Kubernetes-java-client

结构

Client-java-api

--api:对外调用接口

--common:公共方法

--services:二次封装

Client-java-http:工具封装类

--auth

--http

Client-java-model:实体类

--constant 常量类

--custom 公共数据

--model kubernetes实体类

概念：

1. Namespaces :命名空间，一个虚拟的独立空间，k8s的所有服务、节点等资源都需要建立在这个虚拟空间上，并且各命名空间之间的资源相互独立。系统的命名空间为kube-system。  
   相对于我们的环境来说，一个产品：比如宁家鲜生、招商系统，都分别对应自己的namespaces，其相对的模块及服务都部署在自己的namespaces上.  
   对应model中的com.suneee.kubernetes.model.namespace.V1Namespace
2. Pod：节点，简单看可对应为docker容器，但同时也是配置最多的模组，基本配置信息包含：命名空间，名称，镜像名，(端口名称，端口号)(多个)，限定cpu资源，限定memory资源，环境变量设置(map)等。  
   相对于当前环境就是一个rest模块、一个provider模块都是一个Pod。  
   对应的model为com.suneee.kubernetes.model.pod.V1Pod
3. Deployment：部署，对pod节点进行负载管理，即可以控制同时起多少个pod，更新策略等功能。  
   相对于生产环境上一个rest模块会在不同的服务器上部署多个节点进行负载，这里就由deployment来进行控制。  
   对应的model为com.suneee.kubernetes.model.deployment.AppsV1beta1Deployment
4. Service：服务，为一组相同的pod（deployment控制）提供统一的入口地址，同命名空间下的通过服务名即可互相访问，外部访问需要声明nodePort端口。  
   基本配置信息包含：命名空间，名称，(端口名称，端口号，输出外部端口号:可选)(多个，且对应pod配置)  
   对应的model为com.suneee.kubernetes.model.service.V1Service
5. Ingress：nginx服务，提供域名及内部服务的映射功能。  
   基本配置信息包括：命名空间，名称，域名，端口号，服务名  
   相对于当前环境的nginx配置，只不过这里由k8s自动完成  
   对应的model为com.suneee.kubernetes.model.ingress.V1beta1Ingress

接口调用，在com.suneee.kubernetes.api包下：

1. NamespaceApi：对namespace进行控制的接口
2. DeploymentApi：对Deployment进行控制
3. ServiceApi：对Service进行控制
4. IngressApi：对Ingress进行控制