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://golang.org/dl/go1.17.8.linux-amd64.tar.gz> 如果这个网址进入不了就用下面别一个网址下载。注意有些网址下载的go包是最后用命令apt install make解释不了的。

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

**tar -zxvf go1.17.8.linux-amd64.tar.gz #解压到当前目录，或者指定/opt目录用 sudo tar -zxvf go1.17.8.linux-amd64.tar.gz -C /usr/local**

wget https://golang.org/dl/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 #（下面用来看ipfs安装好没有就用ipfs version这个命令）

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

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cd ~/go/pkg/mod/github.com/ipfs/go-ipfs@v0.12.0/，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。

## Ipfs节点参数配置

使用以下命令对ipfs节点进行初始化

|  |
| --- |
| ipfs init #初始化 |
|  |

初始化后显示上图表示初始化成功。

根据上图初始化后生成ipfs配置目录在/root/.ipfs/下，使用命令切换到ipfs目录下：

|  |
| --- |
| cd /root/.ipfs/ #切换目录  ls -l #查看目录文件 |

使用以下命令打开config主配置文件：

|  |
| --- |
| vim /root/.ipfs/config |



打开文件后显示为上图，找到红框中的API和Gateway选项，将IP更换为本机真实IP地址。

为了防止其他外网ipfs节点的接入，使用以下命令删除外网节点：

如果不删除这些外网节点，之后用命令./ipfs-cluster-ctl peers ls查询时就会显示出一些外网IP的地址如下图红柜中的IP就是其他外网的：



|  |
| --- |
| ipfs bootstrap rm --all |

上图显示表示成功删除外网节点。

使用以下命令启动ipfs节点：

|  |
| --- |
| nohup ipfs daemon & （#用nohup &方便在后台运行不掉线。） |
|  |

显示以上图的参数表示ipfs节点启动成功。

浏览器上打开 [http://192.168.1.24:5001/webui](http://192.168.1.23:5001/webui) 可以看到IPFS的网站。



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"]'

输入完上面命令后，关闭ipfs节点可使用以下命令查看进程：

|  |
| --- |
| ps -ef | grep ipfs |

对上图红框中的进程号使用kill -9 13535杀死该进程表示关闭ipfs节点。

最后再次运行 nohup ipfs daemon &

再次浏览器上打开 [http://192.168.1.24:5001/webui](http://192.168.1.23:5001/webui) 变成如下图，网站提示连接到IPFS。



# Ipfs-cluster（IPFS的集群安装）

## 初始化ipfs-cluster-service

使用以下命令对ipfs-cluster-service源码包进行解压：（目的地目录暂时没要求，解压到随便一个目录下）

|  |
| --- |
| tar -zxvf ipfs-cluster-service\_v0.14.5\_linux-amd64.tar.gz -C /opt/ |

使用以下命令进入目录查看文件：

|  |
| --- |
| cd /opt/ipfs-cluster-service/ #切换目录  ls -l #查看文件 |

上图红框中的文件为本次ipfs节点统一管理工具。

使用以下命令对ipfs-cluster-service工具进行初始化：

|  |
| --- |
| cd /opt/ipfs-cluster-service/ #进入目录  ./ipfs-cluster-service init #初始化管理工具 |

显示上图表示管理工具初始化成功。

使用以上命令初始化后，可以发现在root目录下生成了/root/.ipfs-cluster/目录文件。

使用以下使用以下命令进入目录打开主配置文件：

|  |
| --- |
| cd /root/.ipfs-cluster/ #切换目录  ls -l #查看文件 |

## 配置集群参数

使用以下命令打开配置文件：

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

将上图红框中的参数复制更换到其他节点的secret上。

我的secret密钥是b509ba9df0d962509ac6a1c058dbe0daa961fcbebb285b8b8c629549977b664d

注意：每台需要加入ipfs集群中的节点，根据上图红框中的secret参数都要一致。

使用命令打开文件编辑：

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

将上图红框中的三个地方的IP改为本地服务的真实IP地址。

## 启用ipfs管理工具

使用以下命令启动管理工具获取数据：

|  |
| --- |
| cd /opt/ipfs-cluster-service/ #切换目录  ./ipfs-cluster-service daemon #启动ipfs管理工具 |

