IPFS部署操作手册

# 部署环境说明

该部署操作手册适用于ubuntu 20.04 server操作系统，要求有公网连接。是否适用于ubuntu 18.04 server或更高版本需要验证，不确定能否适用于desk版本。

本文只适用于go-ipfs的部署操作。

# 系统与程序环境配置

go-ipfs是基于Go语言的项目，环境要求go语言环境，并在系统变量中进行相应的配置。

## 下载Go语言包(可以打开的网址https://studygolang.com/dl)

（这个网站的下的文件能不能最后编译成功还未知，用的还是以前下载的go1.17的软件包才可以。）

IPFS是基于Go语言的项目，环境要求go version 1.14+。在Go的官方网站下载最新的版本即可[https://golang.org/dl/](https://link.zhihu.com/?target=https://golang.org/dl/" \t "https://zhuanlan.zhihu.com/p/_blank)。现在go的最新版本为go1.17.8， 可以用以下命令下载到服务器本地并解压至指定的目录/usr/local：

**wget https://studygolang.com/dl/golang/go1.18.linux-amd64.tar.gz**

**tar -zxvf go1.18.linux-amd64.tar.gz #解压到当前目录，或者指定/opt目录用 sudo tar -zxvf go1.18.linux-amd64.tar.gz -C /usr/local**

wget https://golang.org/dl/go1.17.8.linux-amd64.tar.gz

tar -C /usr/local -zxvf go1.17.8.linux-amd64.tar.gz

注:

如果wget失败可以到官网[https://golang.org/dl/go1.17.8.linux-amd64.tar.gz](https://link.zhihu.com/?target=https://golang.org/dl/go1.14.6.linux-amd64.tar.gz" \t "https://zhuanlan.zhihu.com/p/_blank)下载镜像，然后在那个目录下打开终端执行上面的命令的第二句。

如果服务器上下载失败，可以在终端机下载完成后，用Xftp连接服务器，将文件拖过去，至于如何连接，请看这篇文章[保姆级教程——Xshell连接虚拟机中的Ubuntu并通过Xftp传输文件](https://link.zhihu.com/?target=https://blog.csdn.net/ExcaliburUlimited/article/details/107718611" \t "https://zhuanlan.zhihu.com/p/_blank)，Xshell和Xftp连接过程相同。

## 部署和配置Go环境

在进入用户主目录，并新建名为go的文件夹，在go的文件夹中建立三个子目录(名字必须为src、pkg和bin)并赋权。创建目录赋权命令如下:

