# 项目说明

王红元 coderwhy


实力IT教育

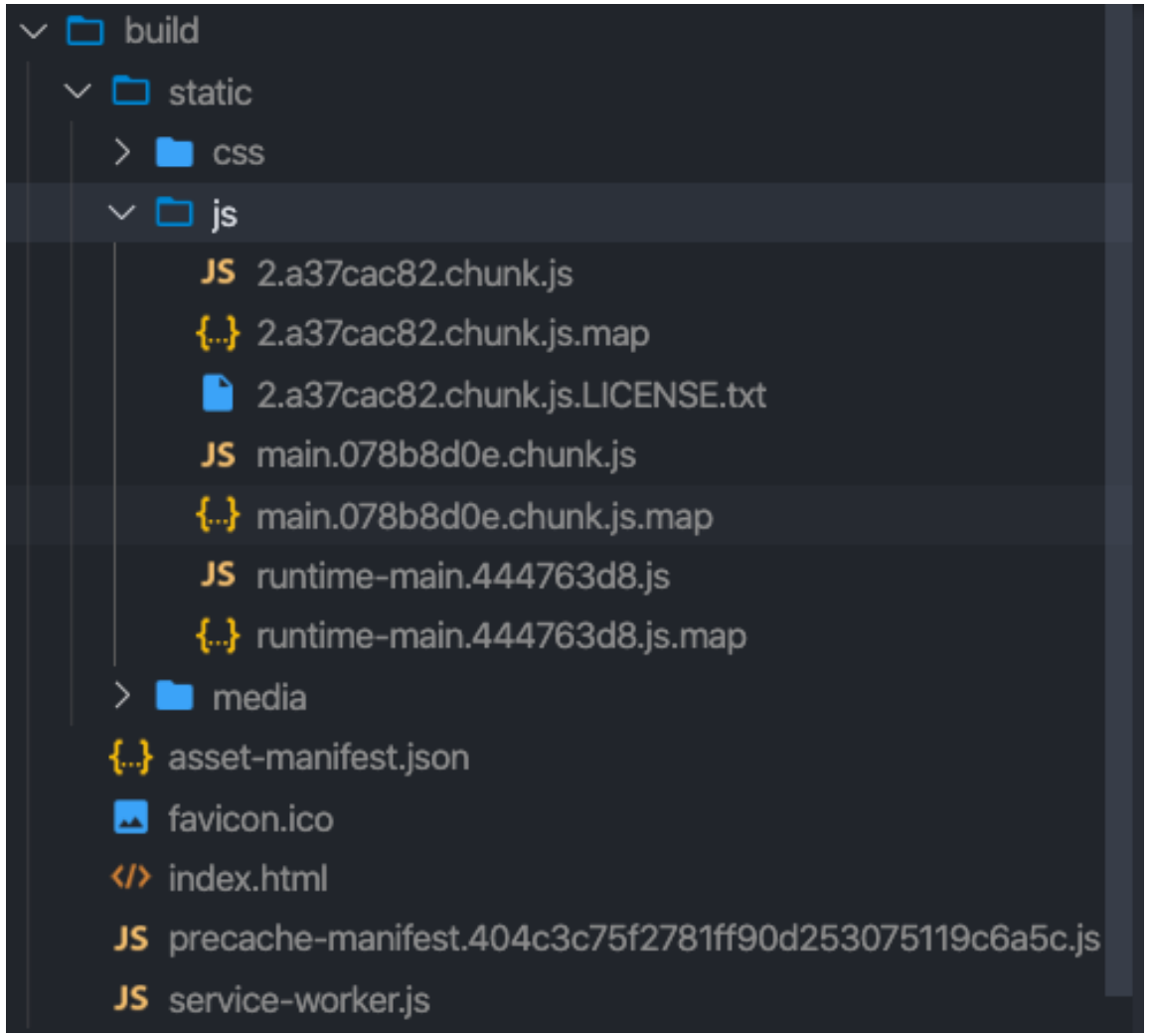
# react项目打包

- 对于使用脚手架创建的项目，打包是一件非常容易的事情：

```
yarn build
```

- 其他文件没有太多要解析的，我们看一下js文件：
- [hash].chunk.js
  - 代表是所有依赖的第三方库， *vendor(第三方库)* 的代码；
- main.[hash].chunk.js
  - 我们自己编写的应用程序代码；
- runtime~main.[hash].js
  - Webpack runtime逻辑的chunk；
  - 用于加载和运行你的应用程序；

