

14Kubernetes系列（十四）节点维护升级参考

Kubernetes系列（十四）节点维护升级参考

在 Kubernetes 持续运行的过程中，难免碰到需要升级内核、升级版本，停机维护等情况，此时如何优雅的平滑维护是非常重要的。本文将简单介绍具体的操作方式，仅供参考。

Kubernetes 中的三个命令：`cordon`、`drain` 以及 `delete` 都可以实现 node 的停止调度，也就是后面再创建的 pod 不会继续被调度到该节点上，他们之间最大的区别是暴力程度不一样。

Cordon 控制

- 临时将节点从 Kubermetes 集群隔离
- 影响最小，只会将 node 节点标识为 `SchedulingDisabled` 状态，也就是禁止调度
- 后面创建的 Pod，将不会调度到这个节点
- 原来节点运行的 Pod 不受影响，继续对外服务
- 具体命令：

```
kubectl cordon [node name]
```

- 恢复调度命令：

```
kubectl uncordon [node name]
```

Drain 控制

简介

- 目标：先控制不可调度，然后将原来的 Pod 驱逐、排干
- 首先，将原来的 Pod 驱逐到其他节点重新创建运行
- 然后，将节点标识为 `SchedulingDisabled` 状态，也就是禁止调度
- 具体命令：

```
kubectl drain [node name] --force --ignore-daemonsets --delete-local-data
```

`--force`：就算 Pod 不被 ReplicationController、ReplicaSet、Job、DaemonSet、StatefulSet 等控制器管理，也直接处理；不加 force 参数只会删除该Node 节点上前面提到的几个控制器类型的 Pod，加上之后所有的 Pod 都将删除

`--ignore-daemonsets`：忽略 DeamonSet 管理的 Pod，否则 DaemonSet 被删除后，仍会自动重建

`--delete-local-data` : 删除本地数据, 即使 `emptyDir` 也将删除

- 恢复调度命令:

```
kubectl uncordon [node name]
```

- `drain` 执行的方式是比较安全的, 它会等到 Pod 容器应用程序优雅的停止之后再删除
- 详细的过程: 先在当前节点删除 Pod, 然后再在其他节点创建对应的 Pod。因此为了保证 Drain 驱逐过程中不中断服务, 必须保证要驱逐的 Pod 副本的数量大于 1, 并且采用“反亲和”策略将这些 Pod 调度到不同的节点。这样子可以保证驱逐过程对服务没有影响。

注意事项:

1. 对节点执行内核升级、硬件维护等操作之前, 你可以使用 `kubectl drain` 命令安全地驱逐节点上面的 pod
2. `drain` 的驱逐方式, 将通过容器指定的 `PodDisruptionBudgets` 来优雅的中止容器, 也就是优雅的终止 Pod 中容器的进程
3. `kubectl drain` 会返回成功驱逐的 Pod
4. 后续, 通过物理机断电或者云平台删除虚拟机类型的节点都不影响整个集群

正常情况下, Kubernetes 的 `PodDisruptionBudgets` 配置时是符合 Pod 驱逐的理想情况的, 也就是说 `maxUnavailable` 设置为 0, `maxSurge` 设置为 1:

```
replicas: 2
strategy:
  type: RollingUpdate
  rollingUpdate:
    maxSurge: 1
    maxUnavailable: 0
```

Delete 删除

- 首先, 将原来的 Pod 驱逐到其他节点重新创建运行
- 驱逐过程: 现在当前节点删除 Pod, 然后再在其他节点创建这些 Pod
- Node 节点删除, Master 将会失去对它的控制, 该节点从集群去除
- `delete` 是一种非常暴力的删除节点方式, 驱逐时都是强制干掉容器的进程, 并没有做到优雅终止 Pod, 相比较而言, `drain` 相对比较安全。
- 执行命令:

```
kubectl delete node [node name]
```

欢迎关注我的公众号“云原生拓展”, 原创技术文章第一时间推送。