

搭建部署 wiki

作者: [yuanhenglizhen](#)

原文链接: <https://ld246.com/article/1595206037439>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

搭建部署wiki

环境准备

操作系统: Centos 7

软件: docker / gitlab / jenkins (blueocean版本)

安装gitlab

```
$ docker run -d -p 443:443 -p 80:80 -p 222:22 --name gitlab --restart always -v /home/gitlab  
config:/etc/gitlab -v /home/gitlab/logs:/var/log/gitlab -v /home/gitlab/data:/var/opt/gitlab git  
lab/gitlab-ce  
# -d: 后台运行  
# -p: 将容器内部端口向外映射  
# --name: 命名容器名称  
# -v: 将容器内数据文件夹或者日志、配置等文件夹挂载到宿主机指定目录
```

按上面的方式, gitlab容器运行没问题, 但在gitlab上创建项目的时候, 生成项目的URL访问地址是容器的hostname来生成的, 也就是容器的id。作为gitlab服务器, 我们需要一个固定的URL访问地址于是需要配置gitlab.rb (宿主机路径: /home/gitlab/config/gitlab.rb)。

```
# gitlab.rb文件内容默认全是注释  
$ vim /home/gitlab/config/gitlab.rb  
# 配置http协议所使用的访问地址,不加端口号默认为80  
external_url 'http://x.x.x.x'  
  
# 配置ssh协议所使用的访问地址和端口  
gitlab_rails['gitlab_ssh_host'] = 'http://x.x.x.x'  
gitlab_rails['gitlab_shell_ssh_port'] = 222 # 此端口是run时22端口映射的222端口  
:wq #保存配置文件并退出  
修改gitlab.rb文件  
# 重启gitlab容器  
$ docker restart gitlab  
此时项目的仓库地址就变了。如果ssh端口地址不是默认的22, 就会加上ssh:// 协议头  
打开浏览器输入ip地址(因为我的gitlab端口为80, 所以浏览器url不用输入端口号, 如果端口号不是8, 则打开为: ip:端口号)
```

安装jenkins

```
docker run \  
-u root \  
--rm \  
-d \  
-p 8080:8080 \  
-p 50000:50000 \  
-v jenkins-data:/var/jenkins_home \  
-v /var/run/docker.sock:/var/run/docker.sock \  
jenkinsci/blueocean
```

mkdocs搭建

目录结构如下

```
.  
├── Dockerfile  
├── Jenkinsfile  
├── README.md  
└── docs  
    └── mkdocs.yml
```

Dockerfile内容如下

```
From squidfunk/mkdocs-material  
ADD . /docs
```

Jenkinsfile内容如下

```
import java.text.SimpleDateFormat  
node {  
    def dockerName='wiki'  
    stage('git pull'){  
        sh 'pwd'  
        git credentialsId: '001', url: 'http://1xx.xxx.xx.xx/sunwei/wiki.git'  
    }  
    stage('docker run') {  
        sh 'pwd'  
        def imageUrl = "squidfunk/mkdocs-material:v1"  
        def customImage = docker.build(imageUrl)  
        sh "docker rm -f ${dockerName} || true"  
        docker.image('squidfunk/mkdocs-material:v1').run("-it -d --name ${dockerName} -p 809  
:8000 --workdir /docs")  
    }  
}
```

配置jenkins

新建一个pipeline项目

Declarative Pipeline (Docker) Pipeline Libraries

分支源

Git

项目仓库: http://**192.168.1.100**/sunwei/wiki.git

凭据: BlueOcean 文件夹凭据

行为:

