

# “ 基于 Rook 在 Kubernetes 集群中部署 ChubaoFS Monitor ”

## 项目申请书

### 个人信息

- 姓名：王正茂
- E-mail: [2741403628@qq.com](mailto:2741403628@qq.com)
- 电话：+8617630707261
- Github 账号: Hats-Wang

### 选我的理由：

我现在是西安交通大学计算机系大二在读的学生。我对分布式系统，云计算等特别感兴趣，我认为这是未来的发展的趋势。因此我在学习的空闲时间自己联系学校的老师，加入学校的实验室开始学习和开发 kubernetes 和 Docker。现在可以做到熟练使用 Docker，会简单 kubernetes 的简单命令并且有自己的 kubernetes 集群；同时我也致力于使用已经掌握的技术改善学习者的困境。例如：在学校的编译器实验中，我使用 Docker 配置好环境打包发给同学，避免环境的重复配置，也推广了新技术的普及；这学期我和学长学姐正在组建学校的云原生社团。此外我可以熟练使用 Ubuntu 和 git。我在寒假中学习了 Go 语言，现在正在继续深入学习。

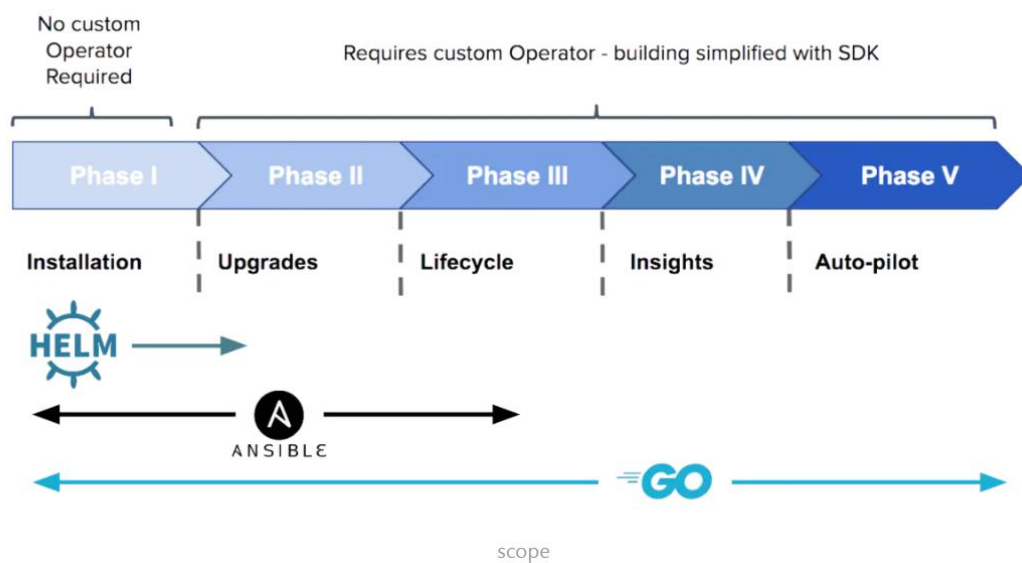
### 项目组件描述：

- prometheus 作为性能监控指标采集模块，实现对主要指标：集群状态相关、节点及磁盘故障相关、数据量、增长量等的监测。

- grafana 作为 prometheus 监控指标的展示前端，通过 prometheus alertmanager 组件进行配置，来实现 ChubaoFS 系统的监控指标报警通知服务。

- consul 用来做动态均衡和带有健康检查的服务发现防止硬编码、提高容灾水平、扩缩容、提高运维效率

- CRD 能持续的监听 K8S 资源对象的变化事件，进行全生命期的监控响应，高可靠的完成部署交付。比老 Helm 发布的底层原理就是调用 K8S 的 apiserver, 逐个 YAML 推送给 K8S, 和我们手动去做没多少差别，所以其弊端就是缺乏对资源的全生命期监控。



