**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. habor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

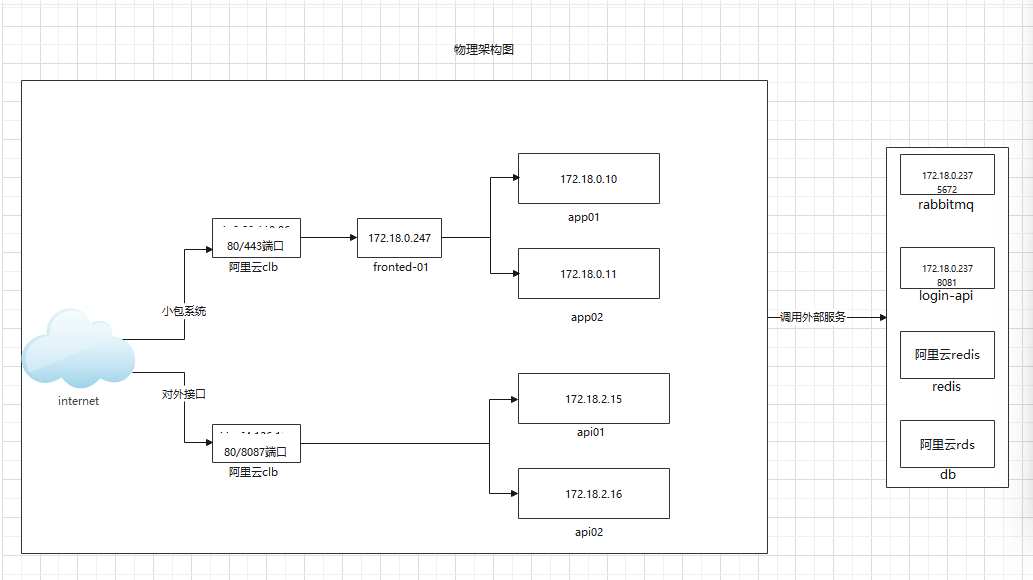
3、jenkins pipeline任务pull gitea仓库相应项目名代码,pull gitea仓库config库的配置文件

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）

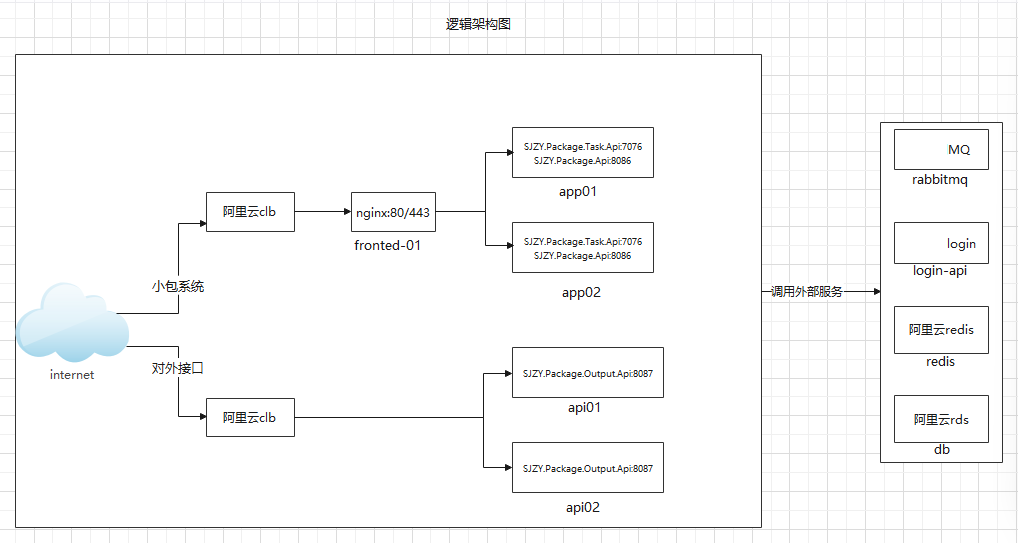
5、jenkins pipeline 任务执行kubectl发布相应项目

**现在小包系统生产环境架构**

1. 物理架构图



1. 逻辑架构图



1. 架构问题
2. nginx存在单点故障。
3. 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。