cd ~

sudo mkdir -p go/{src,pkg,bin}

sudo chmod u+x go/{src,pkg,bin}

运行ls -l命令查看目录创建及权限：

|  |
| --- |
|  |

为用户配置go语言环境变量，首先输入：

vi ~/.profile

打开用户的环境变量配置文件，按”i”键进入编译模式，在最后添加如下内容：

export PATH**=**$PATH:/usr/local/go/bin

export GOROOT**=**/usr/local/go

export GOPATH**=**$HOME/go

export PATH**=**$PATH:$HOME/go/bin

再按”Esc”键退出编译，接着输入:wq，回车就可以保存修改并退出。

输入如下命令即激活修改后的环境变量配置文件：

source ~/.profile

最后验证一下是否成功，输入如下命令查看结果：

go version

go env

|  |
| --- |
|  |

出现上图结果即表示go语言环境部署成功。

但是这样关掉终端，配置就会失效，所以需要对~/.bashrc中也进行相应的配置，输入以下命令开始编译：

vi ~**/**.**bashrc**

然后在最后面添加：

export PATH**=**$PATH:/usr/local/go/bin

export GOROOT**=**/usr/local/go

export GOPATH**=**$HOME/go

export PATH**=**$PATH:$HOME/go/bin

再按”Esc”键退出编译，接着输入:wq，回车就可以保存修改并退出。

输入如下命令即激活修改后的环境变量配置文件：

source ~/.bashrc

Go环境部署与配置完成。

## 安装工具包

部署ipfs需要使用gcc与make等工具，运行以下命令更新系统并安装最新的工具：

sudo su #先切换到root权限, 防止有些文件还是提示权限问题。要不然可能出现下图情况。



sudo apt update && apt upgrade -y #操作以下前更新系统apt先。

sudo apt-get install git

sudo apt install gcc

sudo apt install make

系统更新及工具安装完成后，ipfs部署准备完成。

# Ipfs安装部署

## 下载go-ipfs源码

因为go get国内基本上下载不了，所以需要开启并加上镜像进行下载，命令如下：

go env -w GO111MODULE=on

go env -w GOPROXY=https://goproxy.cn,direct

go get -u github.com/ipfs/go-ipfs

下载成功后如下图提示：

|  |
| --- |
|  |

