

基于jenkins+gitlab+harbor 自动化构建 k8s 发布实现原理和操作方案

- 一、现有项目代码库情况
- 二、Jenkins 创建的 item 情况
- 三、Jenkins CI/CD 自动部署实现功能和操作使用
- 四、Jenkins CI/CD 自动部署原理
- 五、K8s 支持现有域名情况
- 六、Harbor 使用说明
- 七、附图：飞书报警截图 和 高亮显示截图 及 k8s 截图

一、现有项目代码库情况

front/Package_BSI_PC_UI	0 ☆
front 为 nodjs 前端 各个项目代码库	
dotnet/SJZY.Package.Operation	0 ☆
front/Package_OutCustome_PC_UI	0 ☆
dotnet/SJZY.Package.Api	0 ☆
dotnet 为 后端 各个项目代码库	
dotnet/SJZY.Package.Task.Api	0 ☆
dotnet/SJZY.Package.Output.Api	0 ☆
front/Package_Shopman_Web_UI	0 ☆
front/Package_Label	0 ☆

二、Jenkins 创建的 item 情况

item 情况和现在项目开发定义 git 库名一样，如下

Dashboard >

+ 新建任务

👤 用户列表

📁 构建历史

🕒 项目关系

🔍 检查文件指纹

⚙️ 系统管理

👤 我的视图

📁 新建视图

构建队列

▼

队列中没有构建任务

构建执行状态

▼

🖨️ master

dotnet

front

所有

+

S	W	名称 ↓	上次成功
✓	☁	Packet_BSI_PC_UI 前端	20 分 #15
✓	☁	Packet_OutCustome_PC_UI	16 小时 #2
✓	☁	Packet_Shopman_Web_UI	7 分 1 秒 #4
✓	⚙️	SJZY.Package.Api 后端	16 小时 #3
✓	🔌	SJZY.Package.Output.Api	19 小时 #8
✓	⚙️	SJZY.Package.Task.Api	20 小时 #4
✓	🚀	test	1 天 1 小时 #242

三、Jenkins CI/CD 自动部署实现功能和操作使用

pipeline 基本实现功能：

- 1、可以发布任何分支，或标签、或每次提交 id 。
- 2、 pipeline 通过调用 webhook 接口，再通过飞书机器人发送 jenkins 的构建信息到相应的飞书群 (如正在构建的前后端代码名, 第一几次构建, 构建成功或失败等)。
- 3、 Jenkins 每次发布的日志高亮显示，便于排查问题。
- 4、 回滚发布支持 docker 镜像回滚 k8s 方式。
- 5、 回滚发布支持传统基于 git tag 回滚、分支回滚、和 git 每次提交 id 回滚。

操作使用（前后端一样啊）：

Pipeline Packet_BSI_PC_UI

需要如下参数用于构建项目:

IMPLEMENT

部署方式: deploy正常发布 或 rollback回滚

deploy

ROLLBACK_OPTION

回滚方式选择: 只有IMPLEMENT部署方式为rollback时生效, git_rollback 按git分支、版本号、tag回滚, k8s_rollback 按k8s image 的 tag标签 回滚。

git_rollback

BRANCH

git分支选择: 默认: test ,可选择其他 uat master develop

test

IMAGE_TAG

k8s_rollback回滚: 输入需要回滚IMAGE 便签 如: 2022-09-06-19-27 (不输入默认为latest, 执行k8s_rollback回滚操作,可登录harbor服务器查找标签)

latest

TAG

git_rollback回滚: 输入需要回滚标签 (不输入默认为 BRANCH 参数变量分支最新版本)

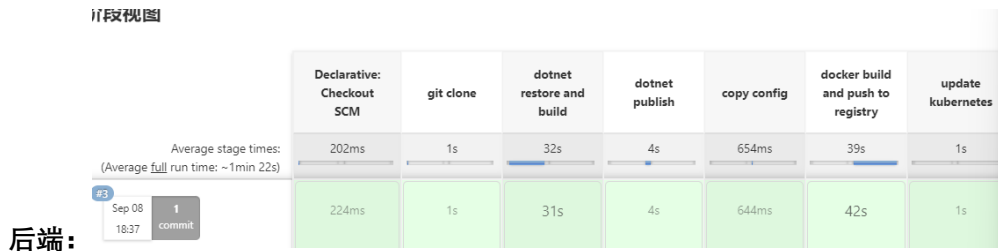
\$BRANCH

开始构建

四、Jenkins CI/CD 自动部署实现原理

实现原理图（前后端一样啊）：





后端:

