

小包系统持续自动化总结

- ci/cd 原理
- 现在小包系统生产环境架构
 - 1、 逻辑架构图
 - 2、 物理架构图
 - 3、 架构问题
- 现有小包系统上线发布情况及问题分析，解决方案
 - 1、 常见的几种发布分析
 - 2、 目前发布方式情况及问题分析
 - 3、 解决方案
- CI/CD 架构规划
 - 1、 测试、UAT 环境架构规划
 - 2、 生产环境 ACK 架构规划
- 虚拟机实现部署 CI/CD 架构支持功能说明
 - 1、 支持存储编排
 - 2、 基于镜像滚动升级
 - 3、 一键回滚
 - 4、 故障自我修复
 - 5、 服务发现和负载均衡
 - 6、 支持配置管理
 - 7、 支持多种发布
- 所有基础服务截图说明
 - 1、 Gitea 代码库截图
 - 2、 jenkins 截图
 - 3、 harbor 截图
 - 4、 config 配置中心截图
- 演示

CI/CD 实现原理

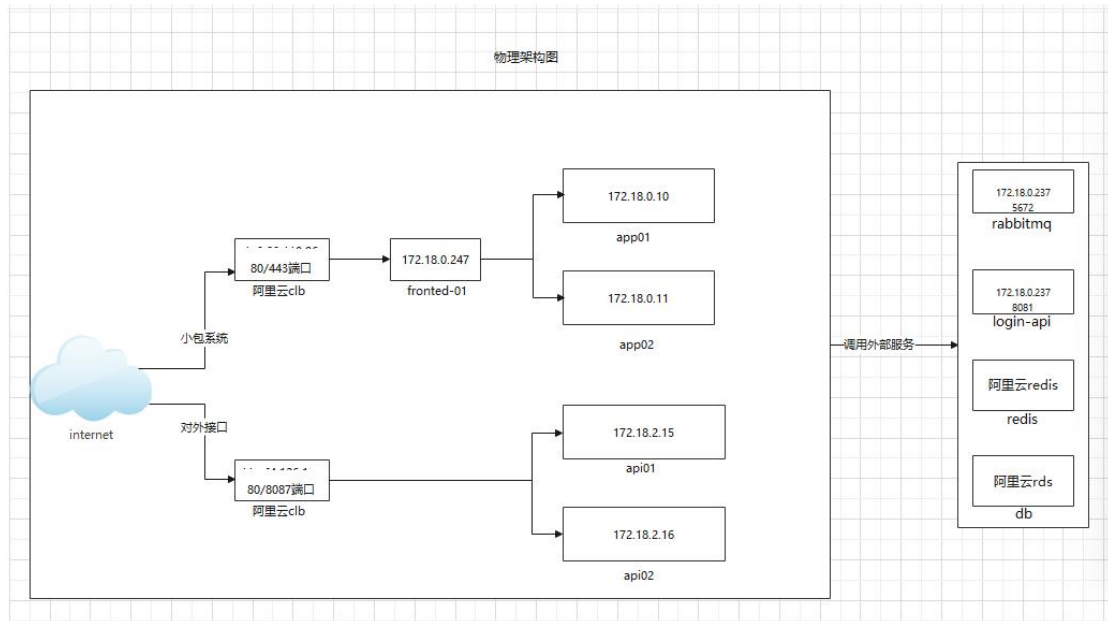
1. ci/cd 原理

k8s 的 ingress,deploy ,svc,pods, 都是按 gitea 各项目名生成
jenkins 的 job 都是按 gitea 各项目名生成