- 思考：随着业务逻辑代码越多，main会变得非常臃肿；



# React打包优化

- 很多模块，其实没有必要一开始就进行加载，会影响首屏加载速度；
- 我们可以让某些组件用到时再加载（懒加载）；
- 如何可以让一个组件进行懒加载呢？
  - 使用react给我们提供的lazy函数即可；

```
const HYFriend = React.lazy(_ => import("../pages/friend"));
const HYMine = React.lazy(_ => import("../pages/mine"));
```

- 但是，修改后运行代码会报错：
  - React希望我们提供一个在组件没有加载出来之前，显示的组件；
  - 我们可以通过Suspense组件传入一个fallback属性；

```
<Suspense fallback={<div>loading</div>}>
  {renderRoutes(routes)}
</Suspense>
```

# 项目手动部署

- 1.有一台服务器：通常会选择阿里云、华为云、腾讯云等都可以。
  - 服务器安装操作系统，通常会安装centos，比较稳定；
- 2.服务器中安装Nginx服务
  - 可以借助于yum工具来安装Nginx；
- 3.配置Nginx的代理
  - 设置Nginx的权限为root；
  - 可以将配置文件进行分离；
  - 单独配置conf.d文件夹下的配置文件；

```
# user nginx;
user root;
```

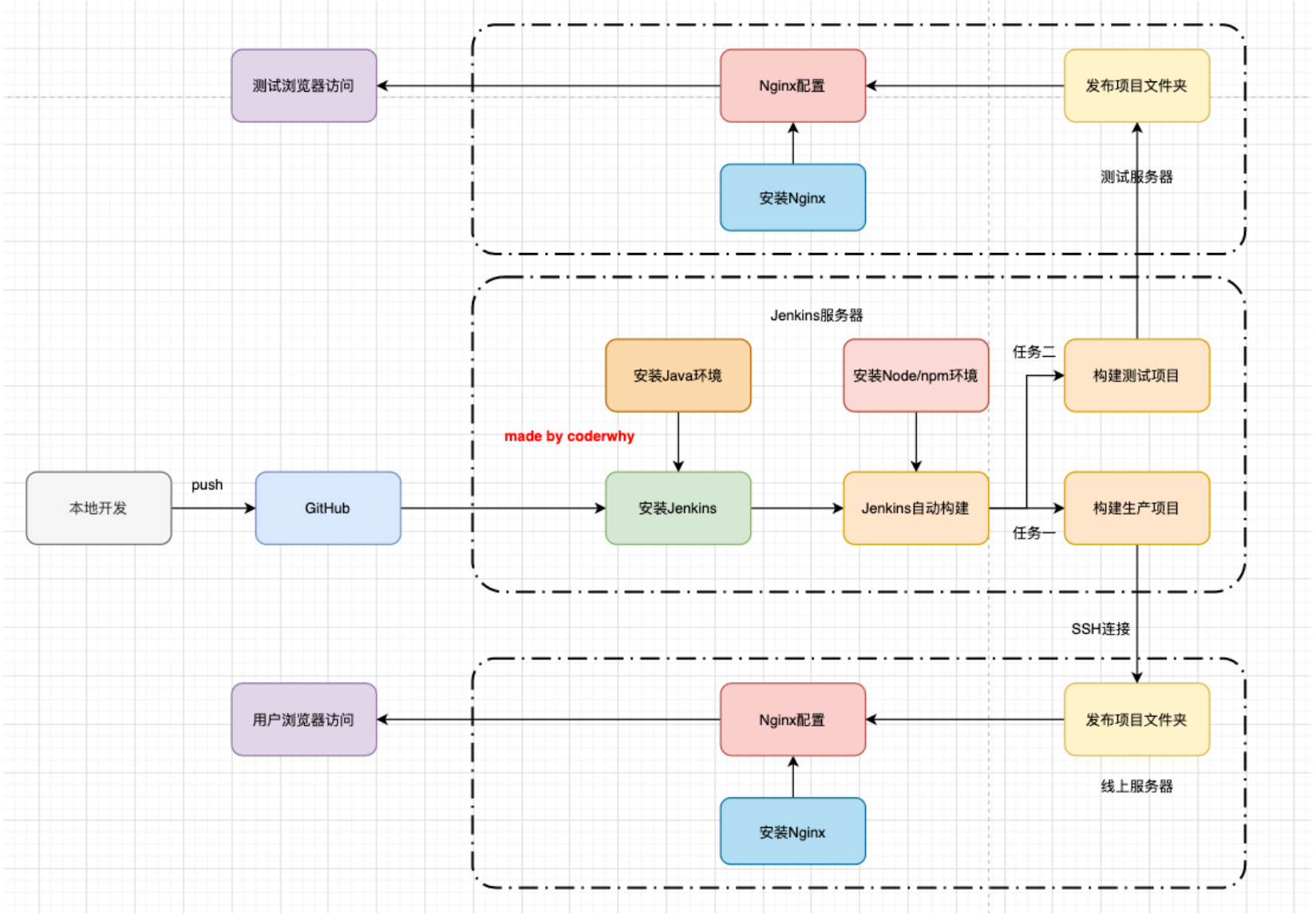
```
# Load modular configuration files from the /etc/nginx/conf.d directory.
# See http://nginx.org/en/docs/ngx_core_module.html#include
# for more information.
include /etc/nginx/conf.d/*.conf;
```

```
server {
    listen       7878;
    server_name  _;
    # root         /usr/share/nginx/html;

    # Load configuration files for the def
    # include /etc/nginx/default.d/*.conf;

    location / {
        root /root/music/build;
        index index.html;
    }
}
```