实现原理（前后端一样啊）:

正常发布:

Jenkins 读取发布参数 IMPLEMENT (默认 **deploy**) ---分支参数 BRANCH---- pipeline 读取 git 代码 ---- copy config(git 实现的啊, 由于目前没有类似微服务配置中心)----编译 (pnpm/dotnet) ---- build docker----上传 **harbor**----update k8s

传统回滚发布:

Jenkins 读取默认发布参数 IMPLEMENT (**git_rollback**) ----分支 TAG 参数 TAG---- pipeline 读取 git 代码 ---- copy config(git 实现的啊, 由于目前没有类似微服务配置中心)----编译 (pnpm/dotnet) ---- build docker----上传 **harbor** ---update k8s

K8s 回滚发布（回滚比传统快很多）:

Jenkins 读取默认发布参数 IMPLEMENT (**k8s_rollback**) --- 镜像标签 IMAGE_TAG---拉取 **harbor** 镜像---- update k8s

五、K8s 支持现有域名情况

支持: 线上环境域名、uat 域名、测试域名等

说明: 只需要配好域名 dns 解析或 hosts, 解析到 k8s 的 vip: 192.168.80.94 即可

六、Harbor 使用说明

前端在 front, 后端在 donet ,登录以后可以看总的使用情况



再次点击 front, 或 dotnet, 可以看到 docker 镜像包个数

< 项目

访问级别 私有

已使用的配额 212.73MiB of 不设限

概要 镜像仓库 成员 标签 扫描器 P2P 预热 策略 机器人账户 Webhooks 日志 配置管理

× 删除 推送命令

名称	Artifacts	下载数	最新变更时间
front/packet	1	2	2022/9/8 下午2:50
front/packet_label	1	1	2022/9/4 下午5:06
front/packet_shopman_web_ui	2	3	2022/9/9 上午10:53
front/packet_bsi_pc_ui	9	13	2022/9/9 上午11:20
front/packet_outcustome_pc_ui	4	7	2022/9/9 上午11:01

页面大小 15 1 - 5 共计 5 条记录

七、附图：飞书报警截图 和 高亮显示截图 及 k8s 截图

飞书报警截图：前端 pipeline 自动化要 8-11 分钟，后端只有 1 分多钟



每个 pipeline stage 高量显示:

```
SJZY.Package.Api > #2
{ "stage": "dotnet restore and build", "status": "success" }
[Pipeline] sh
+ echo -e 'stage: dotnet restore and build'
stage: dotnet restore and build
[Pipeline] git
The recommended git tool is: NONE
using credential 33892de7-5621-4dcd-9e89-4...
> git rev-parse --resolve-git-dir /var/lib/...
Fetching changes from the remote Git repos...
> git config remote.origin.url http://git...
Fetching upstream changes from http://git...
```

类似这种高亮显示

K8s 可以看到刚刚部署更新的 pod:

```
[yunwei@192_168_80_82 ~]$ kubectl get pods -n sjzy
NAME                                READY   STATUS    RESTARTS   AGE
packet-bsi-pc-ui-cff4fb6f-pq24j     1/1     Running   0           51s
packet-label-5dc59d75f5-brdcx       1/1     Running   0           3d
packet-outcustome-pc-ui-5cd9974d55-8hzqb 1/1     Running   0           3m33s
packet-shopman-web-ui-7d6f477fc7-gmn4z 1/1     Running   0           11m
sjzy-package-api-54b9dfcb76-v4vvc    1/1     Running   0           16h
sjzy-package-output-api-946d5b5d6-vngwt 1/1     Running   0           19h
sjzy-package-task-api-66cd5b9fb4-9v4mv 1/1     Running   0           20h
```

本文是 [巧妙绝情](#) 一个字一个图打出来，参考了好多资料，感谢他们的分享，基于 open source 分享精神，转载请注明出出。支持我，请用力 点击 [巧妙绝情](#) 谢谢