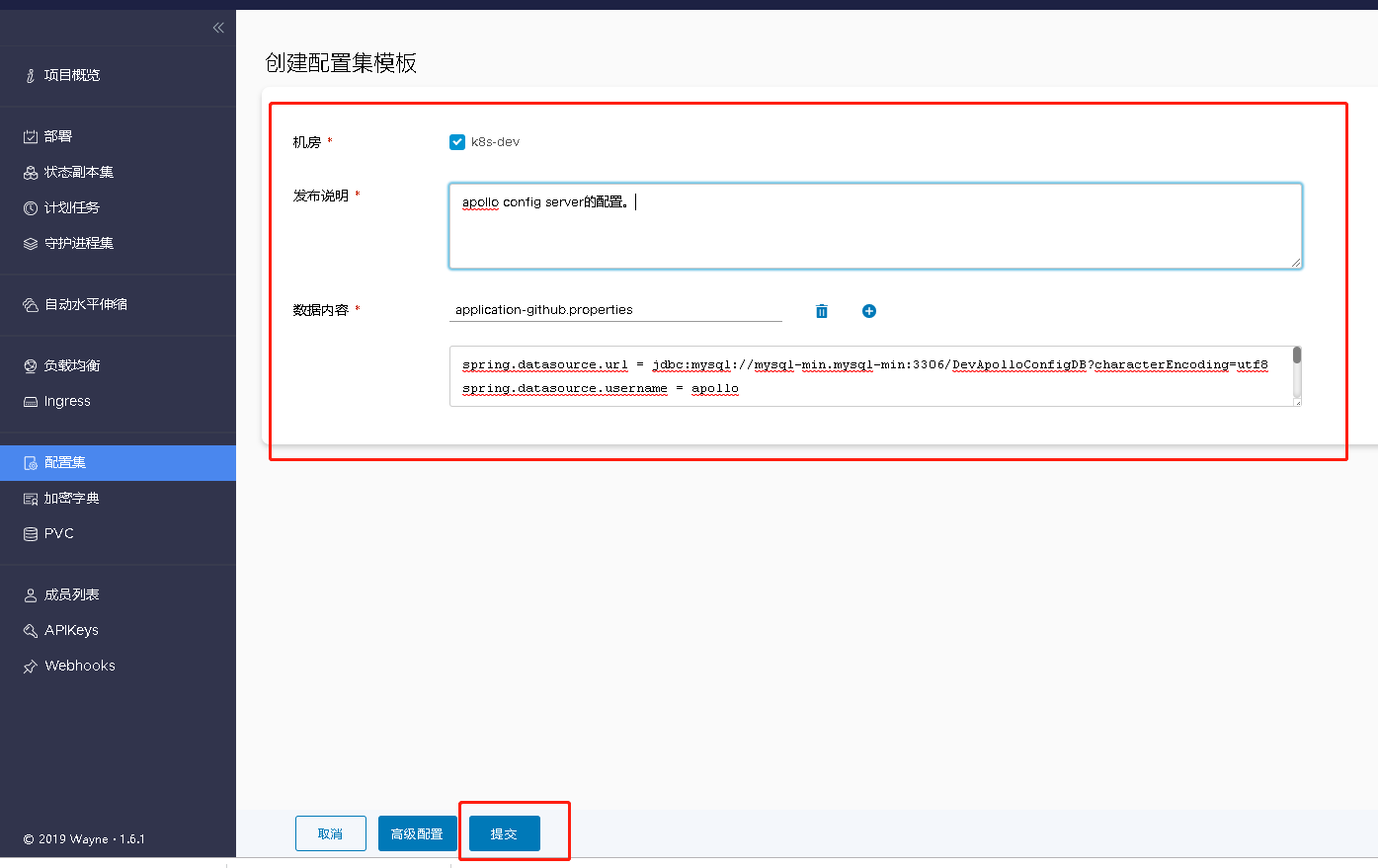


选择高级配置，直接写yaml文件：

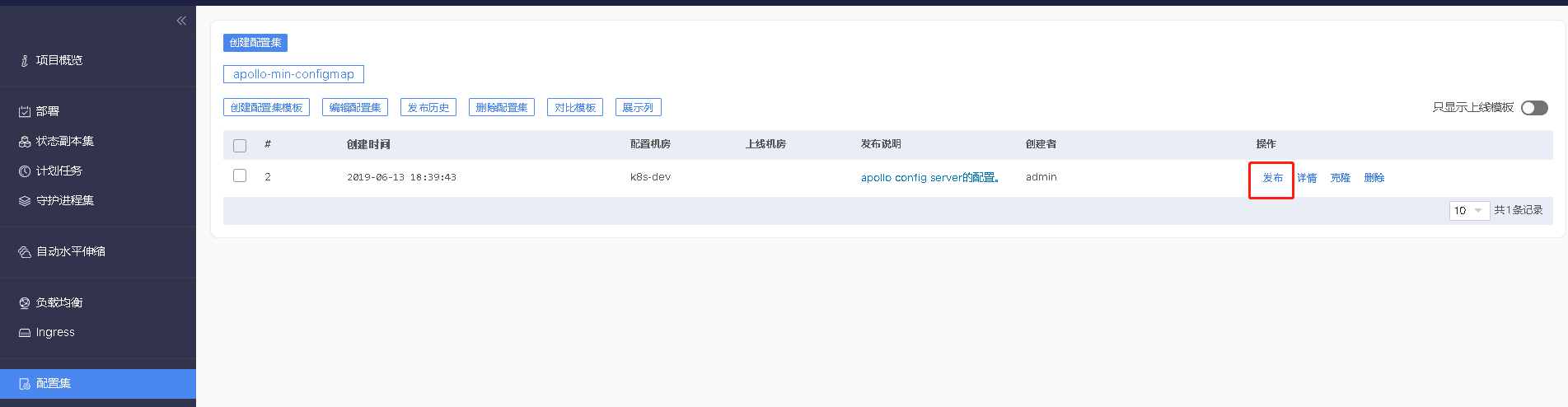




点击保存后：



点击提交完成configmap配置，注意到这里只是将配置放到了wayne自己的配置数据库中，并没有容器化到kubernetes集群中；需要点击发布才会将这个配置发布到kubernetes容器。