- 仓库相关
- 发现分支

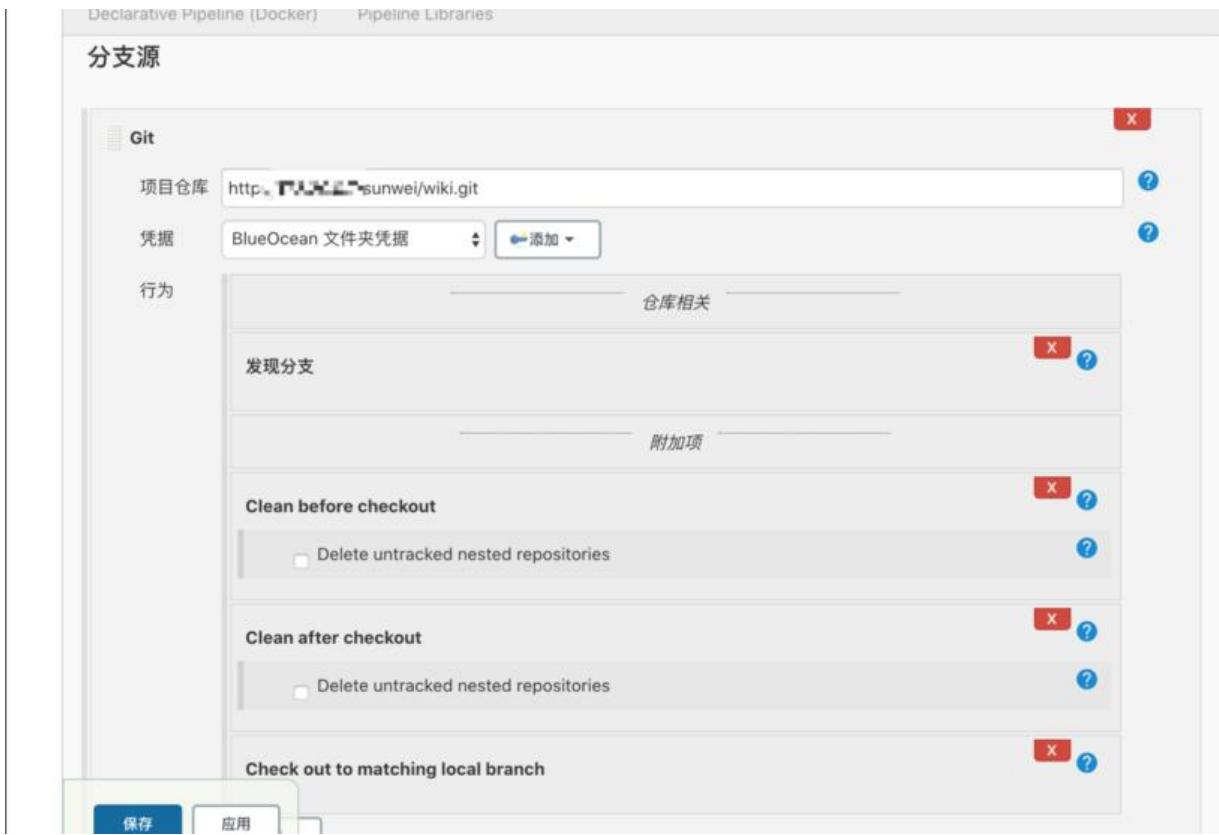
Clean before checkout

- Delete untracked nested repositories

Clean after checkout

- Delete untracked nested repositories

Check out to matching local branch



配置jenkins凭证

jenkins - 配置 - 全局配置

Gitlab

Enable authentication for '/project' end-point

GitLab connections

Connection name: gitlab	A name for the connection
Gitlab host URL: http:// 192.168.1.100	The complete URL to the Gitlab server (e.g. http://gitlab.mydomain.com)
Credentials: GitLab API token <input type="button" value="添加"/>	API Token for accessing Gitlab

Jira



配置gitlab

管理中心 - 项目 - 编辑

W wiki

合 项目
仓库
议题
合并请求
CI / CD
运维
Wiki
代码片段
设置
通用
成员
集成
仓库
CI / CD
运维

集成

Webhooks can be used for binding events when something is happening within the project.

URL
http://example.com/trigger-ci.json

secret Token

Use this token to validate received payloads. It will be sent with the request in the X-Gitlab-HTTP header.

Trigger

Push events
This URL will be triggered by a push to the repository
Branch name or wildcard pattern to trigger on (leave blank for all)

Tag push events
This URL will be triggered when a new tag is pushed to the repository

Comments
This URL will be triggered when someone adds a comment

Confidential Comments
This URL will be triggered when someone adds a comment on a confidential issue

Issues events
This URL will be triggered when an issue is created/updated/merged

Confidential Issues events
This URL will be triggered when a confidential issue is created/updated/merged

Merge request events
This URL will be triggered when a merge request is created/updated/merged

这样每次提交代码时，就会触发自动构建

可以实现wiki, markdown编辑提交自动部署了