1. 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。

1. 解决方案

方案一：启动用多点nginx,后端发布时采用蓝绿发布。

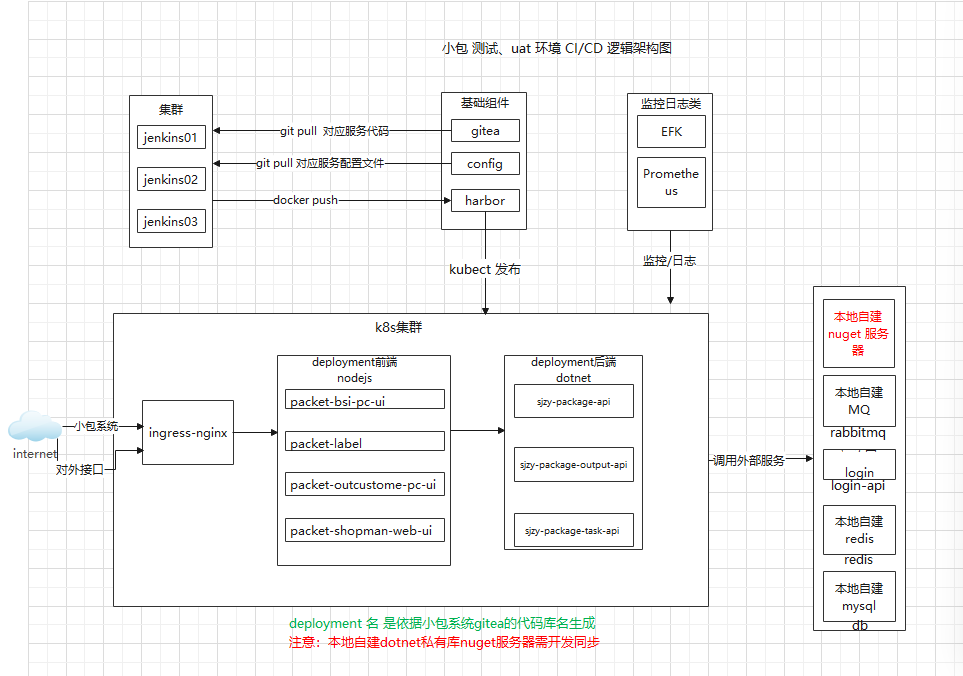
方案二： 采用其它CI/CD高可用架构，如keepalived+lvs/nginx \ k8s等