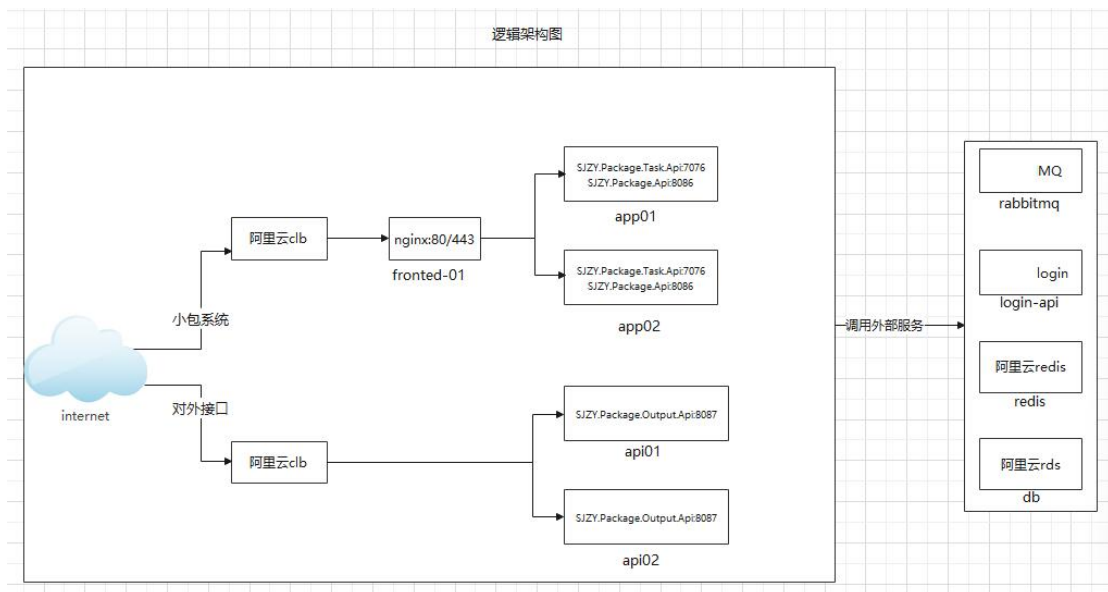
- 1、gitea 各项目名, 代码提交自动触发 webhook.
- 2、jenkins 收到 webhook, 调用相应项目名对应 jenkins job 任务
- 3、jenkins pipeline 任务 pull gitea 仓库相应项目名代码, pull gitea 仓库 config 库的配置文件
- 4、jenkins pipeline 任务执行相应构建(如:编译前后端代码、上传镜像到 harbor)
- 5、jenkins pipeline 任务执行 kubectl 发布相应项目

现在小包系统生产环境架构

1. 物理架构图



2. 逻辑架构图



3. 架构问题

- 1、 nginx 存在单点故障。
- 2、 有问题时回滚比较麻烦，需修改或回滚代码，再手动编译、上传、重启服务。

现有小包系统发布情况

1、 常见的几种发布分析

蓝绿发布

比如 SJZY.Package.Api 业务，我们可以启动两个服务 A 和 B，正常是每组各 50%流量，**升级时前端转发(如 clb\nginx 等)切换流量全部 100%转到 A**，再升级 B。然后再切至 B，升级 A，最后全部放开到 A 和 B。