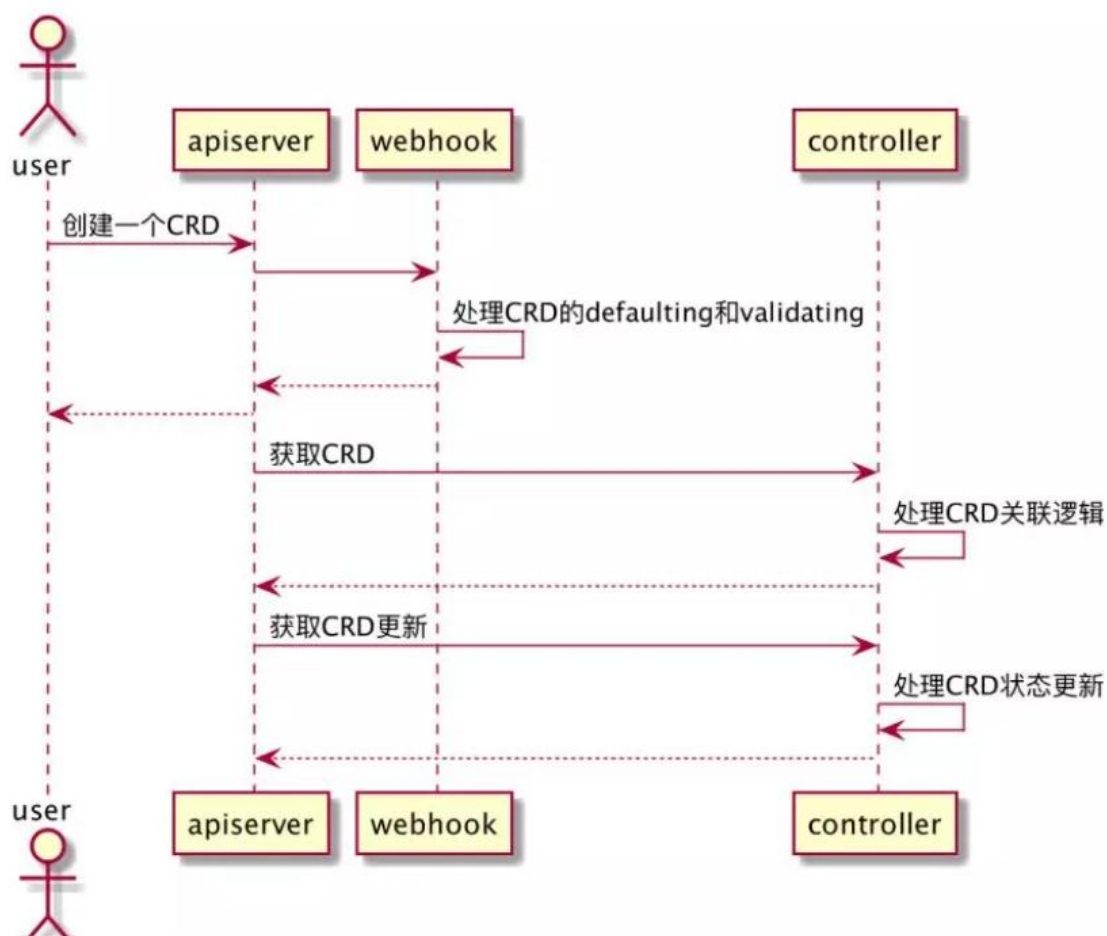
## 项目详细方案:

“基于 Rook 在 Kubernetes 集群中部署 ChubaoFS Monitor”项目是对 Kubernetes 的 client api 进行扩展性开发，以期实现 prometheus、consul、grafana 组件基于 Rook 的方式进行安装, 升级和卸载。本项目参考 rook-ceph 的成功案例, operator 以及 operator framework 进行开发。

Operator 是使用 CustomResourceDefinitions 来以算法和自动化形式封装特定应用

程序的操作知识的控制器。Operator 模式允许我们扩展 Controller 模式，以获得更大的灵活性和更高的表现力。Kubernetes 的运营商越来越多，这种模式正在变成操作复杂分布式系统的主要形式。

主流的 operator framework 主要有两个：kubebuilder 和 operator-sdk。这里采用了 operator-sdk 作为分析的例证。用户创建一个自定义资源 (CRD)；apiserver 根据自己注册的一个 pass 列表，把该 CRD 的请求转发给 webhook；webhook 一般会完成该 CRD 的缺省值设定和参数检验。webhook 处理完之后，相应的 CR 会被写入数据库，返回给用户；与此同时，controller 会在后台监测该自定义资源，按照业务逻辑，处理与该自定义资源相关联的特殊操作；上述处理一般会引起集群内的状态变化，controller 会监测这些关联的变化，把这些变化记录到 CRD 的状态中。



## 项目开发时间计划:

时间点	具体工作
7.1 - 8.15	<ul style="list-style-type: none"><li>● 深入学习 kubernetes 的 operator 和 operator framework</li><li>● 深入学习 rook-ceph 代码实现</li></ul>
7.15 - 8.6	<ul style="list-style-type: none"><li>● 实现 prometheus 基于 Rook 的方式进行安装, 升级和卸载</li><li>● 与老师讨论实现的细节</li></ul>
8.7 - 8.14	<ul style="list-style-type: none"><li>● 中期项目报告</li></ul>
8.16 - 8.20	<ul style="list-style-type: none"><li>● 实现 grafana 基于 Rook 的方式进行安装, 升级和卸载</li></ul>
8.21 - 9.14	<ul style="list-style-type: none"><li>● 实现 consul 基于 Rook 的方式进行安装, 升级和卸载</li></ul>
9.23 - 9.30	<ul style="list-style-type: none"><li>● 与老师交流需要改进的地方</li><li>● 最终项目报告</li></ul>