Docker方式部署的Gitlab迁移全过程



leo_zdz ● 于 2021-03-05 21:29:18 发布 ● 阅读量2.5k ☆ 收藏 11 ▲ 点赞数 5

分类专栏: 运维 文章标签: docker 运维 gitlab linux



运维 专栏收录该内容

0 订阅 3 篇文章

任务

接到任务,公司的gitlab 服务器要做迁移。gitlab服务器原先部署在一台虚拟服务器上(服务器A),采用Docker 方式部署。由于宿主机磁盘 了,几次出现磁盘用满的情况,每次都是删除gitlab日志文件才能坚持一段时间。这次领导决定gitlab服务器迁移到另一台服务器上(服务器B)。

思路

之前的gitlab容器的镜像是经过修改的,负责之前gitlab服务部署的人已经离职很久了,改了哪里没人知道。为了避免麻烦,还是基于修改过的镜(的gitlab容器比较稳妥。

所以我的思路是:

- 1. 基于现有镜像迁移。从之前的服务器A上把修改过的gitlab镜像导出、拷贝、导入到目标服务器B的docker中;
- 2. 拷贝服务器A现有gitlab服务的配置文件(目录)和数据文件(目录)到目标服务器B。
- 3. 基于原镜像和配置及数据文件启动gitlab容器。

这个方案会有坑,一般是文件和目录权限问题,但好在已经有前人替我们踩过了。说干就干,开始验证。

迁移gitlab镜像

简单讲就是导出现有镜像 --> 复制到目标服务器上 --> 在目标服务器的docker中导入镜像。

这里提一下, Docker导出/导入镜像的方式有两种:

- · export/import
- save/load

两种方式的差别是:

- 1. save/load 操作的是镜像(image), docker save 用来将一个或多个镜像保存到文件,然后拷贝到目标服务器上使用 docker load 载入; export 的是容器(container)。docker export 主要用于基于容器来制作镜像,比如基于一个基础镜像启动一个容器,然后安装一些软件、进行一些设 用 docker export 保存为一个镜像。然后,把这个镜像分发出去;
- 2. docker import 可以为镜像指定新名称, docker load 不能对载入的镜像重命名;
- 3. docker export 不支持将多个镜像打包到一个文件中, docker save 支持将多个镜像打包到一个文件中;

这里采用 save/load 方式做镜像的迁移。

• 在服务器A的docker上导出gitlab镜像

docker save -o gitlab-xxx.tar gitlab-xxx:latest

• 拷贝到目标服务器B后,在docker中加载镜像

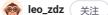
docker load -i gitlab-xxx.tar

在目标服务器B查看镜像列表,确认镜像成功导入。

迁移qitlab数据

服务器A的gitlab容器将配置文件目录、数据文件E

/srv/gitlab/config







- /srv/gitlab/data
- /srv/gitlab/logs

为避免麻烦,直接在服务器B上建立同样的目录,将/srv/gitlab/config 目录和/srv/gitlab/data目录拷贝过来。/srv/gitlab/logs可以不用拷贝。这样 现场基本和服务器A的现场就一致了。

编写docker run的脚本

日常在使用docker时,个人习惯是写一个docker run的脚本文件。这样方便保存容器创建时的一些配置参数,虽然这些参数在docker inspect中大 但还是觉得脚本来得更直观。当然docker-compose .yml可能更符合大多数人的习惯吧,我还是习惯写个bash脚本。

```
1 docker run --detach \
2
    --hostname a8dc5c31ddf7 \
3
    --publish 8443:443 --publish 8080:80 --publish 8022:22 \
4
     --name gitlab \
    --restart always \
5
    --volume $GITLAB_HOME/config:/etc/gitlab \
6
    --volume $GITLAB HOME/logs:/var/log/gitlab \
7
     --volume $GITLAB HOME/data:/var/opt/gitlab \
8
    gitlab-xxx:latest
```

说明:

--hostname a8dc5c31ddf7

容器内的hostname。但为了避免麻烦还是设置成服务器A上的gitlab容器的hostname

--publish 8443:443 --publish 8080:80 --publish 8022:22

暴露的端口,和之前保持一致

--volume \$GITLAB HOME/config:/etc/gitlab

gitlab的配置目录

--volume \$GITLAB_HOME/logs:/var/log/gitlab

gitlab的日志目录

--volume \$GITLAB_HOME/data:/var/opt/gitlab

gitlab的数据目录

\$GITLAB HOME:为了方便,在docker的宿主机上配置环境变量export \$GITLAB HOME=/srv/gitlab

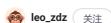
遇到的第一个坑

按以上脚本启动容器。查看容器日志 docker logs -f gitlab, 会发现gitlab容器反复重启。这是遇到的第一个坑。查看容器日志,如下

```
1 \mid \text{If this container fails to start due to permission problems try to fix it by executing:}
2
    docker exec -it gitlab update-permissions
3
     docker restart gitlab
6 Installing gitlab.rb config...
7 cp: cannot create regular file '/etc/gitlab/gitlab.rb': Permission denied
```