滚动发布

比如 SJZY.Package.Api 业务，我们（通过 k8s 等）**两个或以上**服务，升级时自动启动一个新的，关闭一个旧的，直到所有都 OK。

2、 目前发布方式情况及问题分析

目前发布方式：

前端：

手动编译前端文件，并打包上传到服务器，重新 reload nginx

后端：

手动编译文件，打包上传到服务器，停止两台服务器中一台服务（**无修改前端转发**），更新以后开启。

问题分析：

前端为单点 nginx

后端手动发布时，没有做到蓝绿发布，因为停止其中一个服务时，还会有请求过来。

3、 解决方案

方案一：启动用多点 nginx, 后端发布时采用蓝绿发布。

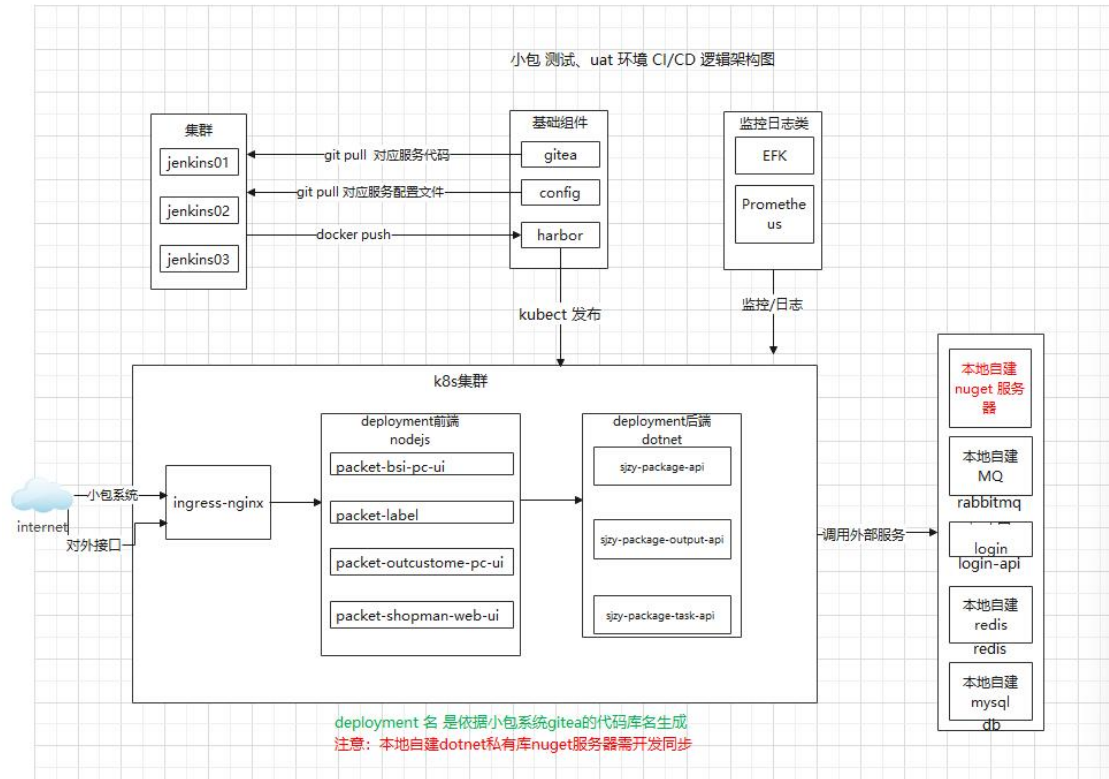
方案二： 采用其它 CI/CD 高可用架构，如 keepalived+lvs/nginx \ k8s 等