# 自动化部署的流程


测试浏览器访问
Nginx配置
发布项目文件夹
安装Nginx
测试服务器
Jenkins服务器
安装Java环境
安装Node/npm环境
任务二
构建测试项目
made by coderwhy
本地开发
push
GitHub
安装Jenkins
Jenkins自动构建
构建生产项目
任务一
SSH连接
用户浏览器访问
Nginx配置
发布项目文件夹
线上服务器
安装Nginx

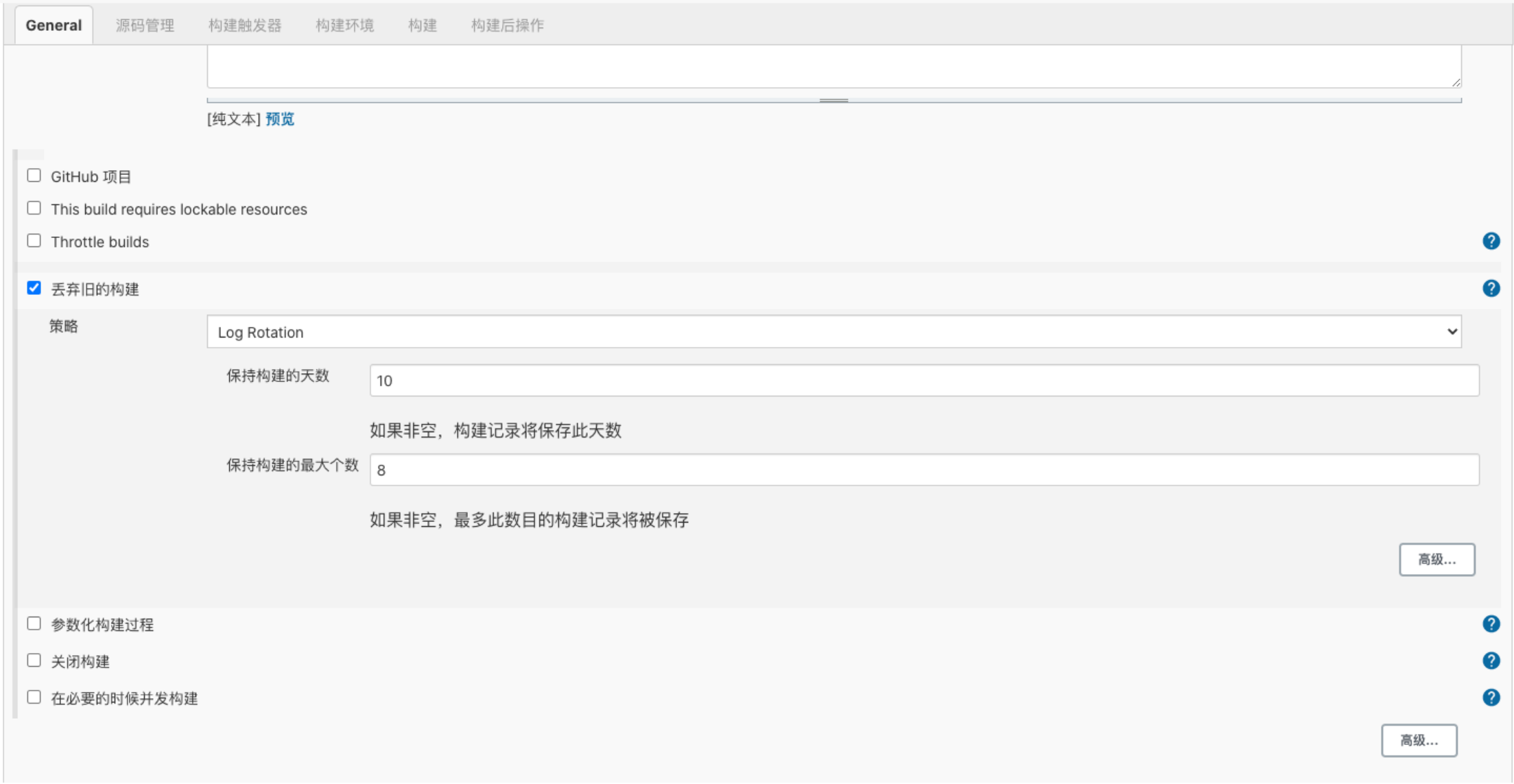
# Jenkins服务器安装

- 1.安装Java环境
  - 运行Jenkins需要依赖Java环境
- 2.Jenkins的安装
  - 我们使用Jenkins来完成自动化打包、部署过程；
  - 可以设置数据源后、通过yum工具安装；
- 3.安装Git/SVN
  - Jenkins需要通过Git或者SVN从代码仓库中下载代码；
  - 可以通过yum工具安装
- 4.Node环境
  - Web项目目前打包都需要依赖node；
  - 我们在服务器进行自动化打包，必然需要有node环境；
  - 提示：这里可以通过yum工具安装，之后通过工具n进行node版本升级；

