

「后端」 k8s 部署实操

由 vista-齐浩天创建于六月 05, 2024

- 学习目标
- 参考材料
- 操作步骤
- 结果提交

学习目标

- 学习理解 k8s 中对于资源的基本编排
- 学习理解 k8s 中对于容器服务请求调度的控制，保障服务高可用
- 学习理解自动化迁移框架
- 学习常规 k8s 故障排查

参考材料

- k8s 官方教程: <https://kubernetes.io/zh/docs/tutorials/kubernetes-basics/>
- demo 工程: <https://code.fineres.com/projects/FX/repos/fxpub-nstarter-demo/browse>
- 容器关闭调度: <https://kms.fineres.com/pages/viewpage.action?pageId=118904910>
- 迁移框架: <https://code.fineres.com/projects/FX/repos/fxs-mongodb-migrator/browse>

操作步骤

先参考文档创建 k3s 集群 → [K3S模板操作说明](#)

1. 部署简道云 demo 服务

已知 k3s 集群内已创建 mongodb、redis 和 ingress controller，在 k3s 集群中部署 demo 容器，服务连接信息参考 → [K3S模板操作说明](#)。

- 建立自己的 namespace，并在该命名空间下编排资源 (yaml 文件) 形式
- 参考 fxpub-nstarter-demo 工程，部署 demo 工程容器服务
 - 编写 ConfigMap 配置
 - 编写 Deployment 配置 (副本数: 1, 设置资源请求: 100m 256Mi, 资源上限: 200m 512Mi, 镜像可以使用 Dockerfile 制作或者参考 Jenkinsfile 拉取 harbor 仓库的镜像)
 - 编写 Service 定义
- 通过 ingress 发布服务到公网
 - 使用自定义域名
 - 经由 80 端口以 http 协议发布给公网

i 注意

- 需要自行创建 DB 和管理员，用户的认证数据库为自建 DB
- 自定义域名不要求公网 DNS 解析，可以在本地 hosts 文件中建立域名指向验证访问效果

2. 调整合理的部署配置，实现服务高可用

- 要求保障服务在重启/更新期间，用户侧请求无明显感知。
- 调整容器副本数为 2

已知 demo 容器有以下行为：

- 容器在 3100 端口的两个路径 /liveness_check 和 /health_check 提供了检测探针
- 检测探针在正常状态下，使用 GET 请求访问返回 200 状态
- /health_check 探针在接收到容器即将关闭状态后，会返回 400 状态。

3. 基于 nstarter-mongodb-migrator 迁移框架，实现 demo 项目的数据库 schema 管理工程

- 初始 schema 版本实现自动创建 users 表索引 { username: 1, phone: -1 }
- 实现一次版本迁移兼容
 - 在 users 表中，使用 mongoshell 插入 username 为 demo 的用户记录。

注意

- 参考 Dockerfile 构建镜像，使用 k8s job 任务 实现 schema 初始化和版本迁移，需要注意可能要修改 sys_meta 版本号来控制重新执行历史迁移
- 参考工程：<https://code.fineries.com/projects/FX/repos/nstarter-mongo-migrator/browse>

4. kubectl 故障排查

- 查看简道云 demo 的日志，过滤出 404 信息
- 查看 demo deployment 详细信息，并过滤出 Events 信息
- 筛选出 CPU 负荷最高的节点和集群中内存使用率最高的 pod
- 创建 busybox pod，容器内安装 curl 工具，分别验证通过 service 和 ingress 查看 demo 服务的 metrics 指标
 - 已知 demo 容器在 3100 端口暴露 metrics 指标

结果提交

- 提交 阶段 1 & 阶段 2 中完整的服务编排文件，以及附上访问 demo 截图作为结果验证
- 提交 阶段 3 中的迁移执行日志，以及数据库中记录截图作为结果验证
- 提交 阶段 4 中的排查命令，并附上记录截图作为结果验证

无标签