

作者：李晓辉

联系方式：

1. 微信：Lxh\_Chat

2. 邮箱：939958092@qq.com

# Operator的概念

Operator 其实就是一种运行在 Kubernetes 上的“智能机器人”。你可以把它想象成一个专门帮你管理应用的“小管家”。它通过扩展 Kubernetes 的 API，来自动完成一些复杂的任务，比如安装、升级、备份、恢复，甚至还能自动修复一些问题。简单来说，它就是把一些原本需要人工操作的事情，用代码的方式自动化完成，让你管理应用的时候更轻松。

## Operator 有啥用

- 自动化管理**：这是它最大的好处。比如，你部署了一个数据库应用，平时升级、备份这些事儿，要是手动操作，那可太麻烦了。有了 Operator，它就能自动帮你搞定这些事儿，你啥都不用管，它自己就能运行得很好。
- 保证一致性**：在 Kubernetes 集群里，你可能会部署很多个相同的应用实例。要是手动去配置，很容易出现这儿配置不一样，那儿又不一样。Operator 就能保证这些实例的配置都是一样的，避免出现这种“乱七八糟”的情况。
- 自愈能力**：要是应用出了点小毛病，比如某个容器挂了，Operator 能自动检测到，然后帮你重启它，或者重新调度一个健康的容器，让应用继续正常运行，就像有个“小医生”在时刻盯着一样。
- 扩展功能**：它还能给 Kubernetes 增加一些更高级的功能。比如，有些复杂的监控、日志收集功能，用普通的 Kubernetes 资源定义很难实现，但有了 Operator，就能轻松搞定。

## 在哪儿用

Operator 主要用在 Kubernetes 集群里，尤其是那些需要自动化管理、复杂的应用场景。比如，你在公司里用 Kubernetes 部署了一些重要的应用，像数据库、中间件、微服务这些，都可以用 Operator 来管理。它能帮你减少人工操作的麻烦，提高管理效率，还能让应用更稳定。

### Operator举例

- 数据库管理**：比如 PostgreSQL 数据库，你可以用一个叫 **PostgreSQL Operator** 的工具来管理它。这个 Operator 能帮你自动安装 PostgreSQL 数据库，还能自动帮你做备份。要是数据库出了问题，比如主节点挂了，它还能自动帮你切换到备用节点，保证数据库一直能正常运行。你啥都不用管，它自己就能搞定。

2. **中间件管理**：像 Kafka 这种消息中间件，也有对应的 **Kafka Operator**。它能帮你自动管理 Kafka 集群，比如自动帮你创建主题、调整副本数量，还能监控 Kafka 的性能，要是发现问题，就自动帮你修复。你只需要告诉它你想要什么配置，剩下的事儿它就全包了。
3. **微服务管理**：如果你的项目是用微服务架构写的，有很多个小服务需要管理，那也可以用 Operator。比如 **Istio Operator**，它能帮你自动管理 Istio 服务网格，自动帮你配置服务之间的通信、安全策略这些复杂的玩意儿，让你的微服务运行得更顺畅。

总之，Operator 就像是给 Kubernetes 集群里的应用配了个“超级管家”，帮你把那些复杂、麻烦的事情都自动化搞定，让你管理应用的时候轻松多了！

# Operator vs YAML

## Operator 安装

Operator 其实就是一种特别厉害的应用管理模式，它就像是给 Kubernetes API 加了个“智能大脑”，专门用来搞定那些复杂的应用生命周期管理。很多社区或者第三方开发者会写这种 Operator，专门用来管理特定的应用或者服务。

### 优点

- **自动化管理**：这玩意儿可太省心了！它能自动帮你搞定安装、升级、备份、恢复这些事儿，根本不用你操心，简直不要太方便。
- **一致性**：不管你部署了多少个应用实例，它都能保证这些实例的状态都是一样的，不用担心这儿那儿不一样。
- **自愈能力**：要是应用出了点小毛病，它自己就能发现，然后自动去修复，简直就像有个“小医生”在时刻盯着。
- **扩展功能**：它还能给 Kubernetes 增加一些更高级的功能，让你管理起来更轻松。

### 缺点

- **复杂性**：要是你没经验，写和维护这个东西可能会有点儿费劲，毕竟它有点复杂。
- **依赖性**：因为它是第三方写的，有时候可能会出现版本不兼容，或者更新不及时的情况，这就有点烦了。

## 普通 YAML 安装

用 YAML 文件来部署资源，这应该是 Kubernetes 和 OpenShift 里最基础、最常用的方法了。你只需要写好 YAML 文件，定义好资源的配置，然后用 `kubectl` 或 `oc` 命令一搞，资源就部署好了。