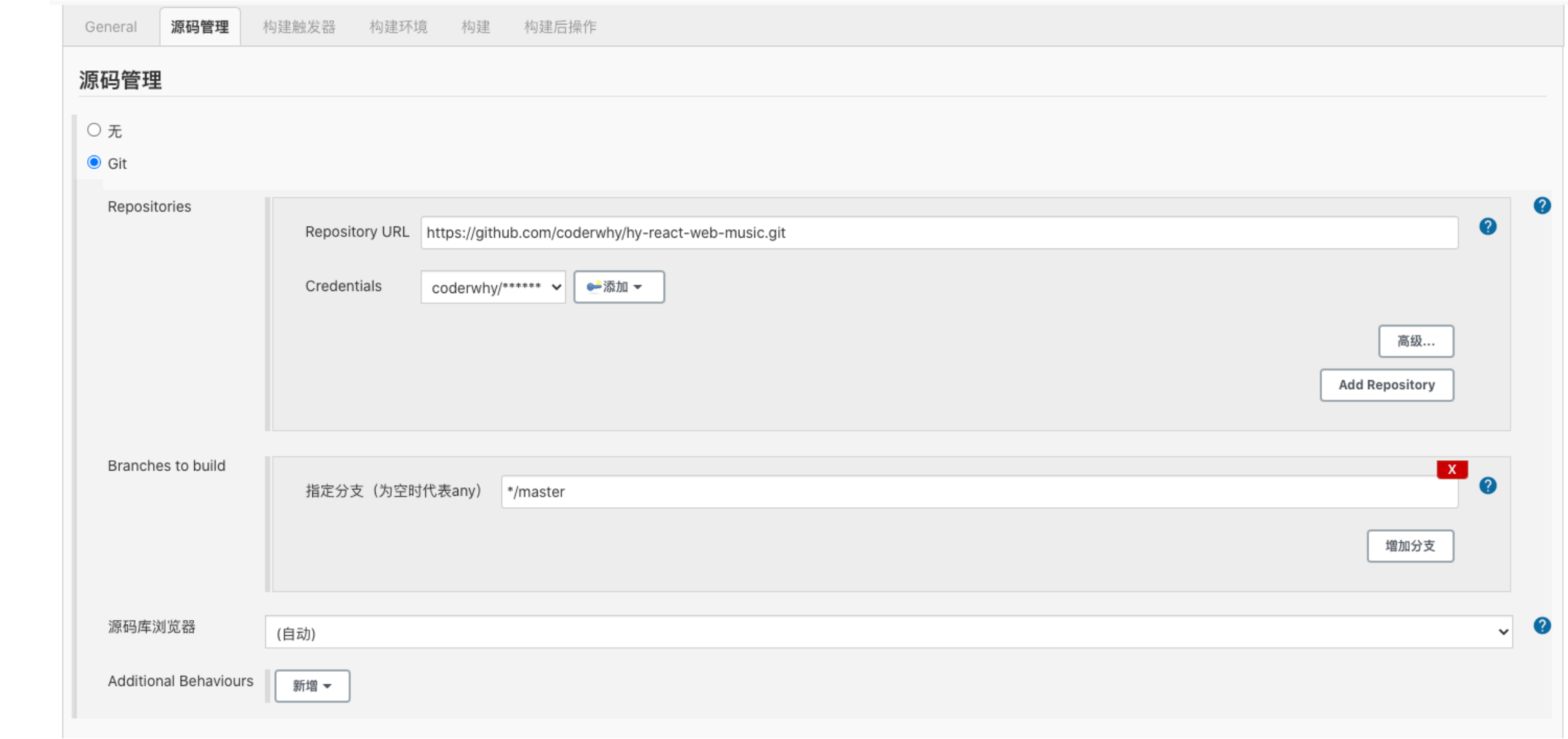
# 配置Jenkins任务

- 登录Jenkins管理后台：
- 第一次会让我们安装插件（推荐插件直接安装即可），创建用户；
- 创建Jenkins任务（根据视频内容演练即可）

# General配置


General
源码管理
构建触发器
构建环境
构建
构建后操作
[纯文本] 预览
GitHub 项目
This build requires lockable resources
Throttle builds
丢弃旧的构建
策略
Log Rotation
保持构建的天数
10
如果非空，构建记录将保存此天数
保持构建的最大个数
8
如果非空，最多此数目的构建记录将被保存
高级...
参数化构建过程
关闭构建
在必要的时候并发构建
高级...

# 源码管理配置


General
源码管理
构建触发器
构建环境
构建
构建后操作
源码管理
无
Git
Repositories
Repository URL
https://github.com/coderwhy/hy-react-web-music.git
Credentials
coderwhy/******
添加
高级...
Add Repository
Branches to build
指定分支（为空时代表any）
*/master
X
增加分支
源码库浏览器
(自动)
Additional Behaviours
新增

# 轮训SCM规则

- 第一颗*表示分钟minute：取值0-59，第几分钟执行
- 第二颗*表示小时hour：取值0-23，第几小时执行
- 第三颗*表示日day：取值1-31，第几日执行
- 第四颗*表示月month：取值1-12，第几月执行
- 第五颗*表示星期week：取值0-7，每周第几天执行

```
#每半小时构建一次OR每半小时检查一次远程代码分支，有更新则构建
H/30 * * * *

#每两小时构建一次OR每两小时检查一次远程代码分支，有更新则构建
H H/2 * * *

#每天凌晨两点定时构建
H 2 * * *

#每月15号执行构建
H H 15 * *

#工作日，上午9点整执行
H 9 * * 1-5

#每周1,3,5，从8:30开始，截止19:30，每4小时30分构建一次
H/30 8-20/4 * * 1,3,5
```

# 触发器配置


构建触发器
触发远程构建 (例如,使用脚本)
其他工程构建后触发
定时构建
GitHub hook trigger for GITScm polling
轮询 SCM
日程表
H/15 * * * *