CI/CD 架构规划

1、测试、UAT 环境架构规划

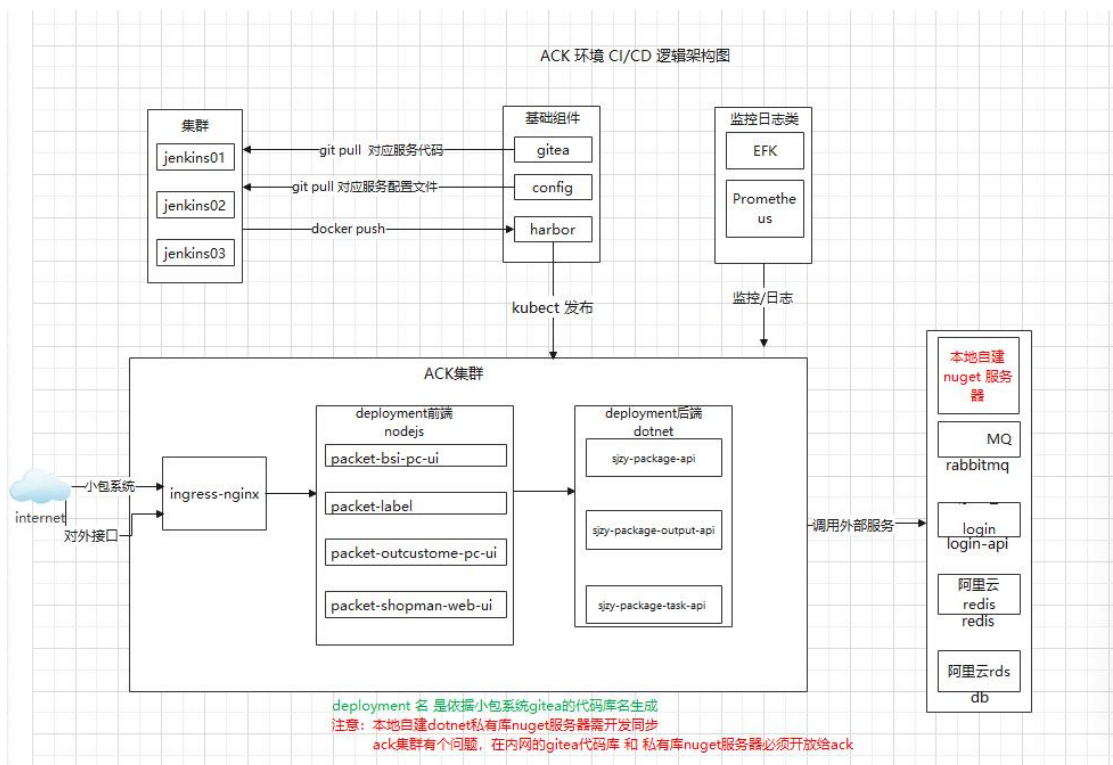
本地机房部署规划如下图

注意红色字



2、生产环境 ACK 架构规划

注意红色字



虚拟机实现部署 CI/CD 架构支持功能说明

1、测试和 UAT 环境架构虚拟机全部实现

- 支持存储编排
- 基于镜像滚动升级
- 一键回滚
- 故障自我修复
- 服务发现和负载均衡
- 支持配置管理
- 支持多种发布

k8s 的 ingress, deploy, svc, pods, 都是按 gitea 名生成

```
[root@T01 ~]# kubectl get nodes -A
NAME        STATUS    ROLES    AGE   VERSION
t01         Ready     control-plane  40d   v1.24.2
t02         Ready     control-plane  40d   v1.24.2
t03         Ready     control-plane  40d   v1.24.2
t04         Ready     <none>       40d   v1.24.2
t05         Ready     <none>       40d   v1.24.2
t06         Ready     <none>       40d   v1.24.2
[root@T01 ~]#
[root@T01 ~]#
[root@T01 ~]# kubectl get deploy,svc,pod,ingress -o wide -n sjzy
NAME                                READY    UP-TO-DATE    AVAILABLE    AGE    CONTAINERS    IMAGES
deployment.apps/packet-bsi-pc-ui    1/1      1              1            3h26m  packet-bsi-pc-ui  harbor:8080/front/packet_bsi_pc_ui
deployment.apps/packet-label        1/1      1              1            3h26m  packet-label      harbor:8080/front/packet_label
deployment.apps/packet-outcustome-pc-ui 1/1      1              1            3h26m  packet-outcustome-pc-ui  harbor:8080/front/packet_outcustome_pc_u
deployment.apps/packet-shopman-web-ui 1/1      1              1            3h26m  packet-shopman-web-ui  harbor:8080/front/packet_shopman_web_ui
deployment.apps/sjzy-package-api     1/1      1              1            3h27m  sjzy-package-api  harbor:8080/back/sjzy.package.api
deployment.apps/sjzy-package-task-api 1/1      1              1            3h27m  sjzy-package-task-api  harbor:8080/back/sjzy.package.task.api

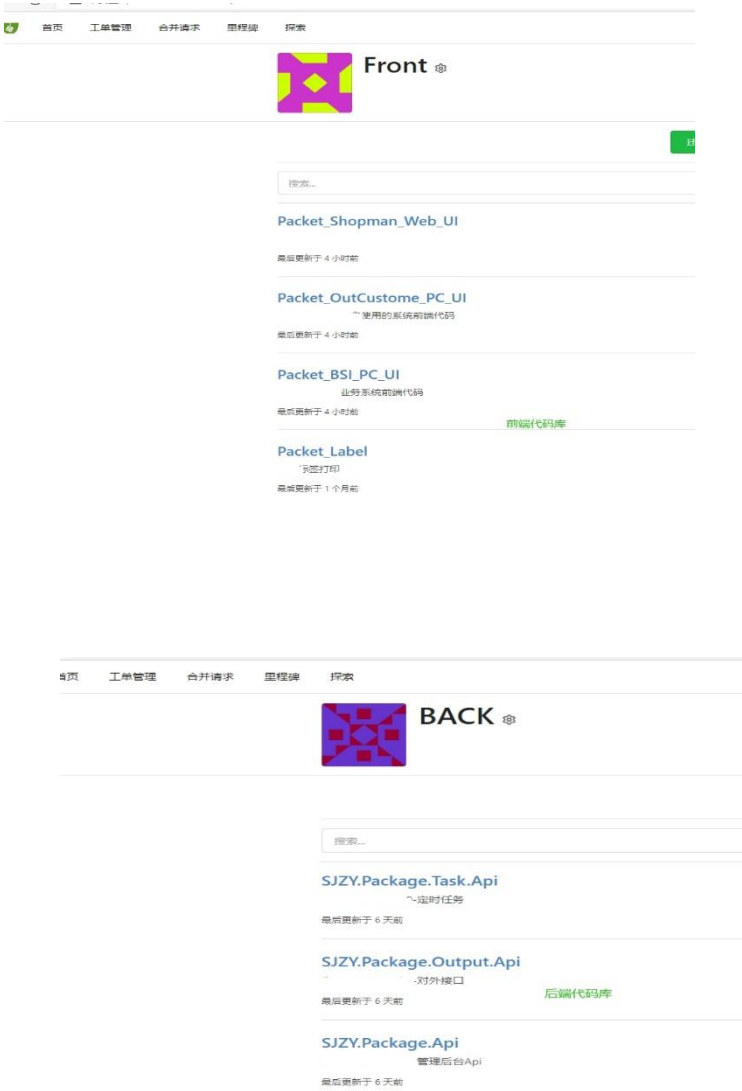
NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE    SELECTOR
service/packet-bsi-pc-ui            ClusterIP     10.99.198.90   <none>         80/TCP     3h26m  app=packet-bsi-pc-ui
service/packet-label                 ClusterIP     10.101.106.64  <none>         80/TCP     3h26m  app=packet-label