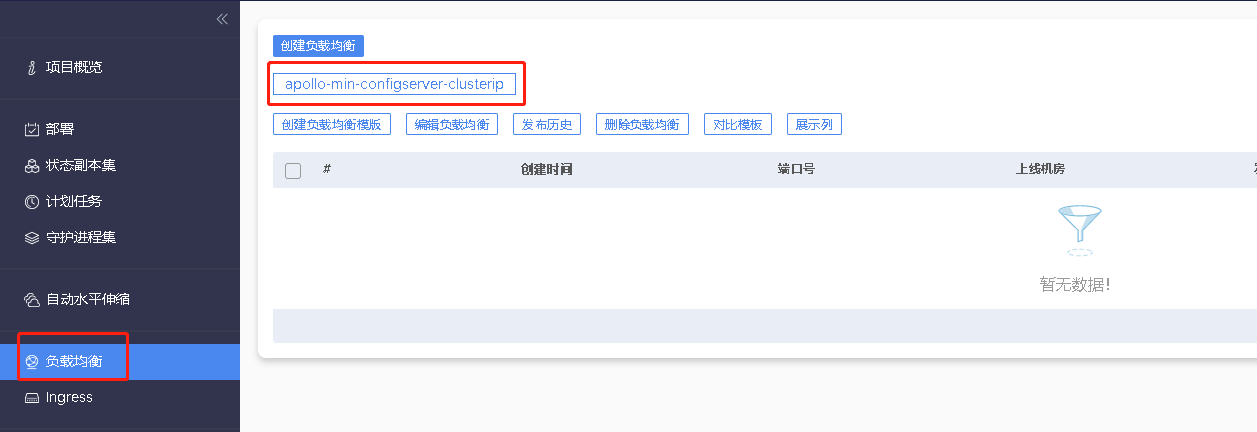
2.创建service

wayne前台选择负载均衡：

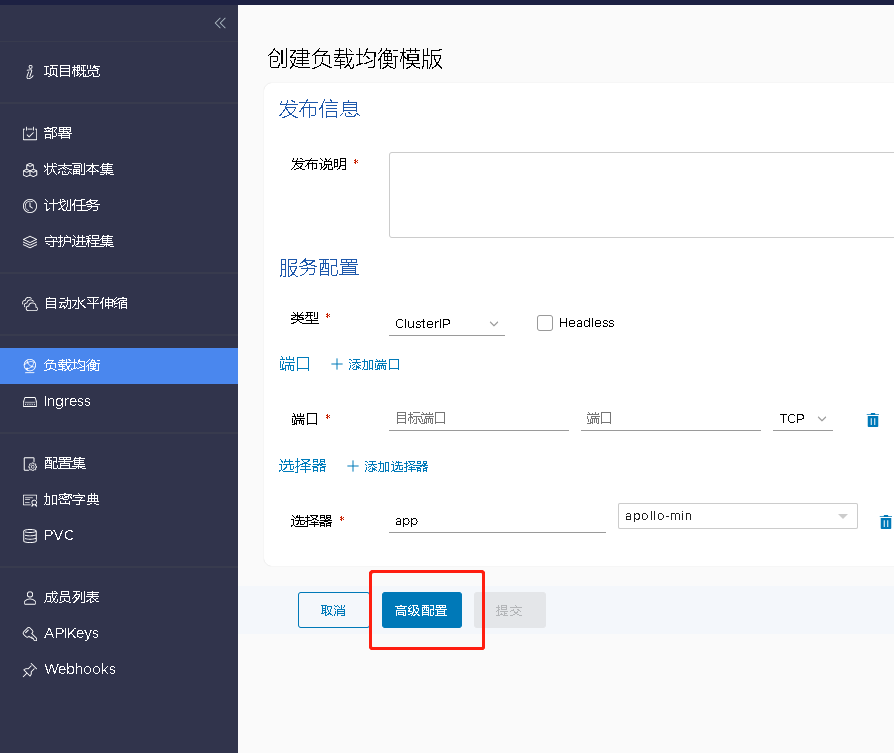


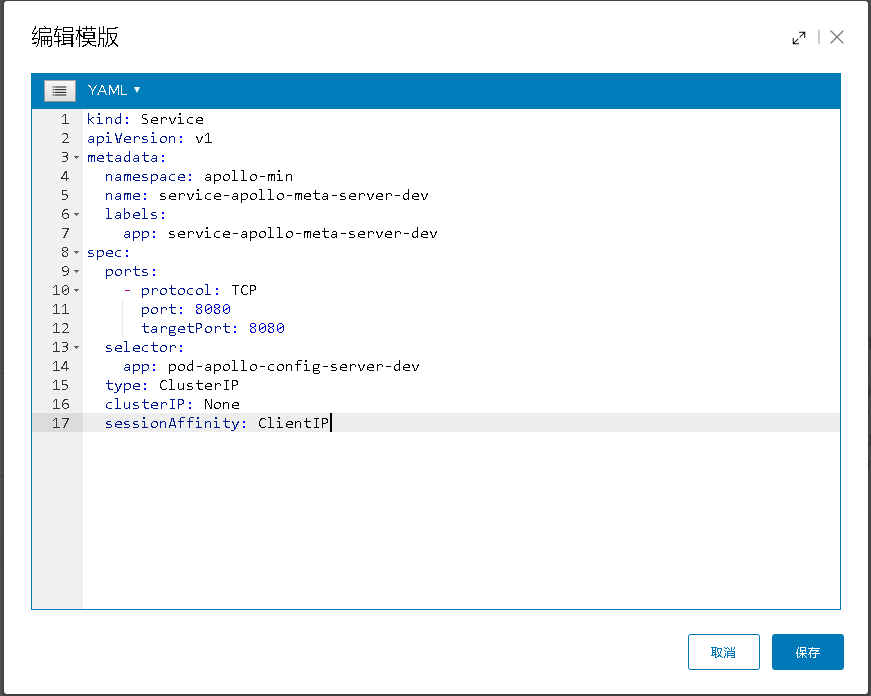


创建负载均衡模板(对应kubernetes中的service)：



同样选择高级配置：





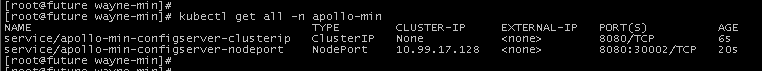


同样方式部署nodeport类型的负载均衡/service，最终结果：



之所以有两个负载均衡，是因为clusterIP类型的service是提供给容器内部服务使用;nodeport类型的service是暴露配置服务给容器外部，这样容器中的apollo可以同时为容器内部和外部的应用提供配置中心的服务。

点击发布，将负载均衡/service部署到kubernetes容器中：



3.创建StatefulSet

在状态副本集中配置后进行发布，流程类似，不再赘述。