go-ipfs的源码将会下载到~/go/pkg/mod/github.com/ipfs/go-ipfs@v0.12.2/的目录下。

## 编译go-ipfs源码

进入~/go/pkg/mod/github.com/ipfs/go-ipfs@v0.12.2/，运行命令对以下两个文件进行赋权：

chmod 777 bin/check\_go\_version

chmod 777 plugin/loader/preload.sh

然后执行以下命令开始编译go-ipfs：

make install

如果编译正常结束则会出现以下提示：

|  |
| --- |
|  |

编译成功，输入以下命令验证：

ipfs version

如出现以下提示，则表示go-ipfs部署成功：

|  |
| --- |
|  |

注：如果编译时碰到以下问题：

|  |
| --- |
|  |
|  |

则需要对提示Permission denied的文件进行chmod 777赋权，

之后再重新运行make install进行编译即可。

## 恢复配置

恢复配置，保持系统运行安全，命令如下：

go env -w GO111MODULE**=**off

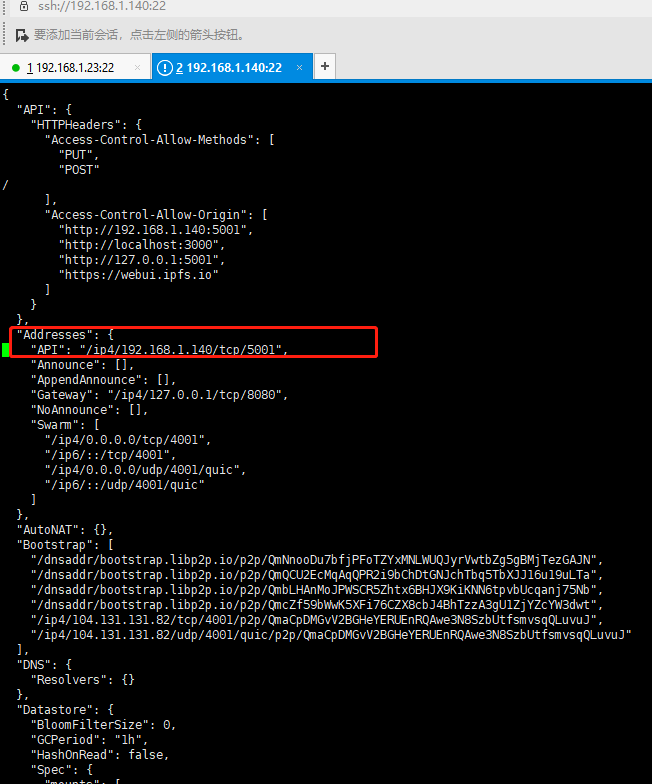
Ipfs部署完成。

**4.运行**

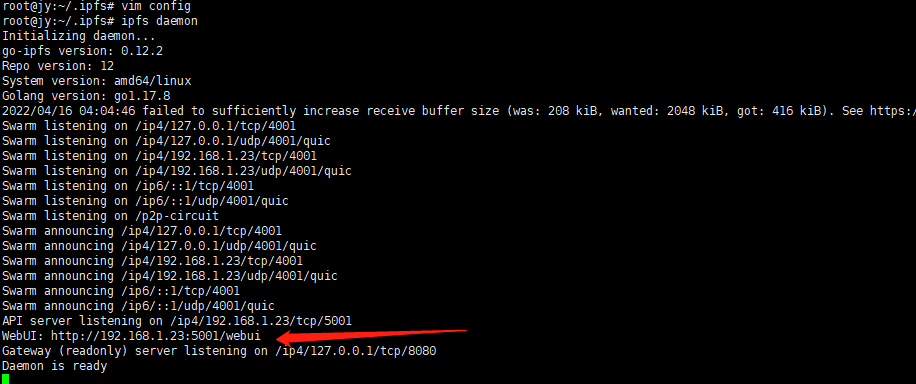
Ipfs init #初始化

cd /root/.ipfs/

vim config #把IP改成本服务器的IP地址。如下图先只改这个为实际IP，,然后;wq保存。



ipfs daemon 然后出现下图，复制地址在浏览器上打开。



浏览器上打开 <http://192.168.1.23:5001/webui>



从网页上的第2点，看到有需要输入的两条命令提示，照着输入。

ipfs config --json API.HTTPHeaders.Access-Control-Allow-Origin '["http://192.168.1.140:5001", "http://localhost:3000", "http://127.0.0.1:5001", "https://webui.ipfs.io"]'

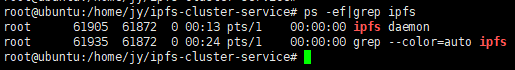
ipfs config --json API.HTTPHeaders.Access-Control-Allow-Methods '["PUT", "POST"]'

最后再次运行 ipfs daemon



**5 . 运行官方的一个数据包服务端（这里是服务端的处理）**

Ctrl + z # 把ipfs daemon进行退到 后台运行。将****任务中断****, 但是此任务并没有结束,他仍然在进程中，有命令ps -ef|grep ipfs查可以看到进程还在。