将红框中的内容复制下来，然后ctrl+c关闭管理节点。

使用以下命令对管理工具重启启动：

|  |
| --- |
| cd /opt/ipfs-cluster-service/ #切换目录  ./ipfs-cluster-service daemon bootstrap /ip4/192.168.1.24/tcp/9096/p2p/12D3KooWJSgxUaxdhFhtAuoAMYCpnPxGphZ7ZrC7PqDBiTX1ikW8 #启动管理工具 |

如上图显示表示启动成功。

注意：运行上图的启动管理工具前，首先需要使用ps -ef | grep ipfs查看ipfs节点是否有启动，没有的话就运行ipfs daemon & 启动ipfs节点。

## 安装ipfs-cluster-ctl

使用以下命令解压ipfs-cluster-ctl源码包：

|  |
| --- |
| tar -zxvf ipfs-cluster-ctl\_v0.14.5\_linux-amd64.tar.gz -C /opt/ |

使用以下命令进入解压目录查看文件：

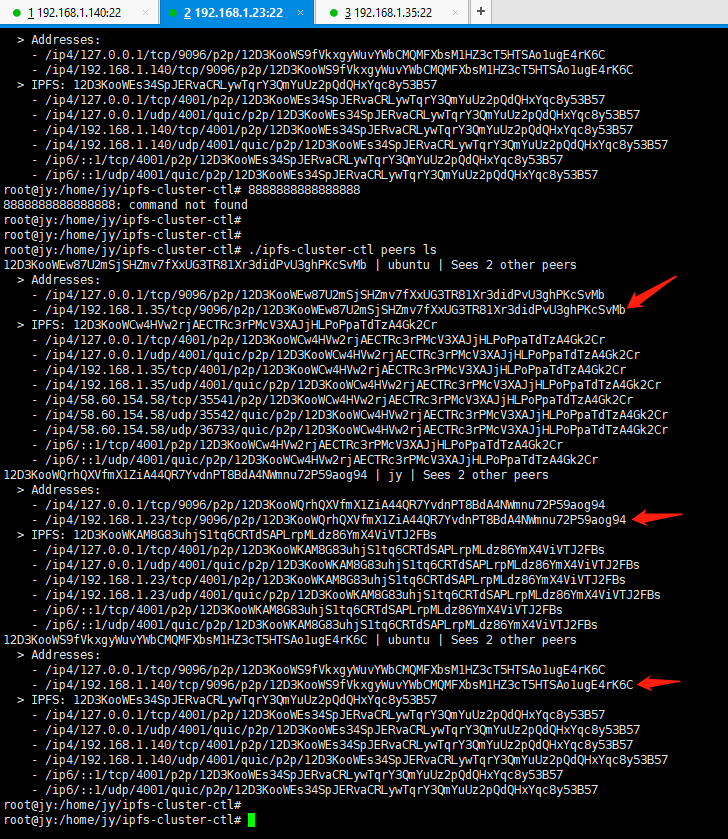
|  |
| --- |
| cd /opt/ipfs-cluster-ctl/  ls -l |

使用以下命令运行上图中的ipfs-cluster-ctl脚本：

|  |
| --- |
| ./ipfs-cluster-ctl id |

显示为上图表示安装成功。

/home/jy/ipfs-cluster-ctl/ipfs-cluster-ctl peers ls #这条命令查看集群的命令，显示出总多少台机器在这个集群。如下图共三台。



