软工III - Jenkins/Gitlab Runner使用相关要求

1. 简介

- **Jenkins**

> Jenkins是一个开源的、提供友好操作界面的持续集成(CI)工具，起源于Hudson（Hudson是商用的），主要用于持续、自动的构建/测试软件项目、监控外部任务的运行。Jenkins用Java语言编写，可在Tomcat等流行的servlet容器中运行，也可独立运行。通常与版本管理工具(SCM)、构建工具结合使用。

- **Gitlab Runner**

> GitLab Runner is an application that works with GitLab CI/CD to run jobs in a pipeline. GitLab Runner is open-source and written in Go. It can be run as a single binary; no language-specific requirements are needed. You can install GitLab Runner on several different supported operating systems. Other operating systems may also work, as long as you can compile a Go binary on them. GitLab Runner can also run inside a Docker container or be deployed into a Kubernetes cluster.

2. 要求

本课程项目过程中，为了更多地体现出项目的工程性，同时能够让同学们能够掌握常见的软件开发维护工具，要求学生能够掌握并使用 Jenkins **或** Gitlab Runner 进行持续集成和发布。具体要求如下：

Jenkins 要求：

1. 每小组**安装部署一套Jenkins**环境（可在云服务器/本机上进行搭建，推荐使用Docker方式安装）；
    - 如果使用本机搭建，那么请保留必要的截图证明项目在进行持续构建；
    - 如果使用服务器搭建，请保证本学期服务器可访问；
    - 如果有其他 CICD 软件，请联系助教讨论；
2. 掌握使用Jenkins**创建Job实现对项目的自动化构建和部署发布**，实现CI/CD(要求Job均使用Pipeline创建)；
3. 要求Job能够Hook到Gitlab上的提交或变化，**自动触发构建任务**；
4. Jenkins提供了大量优秀方便的插件，掌握插件的安装使用，**要求能够在构建任务执行后查看覆盖率报告**；
5. Pipeline脚本应当被包含在项目设计文档中；

Gitlab Runner 要求：

1. 部署 Gitlab Runner 环境，并连接到 内网Gitlab服务器 **或者** git.nju.edu.cn 上，请截图记录并说明。
    1. git.nju.edu.cn 上自带的 runner 不计入这里的截图，意味着你需要 **额外部署。**
    2. 在部署并截图说明完成后，可以停止运行自己的runner，后续构建允许只使用 git.nju.edu.cn 自带的runner。
2. 掌握使用 Gitlab Runner 进行 CICD 的自动构建和发布方式。
3. 完成以下要求：
    1. 前端：
        1. 能够修改前端依赖下载地址到国内镜像站。

    2. 能够自动下载依赖；
    3. 能够自动构建前端项目；
    4. 将前端的构建产物自动发布到目标服务器上。
2. 后端：
    1. 能够将 maven 的下载地址修改到国内镜像。（推荐使用 repo.nju.edu.cn, 说明文档在 doc.nju.edu.cn ）
    2. 能够自动下载 maven 依赖。
    3. 能够对项目进行测试，并产生 html 形式的测试报告。
    4. **将测试报告部署到自己的服务器上。**
    5. 能够对 maven 项目进行构建，将构建产物自动发布到目标服务器上。
    6. 能够远程重新启动目标服务器上的后端进程。
3. 测试分支：
    1. 小组需要至少两个部署：master/main 分支，当该分支代码变动时自动部署到生产服务器上；其他分支，当其他分支代码构建完成后，**手动**触发部署到测试服务器上。
    2. 生产服务器和测试服务器可以是同一个服务器的不同端口。
    3. 对应的，在前后端，小组至少需要两个不同的配置文件：spring-boot-starter-actuator 接口的配置、日志等级的配置等。
    4. 后端的测试报告在所有分支都需要。
4. 如果决定选用 git.nju.edu.cn + gitlab runner,那么请填写 此统计表。

3. 检查

在项目检查时，将对每组同学的Jenkins进行检查，主要检查如下几点：

- 是否能够正常登录运行
- 检查构建记录，代码是否持续部署到服务器上
- 查看自动触发
- 检查覆盖率报告
- 检查Pipeline脚本
- 每次检查截止点的代码要求必须可以构建和运行

如果使用了 gitlab runner，那么检查点如下：

- 检查构建记录，代码是否持续部署到服务器上；
- 检查自动触发和手动触发；
- 检查生产服务器和测试服务器的运行情况；
- 检查部署的测试报告；
- 检查Pipeline脚本
- 每次检查截止点的代码要求必须可以构建和运行
- 手动触发一次测试服务的构建，检查构建结果。

4. 教程与答疑

(1) 教程

关于Jenkins的安装、使用，可以查看Jenkins用户手册进行学习和参考；

关于Gitlab Runner的安装、使用，可以查看Gitlab Runner官方手册进行学习和参考；

(2) 答疑

在Jenkins/Gitlab Runner安装、使用过程中，遇到难以自行解决的问题，可采取以下方式进行答疑:

- 前往**Moodle讨论区**中的**Jenkins/Gitlab Runner答疑专用帖**下跟贴提问，会有助教针对问题进行解答，同时可以方便其他同学参考；

5. 常见问题说明

- **Q:我的服务器是外网服务器，无法访问内网Gitlab，怎么办?**

  A:可参考如下操作方法：

  1. 注册https://git.nju.edu.cn/ 的账号，并在此账号中创建与内网仓库相对应的空仓库（请注意，创建时不应勾选自动创建READMEmd）。
  2. 在内网Gitlab（http://172.29.4.49/）的自己的账号所对应的仓库中进行如下操作：
     - 进入仓库的Settings->Repository->Mirroring repositories（请注意，如果没有这些选项，则说明仓库尚未初始化）。
     - Git repository URL填写git.nju中的空仓库URL（如https://username@gitlab.company.com/group/project.git）。
     - Mirror direction选择Push，Authentication method选择Password，密码填写自己git.nju的密码（不是内网Gitlab密码）。
     - 点击Mirror repository，创建镜像。
     - 至此，当你对内网Gitlab进行Push时，内网Gitlab会自动向git.nju进行Push，后续Hook可直接在git.nju上进行。（亦可点击Update now进行手动Push更新。）
     - 如果出现错误（如：GitLab: You are not allowed to force push code to a protected branch on this project），则需要在git.nju的对应仓库的Settings->Repository->Protected branches中取消目标Push分支的protected属性。
- **Q:我可以使用 Github Actions 吗?**

  A:鉴于 Github 的访问连通性问题，不可以。

- **Q:我可以使用其他代码托管平台吗?**

  A:请保证 http://172.29.4.49/ 上必须有小组的**完整的提交记录**，不可以只有最终代码的一次提交。如果使用其他托管平台，请确保在作业截止日期前，项目的可见性为 private。