解压ipfs-cluster-service\_v0.14.5\_linux-amd64 (2).tar.gz 到服务器的某个用户目录。

cd ipfs-cluster-service

./ipfs-cluster-service init #重复运行这命令会把上一次保存的service.json恢复默认配置。

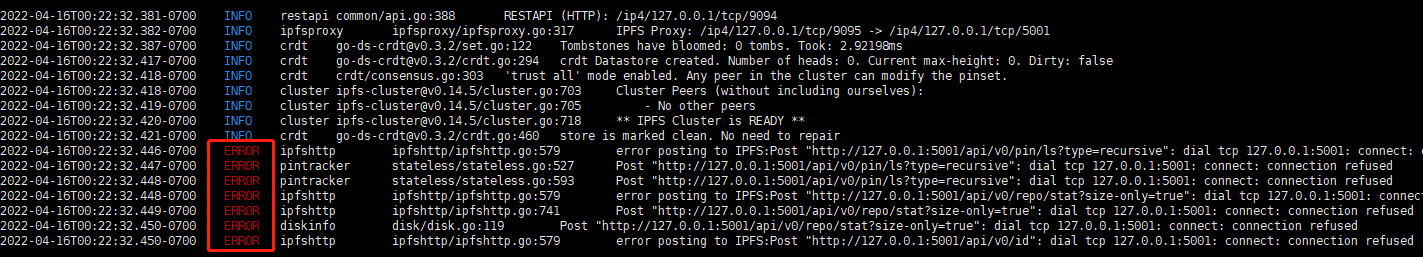
vim /root/.ipfs-cluster/service.json

#修改下图中的这三个位置为真实服务器的IP地址。

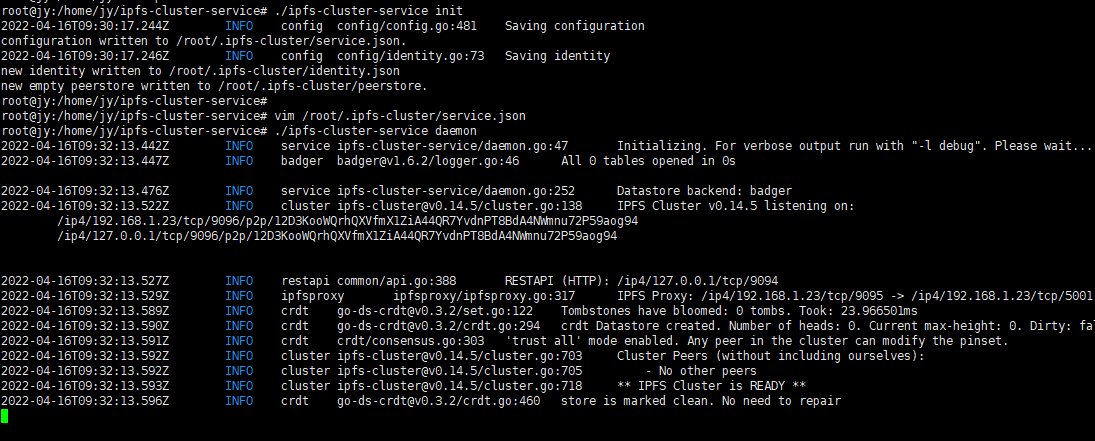


编辑好后保存。

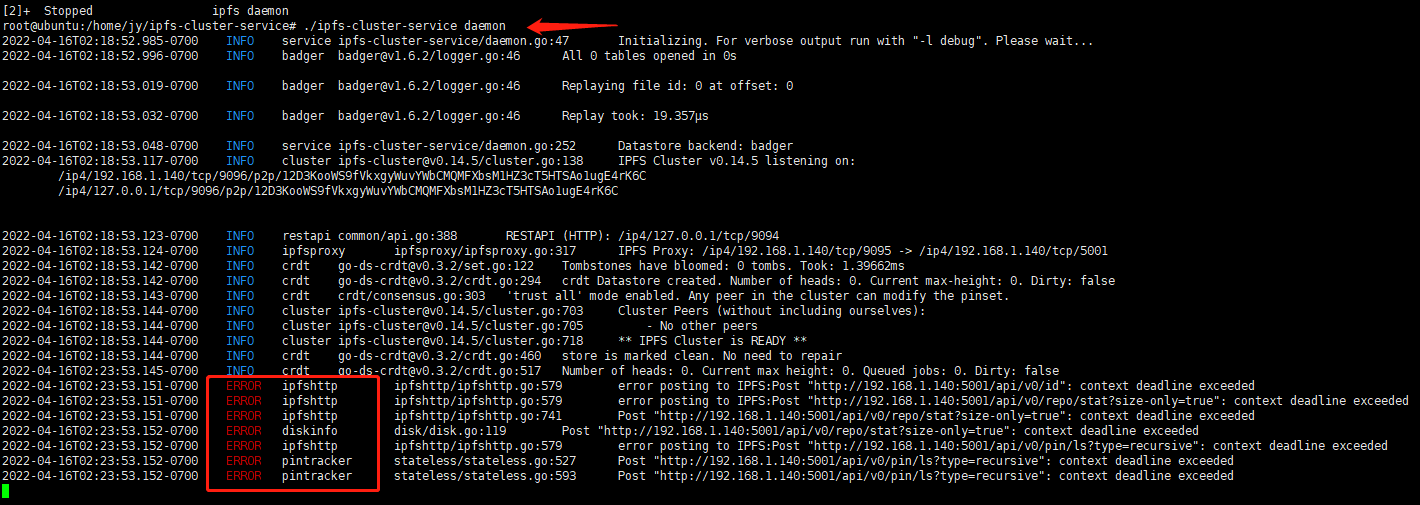
./ipfs-cluster-service daemon #运行服务器端。只有修改了上图三个位置的IP后运行才不会报错。如果没有修改好就会变成下图这样。



成功运行后如下图结果：



运行时间久了后又开始出现一些报错，如下图：



**6. 运行官方的一个数据包客服端（这里是客服端的处理）**

cd ipfs-cluster-ctl