

基于云原生技术的软件开发 - 大作业

作业说明

开发一个 Spring Boot 应用，并使用云原生功能

1. 功能要求

1. 实现一个 REST 接口（简单接口即可，比如 json 串 {"msg":"hello"}）
2. 接口提供限流功能，当请求达到每秒 100 次的时候，返回 429（Too many requests）
3. 加分项：当后端服务有多个实例的时候（一个 Service 包含若干个 Pod），如何实现统一限流

2. DevOps 要求

1. 为该项目准备 Dockerfile，用于构建镜像
2. 为该项目准备 K8s 编排文件，用于在 K8s 创建服务（service）
3. 准备 Jenkins 持续集成流水线，实现代码构建/单元测试/镜像构建功能（需要写至少一个单元测试）
4. 准备 Jenkins 持续部署流水线，实现部署到 Kubernetes 集群的功能，该流水线的触发条件为持续集成流水线执行成功
5. 注意：持续集成流水线和持续部署流水线也可以合二为一。

3. 扩容场景

1. 为该 Java 项目提供 Prometheus metrics 接口，可以供 Prometheus 采集监控指标
2. 在 Grafana 中的定制应用的监控大屏
3. 使用压测工具（例如 Jmeter）对接口进行压测，观察 Grafana 监控数据

4. 通过 Kubernetes 命令进行手工扩容，并再次观察 Grafana 监控数据
5. 编写 Rolling Update CRD，用于记录 Deployment scale 过程中，ReplicaSet 以及 Pod 的变化：
 - a. 定义 CRD 模型
 - b. 定义 Controller
 - c. Watch Deployment 滚动升级产生的事件，通过 CRD 模型进行记录（RS 的变化，Pod 的变化）
 - d. 提供 API，可以查询从滚动升级开始到结束，可以使用 informer 来 watch 资源的变化，并把变化信息打印出来。

升级流程参考



6. 加分项：使用 K8s HPA 模块根据 CPU 负载做服务的 Auto Scale

分数说明

本次作业占总评 55 分，分数分配如下

1. 功能要求（10 分）

- 1.1 实现接口（2 分）
- 1.2 实现限流功能（8 分）
- 1.3 统一限流（bonus 5 分）

2. DevOps 要求（15 分）

- 2.1 Dockerfile，用于构建镜像（3 分）
- 2.2 Kubernetes 编排文件（5 分）
- 2.3 持续集成流水线（5 分）
- 2.4 持续部署流水线（2 分）

3. 扩容场景（30 分）

- 3.1 Prometheus 采集监控指标 (4 分)
- 3.2 Grafana 定制应用监控大屏 (4 分)
- 3.3 压测并观察监控数据 (2 分)
- 3.4 实现 Rolling Update CRD 以及 Controller (20 分)
- 3.5 Auto Scale (bonus 5 分)

参考

- [K8s crd-operator 入门实践全流程 · 语雀](#)
- [Jenkins+流水线部署应用集成+k8s+具体配置](#)

提交要求

只需提交一份项目说明文档，必须包含以下内容：

1. 限流功能代码说明和截图
2. Dockerfile , K8s 编排文件截图及说明
3. Jenkins 持续集成、持续部署、持续测试配置截图及说明，以及后续验证流水线成功的截图
4. 监控指标采集的配置及说明；Grafana 监控大屏截图
5. 压测工具配置说明，及相应压测监控截图；K8s 手工扩容后，相应压测监控截图

文档内容不限于以上所述，可以任意添加其余说明，使得文档更清晰

一组由一人提交即可，文档内写明组员的信息，姓名和学号

注意：因为只会根据文档评分，文档一定要完整准确清晰地体现所做的工作，请大家对自己负责！

统一提交 pdf 文件，统一文件命名，组号.pdf，如所在组为 1 组，则文件名为：1.pdf

分组

分组方式

自由组合，三人一组（也可以一个人或者两个人一个小组），跟助教登记，后期账号会按照小组生成和发放。

分组截止时间

2022 年 7 月 8 日 23:59 截止

提交时间

2022 年 7 月 31 日 23:59 截止