**CI/CD架构规划**

1. 测试、UAT环境架构规划

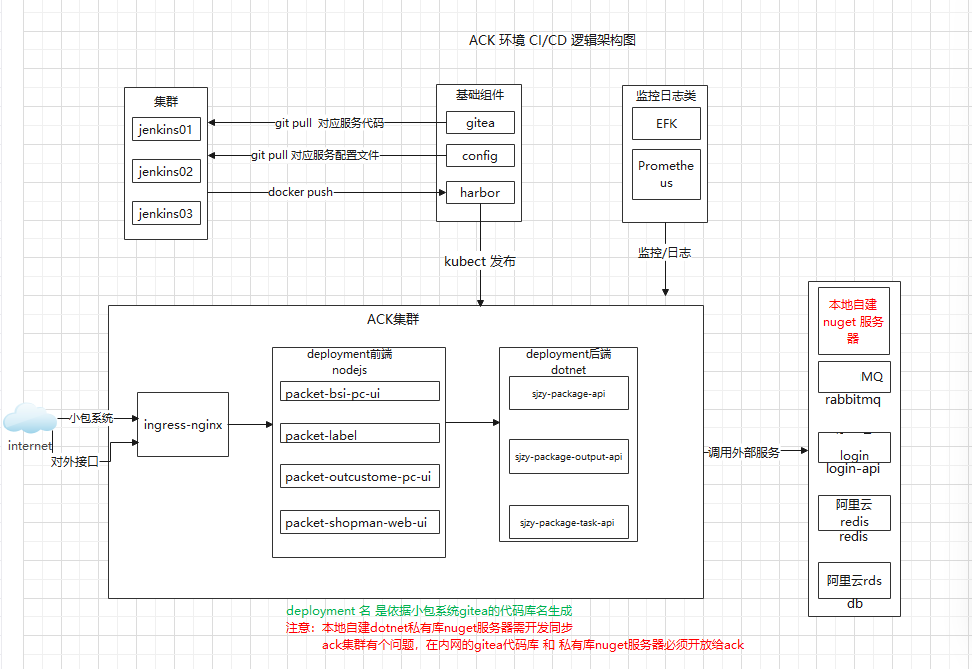
本地机房部署规划如下图

注意红色字



1. 生产环境ACK架构规划

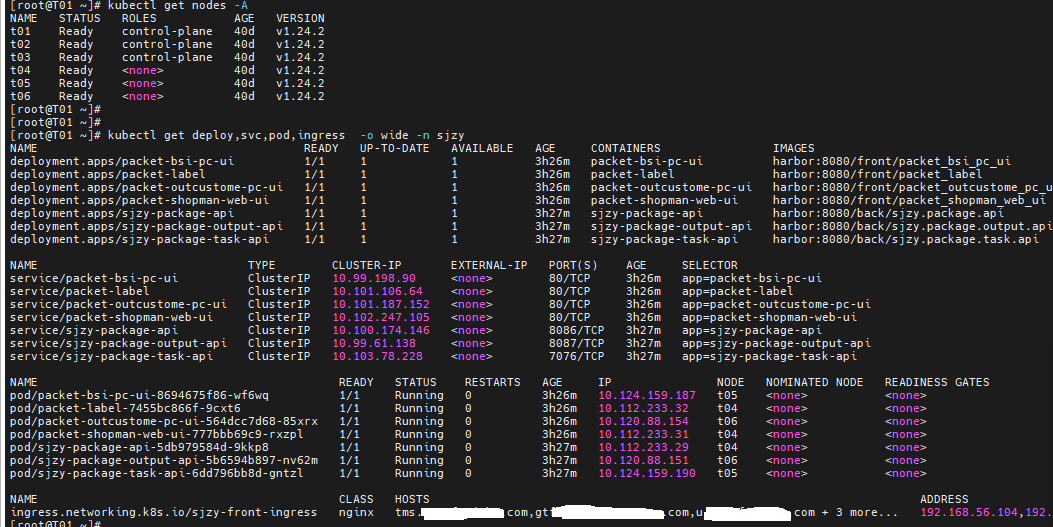
注意红色字



**虚拟机实现部署CI/CD架构支持功能说明**

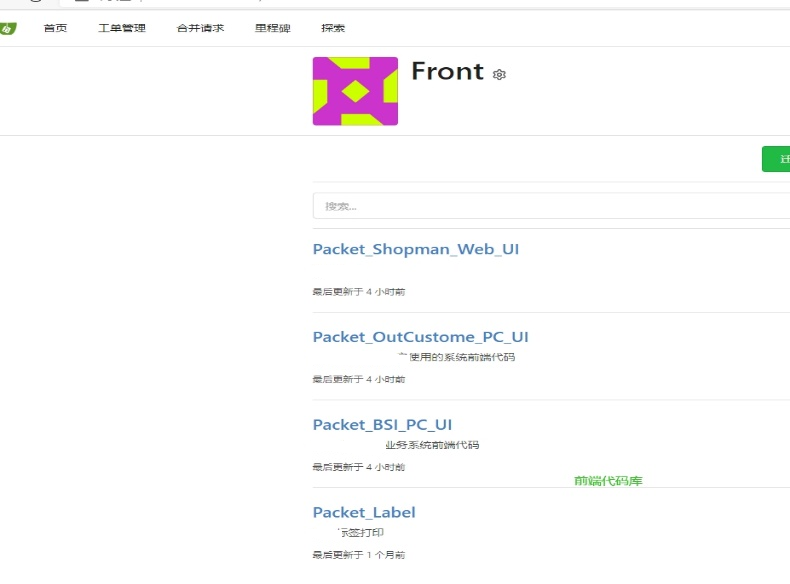
1. 测试和UAT环境架构虚拟机全部实现
2. 支持存储编排
3. 基于镜像滚动升级
4. 一键回滚
5. 故障自我修复
6. 服务发现和负载均衡
7. 支持配置管理
8. 支持多种发布

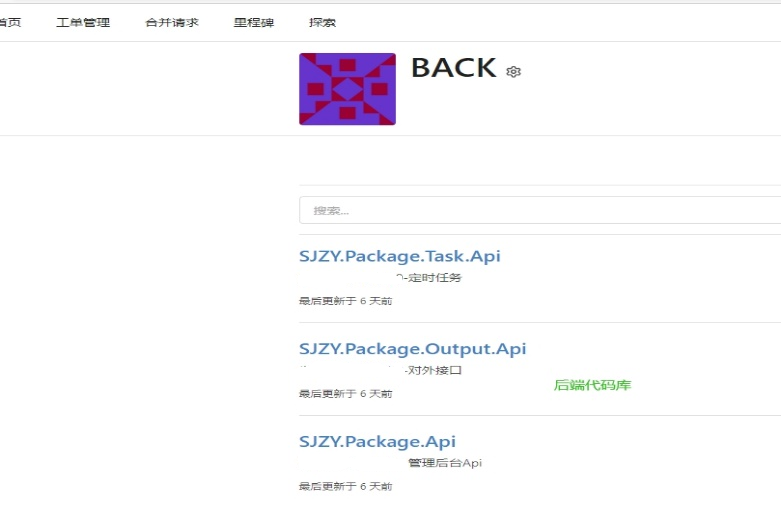
k8s 的ingress,deploy ,svc,pods, 都是按 gitea 名生成

****

**所有基础服务截图说明**

1. Gitea代码库截图

****

****

1. jenkins截图 （job名按 gitea代码名字生成）

**图形用户界面, 应用程序

描述已自动生成**

1. harbor截图

**图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成** **图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成**文本

描述已自动生成

1. config配置中心截图

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

**演示**