从日志上看是权限问题。可以按照日志提示信息做如下两个操作。注意,在容器重启过程中也是可以执行的,由于容器可能正在重启的过程中, 行可能不会成功,可以执行多次。或者docker stop gitlab在docker start gitlab,立即执行下面这两个操作。

```
1 | docker exec -it gitlab update-permissions
2
3 docker restart gitlab
```



docker stop gitlab , 然后 docker start gitlab , 立即进入容器 docker exec -it gitlab bash , 在容器中 , 执行gitlab-ctl reconfigure 。实践证明 , 这 效的。执行gitlab-ctl reconfigure成功后,可正常访问gitlab了,可以看到之前的项目和数据都完好无损。

等等,以为这就可以了?再查看一下gitlab容器日志。

遇到的第二个坑

现在虽然qitlab可以正常访问了,但是容器日志中频繁输出如下错误:

caller=main.go:226 err="open /var/opt/gitlab/alertmanager/data/nflog: permission denied"

查看gitlab各服务状态: docker exec gitlab gitlab-ctl status

```
1 [root@localhost data]# docker exec gitlab gitlab-ctl status
2 down: alertmanager: (pid 8502) 366s; run: log: (pid 489) 1206s
3 | run: gitaly: (pid 473) 1206s; run: log: (pid 472) 1206s
4 | run: gitlab-monitor: (pid 478) 1206s; run: log: (pid 477) 1206s
```

alertmanager是down状态。所幸,其余服务都是run的状态。参考这里 https://gitlab.com/gitlab-org/omnibus-gitlab/-/merge_requests/2691

alertmanager由于文件路径权限问题导致无法启动,这是gitlab早期版本的一个bug,我用的gitlab刚好就是那个早期版本。在容器内处理一下吧。 录权限,在容器内修改权限。

进入容器: docker exec -it gitlab bash

在容器内看目录 /var/opt/gitlab/alertmanager/data 权限

```
1 root@a8dc5c31ddf7:/var/opt/gitlab/alertmanager# ll
                                    root 42 Mar 5 03:45 ./
3 drwxr-x---. 3 gitlab-prometheus
4 drwxr-xr-x. 20 root
                                    root 4096 Mar 5 03:45 ../
  -rw-r--r--. 1 gitlab-prometheus
                                          132 Mar 5 03:29 alertmanager.yml
                                    root
6 drwx----. 2 root
                                    root 98 Mar 5 08:30 data/
```

可以看到, /var/opt/gitlab/alertmanager/data/目录的owner是root

修改容器内路径/var/opt/gitlab/alertmanager/data的 owner 为 gitlab-prometheus

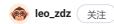
chown -R gitlab-prometheus:gitlab-prometheus data/

修改后

```
1 root@a8dc5c31ddf7:/var/opt/gitlab/alertmanager# ll
2 total 8
3 drwxr-x---. 3 gitlab-prometheus
                                                        42 Mar 5 03:45 ./
                                     root
4 drwxr-xr-x. 20 root
                                                      4096 Mar 5 03:45 ../
                                     root
                                                       132 Mar 5 03:29 alertmanager.yml
  -rw-r--r--. 1 gitlab-prometheus
                                     root
6 drwx-----. 2 gitlab-prometheus
                                     gitlab-prometheus 98 Mar 5 08:30 data/
```

再次查看容器内gitlab各服务状态

```
1 [root@localhost data]# docker exec gitlab gitlab-ctl status
  run: alertmanager: (pid 8502) 366s; run: log: (pid 489) 1206s
  run: gitaly: (pid 473) 1206s; run: log: (pid 472) 1206s
  run: gitlab-monitor: (pid 478) 1206s; run: log: (pid 477) 1206s
5 | ...
```



总结

- 1. 基于docker部署的gitlab服务迁移,直接复制配置和数据目录,使用同版本镜像启动容器的方式是可行的,也并不麻烦。一般会遇到权限问题,ì gitlab-ctl reconfigure 来解决;
- 2. 第二个坑那种bug问题,可以手动改owner来解决权限问题。
- 3. 在没有做配置修改的情况下,执行 "gitlab-ctl reconfigure" 和 "重启容器" (restart或者stop+start) 管用。解决问题就是好的。