上次运行的时间 Friday, August 14, 2020 7:12:56 PM CST; 下次运行的时间 Friday, August 14, 2020 7:12:56 PM CST.
忽略钩子 post-commit

# 构建环境配置

## 构建环境

- [ ] Delete workspace before build starts
- [ ] Use secret text(s) or file(s)
- [ ] Provide Configuration files
- [ ] Send files or execute commands over SSH before the build starts
- [ ] Send files or execute commands over SSH after the build runs
- [ ] Abort the build if it's stuck
- [ ] Add timestamps to the Console Output
- [ ] Execute shell script on remote host using ssh
- [ ] Inspect build log for published Gradle build scans
- [x] Provide Node & npm bin/ folder to PATH

NodeJS Installation: Node12.18.3

Specify needed nodejs installation where npm installed packages will be provided to the PATH

npmrc file: - use system default -

Cache location: Default (~/.npm or %APP_DATA%\npm-cache)

- [ ] With Ant

# 构建配置

## 构建

执行 shell

命令

```
pwd
ls

node -v
npm -v
git --version
java -version
echo 'yarn版本号:'

echo '构建的版本号:'${BUILD_NUMBER}

npm install
npm run build

pwd


echo '----- 以上的列出的文件是 jenkin 服务 workspace 中 testweb 目录下的文件-------'
```

查看 可用的环境变量列表

高级...

增加构建步骤

# 构建后操作（需要配置 publish over shh）