service/packet-outcustome-pc-ui      ClusterIP     10.101.187.152 <none>         80/TCP     3h26m  app=packet-outcustome-pc-ui
service/packet-shopman-web-ui       ClusterIP     10.102.247.105 <none>         80/TCP     3h26m  app=packet-shopman-web-ui
service/sjzy-package-api             ClusterIP     10.100.174.146 <none>         8086/TCP   3h27m  app=sjzy-package-api
service/sjzy-package-output-api     ClusterIP     10.99.61.138   <none>         8087/TCP   3h27m  app=sjzy-package-output-api
service/sjzy-package-task-api       ClusterIP     10.103.78.228  <none>         7076/TCP   3h27m  app=sjzy-package-task-api

NAME                                READY    STATUS    RESTARTS    AGE    IP             NODE    NOMINATED NODE    READINESS GATES
pod/packet-bsi-pc-ui-8694675f86-wf6wq 1/1      Running   0            3h26m  10.124.159.187 t05     <none>             <none>
pod/packet-label-7455bc866f-9cxt6      1/1      Running   0            3h26m  10.112.233.32  t04     <none>             <none>
pod/packet-outcustome-pc-ui-564dcc7d68-85xrx 1/1      Running   0            3h26m  10.120.88.154  t06     <none>             <none>
pod/packet-shopman-web-ui-777bb69c9-rxzpl 1/1      Running   0            3h26m  10.112.233.31  t04     <none>             <none>
pod/sjzy-package-api-5db979584d-9kcp8    1/1      Running   0            3h27m  10.112.233.29  t04     <none>             <none>
pod/sjzy-package-output-api-5b6594b897-nv62m 1/1      Running   0            3h27m  10.120.88.151  t06     <none>             <none>
pod/sjzy-package-task-api-6dd796bb8d-gntzl 1/1      Running   0            3h27m  10.124.159.190 t05     <none>             <none>


NAME                                CLASS    HOSTS                                ADDRESS
ingress.networking.k8s.io/sjzy-front-ingress nginx    tms.192.168.56.104.com,gt192.168.56.104.com,u192.168.56.104.com + 3 more... 192.168.56.104,192.168.56.104,192.168.56.104
```