## IPFS的上传与下载文件 (安装好集群后，进行某台机器上转文件另一台机器下载文件的试验。)

rz 命令随便上传一个文件到服务器的某个目录。比如到 /home/jy/555.txt

./ipfs-cluster-ctl add 555.txt #文件名,此命令表示上传文件

转着系统自动生成哈希值。如下图，QmNbuSpWHVDuwKX26UkfsSabh5pcb2ME1UK9mgXrSQuJr7就是哈希值。



./ipfs-cluster-ctl pin add 上传文件hash值    #此命令表示数据分发。

然后去到另一台机器使用命令ipfs get 哈希值               #只要有ipfs节点都是可以下载的

cat 555.txt #查看刚才下载的内容。

1. **IPFS群组的存储路径与管理的问题。**

**每一个节点默认的存储路径为本机的/root/.ipfs 这个时候我们想要更大的空间需要进行挂载硬盘的操作，如下：**

**mount /dev/sdb /root/.ipfs #之后这个路径就可以存储这个挂载盘的最大容量。**

**如果这个新挂载的硬盘是全新的，需要进行格式化操作mkfs.xfs /dev/sdb**

**关于存储目录也可以参考以下网：https://blog.csdn.net/wzygis/article/details/114324354?spm=1001.2101.3001.6661.1&utm\_medium=distribute.pc\_relevant\_t0.none-task-blog-2%7Edefault%7ECTRLIST%7EPayColumn-1.pc\_relevant\_default&depth\_1-utm\_source=distribute.pc\_relevant\_t0.none-task-blog-2%7Edefault%7ECTRLIST%7EPayColumn-1.pc\_relevant\_default&utm\_relevant\_index=1**

1. **其中一个节点上传的新文件或数据压缩包，如果只是文件名不一样，IPFS可以识别到里面的东西是一样的，此时每个节点的总存储空间没有变化。但此时这个新文件名的哈希值是不一样。**



1. **修改默认存储路径与存储最大的空间限制。一共要修改两个配置文件。**

**其中一个：改下图的这两个箭头的位置使得改变存储位置。**

**vim /root/.ipfs/config**



另外一个文件要修改 vim /root/.ipfs/datastore\_spec

改下面两个地方。



1. 对之前上传的文件进行删除与空间解放的测试。

先进行 ipfs pin rm 命令从本地存储中解除指定ipfs对象的固定。

ipfs pin rm 哈唏值

ipfs repo gc #命令扫描仓库中的对象，并依照先后顺序移除没有固定的对象， 以便回收磁盘空间。

然后再用du -h --max-depth=1 /root/.ipfs #查IPFS默认的存储空间容量做对比，如果之前没有进行修改存储路径的话。

1. 进行验证文件哈希是否存在。

删除或着上传文件后, 除了用du -h --max-depth=1命令查容量外，还可以配合下面的命令进行验证文件哈希是否存在。

用ipfs pin ls |grep Qmf6r8PYPM158amesk51h5iGQ3jWRLLzjnUZ69en68ZTQc 这个命令查询在不在集群内。

1. 对于ipfs add 文件名，与ipfs pin add 文件的哈希值的使用。这两个命令使用只能把文件上传到单机的存储库上，别的节点机需要下载时也可以通过哈希值下载，但是一旦本机关了或坏了，其他节点机不能下载。

与上面第5节说的命令 ./ipfs-cluster-ctl add 文件名，是不一样的，第5节这个命令是可以即时同步到所有节点机上的存储库上的。（一个只对单机存储容量有影响，一个是对所有节点机存储容量有影响。）

1. 对于大文件的上传比较好的方式用法，对于超过1G或更大文件的上传：

步骤1：ipfs add 文件名 #得到哈希值。

步骤2： 工具包的目录/ipfs-cluster-ctl pin add 哈希值,

这样上传速度会快很多，不会影响到后面的进程速度。要不然一开始用工具包的目录/ipfs-cluster-ctl  add 文件名，0.9G的文件需要用时6-9分钟。

配置启动环境，使可以使用下面的命令：

root@aips:/home/aipsb# systemctl restart ipfs

root@aips:/home/aipsb# systemctl restart ipfs-cluster