# 优点

- **简单直接**：写 YAML 文件特别简单，要是你只是想快速部署个简单应用，这方法再合适不过了。
- **灵活性高**：你可以随便定义各种资源配置，想怎么配就怎么配，没有任何约束。
- **透明性**：所有的配置都清清楚楚地写在 YAML 文件里，一看就懂，出了问题也方便调试。

# 缺点

- **手动管理**：你得自己手动管理应用的整个生命周期，比如升级、备份、故障恢复这些事儿，一个环节都不能少。
- **一致性难以保证**：要是你在多个环境里手动应用配置文件，很容易就出现配置不一致的情况，特别容易出错。
- **缺乏高级功能**：它没啥自动化和高级管理功能，适合那些比较简单、没啥复杂需求的应用场景。

## 总结

- **Operator 安装**：要是你的应用比较复杂，又想自动化管理，那它绝对是首选。它能帮你搞定自动化操作、自愈和一致性，虽然有点复杂，还依赖第三方，但好处多多。
- **普通 YAML 安装**：要是你的应用比较简单，没啥复杂的管理需求，那就用 YAML 文件吧。它简单、透明，操作起来很方便，就是得自己手动管理，没啥高级功能。

你觉得哪种更适合你呢？

# Operator的核心概念

## 1. Custom Resource（自定义资源）

- 这玩意儿就像是给 Kubernetes API 扩了个“新成员”。你可以自己定义新的资源类型，比如你想在 Kubernetes 里管理一个数据库，就可以定义一个叫 `Database` 的自定义资源。这就相当于你给 Kubernetes 说：“嘿，我这儿有个新东西，你帮我管管！”

## 2. Custom Resource Definition（CRD，自定义资源定义）

- CRD 就是给自定义资源“立规矩”的东西。它告诉 Kubernetes，你定义的这个新资源到底是个啥，能干些啥，怎么操作它。有了 CRD，你就可以像操作 Kubernetes 里的其他资源一样，去创建、读取、更新和删除你定义的自定义资源了。

## 3. Controller（控制器）

- 控制器就像是个“小警察”，它在 Kubernetes 里巡逻，看看资源的状态是不是正常。要是发现有啥不对劲的，它就会自动出手，帮你把事情搞定。在 Operator 里，控制器专门盯着自定义资源，要是发现需要部署、更新或者扩缩容，它就自动帮你操作。

## 4. Operator Lifecycle Manager（OLM）

- OLM 就像是个“大管家”，专门帮你管理 Operator 的事儿。你想安装、升级或者卸载 Operator，OLM 都能帮你搞定，还能确保这些操作靠谱，不会出乱子。

## 5. Cluster Service Version（CSV）

- CSV 就像是 Operator 的“身份证”，它是个 YAML 文件，里面写了这个 Operator 的版本信息、能干啥、需要啥权限等等。OLM 就靠它来管理 Operator 的。

## 6. CatalogSource

- CatalogSource 就像是个“超市货架”，上面摆满了各种可用的 Operator。你想找啥 Operator，就去这个货架上看看，然后通过它来安装你需要的那个。

## 7. InstallPlan

- InstallPlan 就像是个“安装指南”，它告诉 OLM 怎么安装一个 Operator，需要哪些步骤，还有哪些依赖项。OLM 会按照这个指南一步步来，帮你把 Operator 安装好。

## 8. Subscription

- Subscription 就像是你给 Operator 设了个“自动更新提醒”。你可以告诉它，你想跟踪哪个 Operator，要不要自动更新，多久更新一次。这样，你就不怕错过新版本啦。

## 9. Cluster Version Operator (CVO)

- CVO 是 OpenShift 集群里一个特别重要的家伙，它负责管理整个集群的版本升级和配置管理，就像是个“总指挥”。

# Operator 分类

### 1. 基础 Operator

- 基础 Operator 就像是个“专项管家”，它主要管一个特定的应用或者服务。比如，你有个数据库，它就专门帮你管这个数据库；你有个消息队列，它就专门帮你管消息队列。

### 2. 应用 Operator

- 应用 Operator 就像是个“全能管家”，它能管一个完整的应用栈。比如，你有个 Web 应用，它不仅能管前端，还能管后端，甚至连数据库都能一起管，把整个应用都搞定。

### 3. 集群 Operator

- 集群 Operator 就像是个“基础设施管理员”，它负责管 Kubernetes 集群里的一些底层组件和功能。比如，网络插件、存储解决方案这些事儿，它都能帮你搞定。

这些概念听起来是不是有点复杂？其实你把它想象成一个智能的“管家系统”，每个部分都有自己的职责，帮你把 Kubernetes 里的事儿都管得井井有条！

# 使用 Web 控制台安装 operator

打开控制台，并演示安装：File Integrity Operator，控制台URL如下：

<https://console-openshift-console.apps.ocp4.example.com>

本文档在线版本：<https://www.linuxcenter.cn>