以上是我的一次gitlab容器迁移经历。未必完全适合您的情况,在没有备份又不明就里的情况下,不可盲目照搬操作。以上遇到的问题和解决办法

文章知识点与官方知识档案匹配,可进一步学习相关知识

云原生入门技能树 首页 概览 20584 人正在系统学习中

迁移docker部署的GitLab

shijin741231的

GitLab docker迁移工作并不复杂,主要麻烦在GitLab本身的权限上。迁移可通过容器启动日志和容器内交互方式排除故障。

5条评论



Docker部署Gitlab及使用_gitlab openwrt

1.获取gitlab镜像 docker pull gitlab/gitlab-ce 2.启动docker容器 (1)创建文件目录 mkdir -p /srv/gitlab/config mkdir -p /srv/gitlab/logs mk

Docker 部署 gitlab_gitlab docker部署文档

sudo docker pull gitlab/gitlab-ce:latest 3、查看镜像文件 docker images 4、创建Gitlab挂载目录在/root/docker/gitlab下分别创建config,logs,data目录 5、创建docker中的

【非常详细】docker中gitlab的数据迁移

huiker HUI的情

下载镜像 docker pull gitlab/gitlab-ce:latest # 拉取最新镜像 docker pull gitlab/gitlab-ce:11.10.1-ce.0 # 拉取指定版本镜像 查看镜像 docker images 创建容器挂载目录 cd /c

docker化gitlab升级 最新发布

weixin 43620122於

2)gitlab-rake gitlab:backup:restore BACKUP=1628134936_2021_08_05_13.3.6 数据恢复(中间选项一直按y)备份还原:(如果是在其他服务器恢复备份,一定要记得将

使用docker安装gitlab_docker 部署 gitlab,使用域名并替换80端口-CSDN...

-/usr/local/docker/gitlab/data:/var/opt/gitlab -/usr/local/docker/gitlab/logs:/var/log/gitlab 安装完成后 访问地址:http://ip:8080 端口8080 是因为我在配置中设置的外部设

迁移GitLab(基于Docker)_docker的git移植到其他机器

迁移GitLab(基于Docker) 前面介绍了基于Docker 搭建GitLab,本次将详细介绍如何将GitLab从一台服务器迁移到其他服务器。准备首先我们需要保证源服务器和目标服务是

docker部署的gitlab迁移 docker exec -it xxx /bin/bash 进入gitlab容器,输入命令gitlab-rake gitlab:backup:create 生成备份文件。默认路劲在/var/opt/gitlab/backups/

docker中的gitlab数据迁移

- 只为伊人 †

ZYXZH於

使用docker部署gitlab之后,数据整体迁移到其他服务器公司的gitlab原先部署在xxx.xxx.xxx.217服务器的docker上面,目前正常使用,由于服务器内存问题,需将gitlab整

基于docker容器的gitlab迁移与数据恢复

weixin_43880061的

将在docker容器里里面的/var/opt/gitlab/backups/目录下产生一个1665386470_2022_10_10_13.9.2_gitlab_backup.tar的压缩包。101上的gitlab容器是通过docker-compc

docker中gitlab迁移

weixin_43359459的

如果没有挂载数据卷,但是又不知道文件在哪里,可以使用命令在linux本机上查找这个备份的文件。的文件,我需要找到它但是不确定具体位于哪个目录下。我可以执行

Docker折腾记: (3)Docker Compose构建Gitlab,从配置(https,邮箱验证)到基本可用 热门推荐

ccstudyforever的

前言 gitlab 11.1内置了CI/CD,这个特性从gitlab 8+就开始有了,不过配置比较琐碎 经过几个大版本的迭代,现在已经简化了使用方式,也修复了一些坑,这个特性大大吸引了我,\$

从0开始的Gitlab-CICD部署过程

代码更新后自动<mark>部署</mark>:使用Gitlab-CI以前写的代码更新后,都是在本地(Windows)重新编译后通过远程连接复制到服务器,然后再双击运行,以达到更新的目的。后来学长

Gitlab + Docker + Jenkins + Harbor实现自动化部署

qq_41919924的t

4

该文为记录自己在生产环境中,使用Gitlab+Docker+Jenkins+Harbor等技术实现自动化部署的流程与踩过的坑,供大家学习和自己积累。

K8s+Jenkins+GitLab+Harbor+CICD+SonarQube持续写