所有基础服务截图说明


1、Gitea 代码库截图





2、jenkins 截图（job 名按 gitea 代码名字生成）


Jenkins


Dashboard >

 用户列表

 构建历史

 项目关系

 检查文件指纹

 凭据

构建队列

队列中没有构建任务

构建执行状态

master

1 空闲








jenkins02

1 空闲

jenkins03

1 空闲

所有

S	W	名称 ↓
✓		Packet_BSI_PC_UI
✓		Packet_Label
✓		Packet_OutCustome_PC_UI
✓		Packet_Shopman_Web_UI
✓		SJZY.Package.Api
✓		SJZY.Package.Output.Api
✓		SJZY.Package.Task.Api

根据前端和后端代码库搞的jenkins

图标: 小 中 大

3、harbor 截图

受欢迎的镜像仓库

名称
back/sjzy.package.api
back/sjzy.package.output.api
back/sjzy.package.task.api
front/packet_outcustome_pc_ui
front/packet_label
front/packet_bsi_pc_ui
front/packet_shopman_web_ui
front/packet

根据前后端代码生成的image仓库名

back/sjzy.package.api

标签	命令
: 2022-07-15-17-01	docker pull harbor/back/sjzy.package.api:2022-07-15-17-01
: 2022-08-01-19-41	docker pull harbor/back/sjzy.package.api:2022-08-01-19-41
: 2022-08-02-10-15	docker pull harbor/back/sjzy.package.api:2022-08-02-10-15
: 2022-08-02-16-59	docker pull harbor/back/sjzy.package.api:2022-08-02-16-59
: 2022-08-03-18-54	docker pull harbor/back/sjzy.package.api:2022-08-03-18-54
: latest	docker pull harbor/back/sjzy.package.api:latest

进入其中一个镜像库可以看到不同时间生成镜像标签

```
[yunwei@192_168_80_73 ~]$ docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
harbor:8080/front/packet_bsi_pc_ui    2022-08-08-14-40   6696a58fc3c6       5 hours ago        406MB
harbor:8080/front/packet_bsi_pc_ui    latest             6696a58fc3c6       5 hours ago        406MB
harbor:8080/front/packet_label        2022-08-08-14-25   0da4d1f48afb       5 hours ago        391MB
harbor:8080/front/packet_label        latest             0da4d1f48afb       5 hours ago        391MB
harbor:8080/front/packet_shopman_web_ui 2022-08-08-14-25   e7d6b10c71ac       5 hours ago        388MB
harbor:8080/front/packet_shopman_web_ui latest             e7d6b10c71ac       5 hours ago        388MB
harbor:8080/front/packet_outcustome_pc_ui 2022-08-08-14-20   bd63326a2ed3       5 hours ago        399MB
harbor:8080/front/packet_outcustome_pc_ui latest             bd63326a2ed3       5 hours ago        399MB
harbor:8080/front/packet_outcustome_pc_ui 2022-08-08-09-07   a2b617d4c35b       10 hours ago       378MB
```

命令行 类似于这样

4、config 配置中心截图

config / config

关注

<> 代码

🕒 工单

🔗 合并请求

📁 项目

📅 版本发布

📖 百科

🔔 动态

小包系统 后端配置文件 前端配置文件

🕒 17 提交

🔗 1 分支

📌 0 Git 标签

🔗 分支: master

比较

转到文件

HTTP http://19

📁 lijing d55d38de92 delete Packet_BSL_PC_UI.proxy.conf.bak

back

modified appsettings.json.SJZY.Package.Api

front

delete Packet_BSL_PC_UI.proxy.conf.bak

README.md

add README.md

📖 README.md

小包系统 config file

注意：都是按gitea代码仓库的项目名，创建的配置文件。

目录back保存为后端各个任务配置文件。

目录front保存为前端各个任务配置文件。

配置文件包括：

1、创建pod需要的镜像dockerfile

2、生成新镜像时，需要的程序配置文件

config / config

关注

<> 代码

🕒 工单

🔗 合并请求

📁 项目

📅 版本发布

📖 百科

🔔 动态

小包系统 后端配置文件 前端配置文件

🕒 17 提交

🔗 1 分支

📌 0 Git 标签

🔗 分支: master

config / front

HTTP http://19

📁 lijing d55d38de92 delete Packet_BSL_PC_UI.proxy.conf.bak

..

packet

add README.md

Packet_BSL_PC_UI.proxy.conf

add README.md

Packet_Label.proxy.conf

add README.md

Packet_OutCustome_PC_UI.proxy.conf

add README.md

Packet_Shopman_Web_UI.proxy.conf

add README.md

README.md

add README.md

dockerfile

dockerfile

📖 README.md

*****小包系统 *****

前端配置

注意：镜像“harbor:8080/front/packet”，是基础nginx镜像，通过packet文件的dockerfile创建。

config / config

<> 代码

🕒 工单

🔗 合并请求

📁 项目

📦 版本发布

📖 百科

🔄 动态

小包系统 后端配置文件 前端配置文件

🕒 17 提交

🌿 1 分支

🏷 0 Git标签

🌿 分支: master

config / back

👤 lijing

71b9b4ef2e

modified appsettings.json.SJZY.Package.Api

⬅️

📄 Dockerfile.SJZY.Package.Api

add front config

📄 Dockerfile.SJZY.Package.Output.Api

add front config

📄 Dockerfile.SJZY.Package.Task.Api

add front config

📄 README.md

add front config

📄 appsettings.json.SJZY.Package.Api

modified appsettings.json.SJZY.Package.Api

📄 appsettings.json.SJZY.Package.Output.Api

add front config

📄 appsettings.json.SJZY.Package.Task.Api

add front config

📖 README.md

*****小包系统 *****
后端配置

演示