****

**Ceph Teuthology(Arm-Centos7)**

Contents

[1. postgresql安装: 4](#_Toc531701013)

[2. paddles配置: 5](#_Toc531701014)

[3. pulpito配置: 6](#_Toc531701015)

[4. supervisor配置: 6](#_Toc531701016)

[5. ntp服务配置: 7](#_Toc531701017)

[6. git服务器配置: 8](#_Toc531701018)

[7. gitlab配置: 10](#_Toc531701019)

[8. jenkins配置: 18](#_Toc531701020)

[9. ceph-build配置: 19](#_Toc531701021)

[10. jenkins-job-builder 19](#_Toc531701022)

[11. teuthology配置: 20](#_Toc531701023)

[12. 附录： 22](#_Toc531701024)

[ redis配置 22](#_Toc531701025)

[ my.cnf 22](#_Toc531701026)

[ nginx.conf 24](#_Toc531701027)

[ 参考： 27](#_Toc531701028)

[ 问题集： 27](#_Toc531701029)

***\* 版本修订记录 \****

|  |  |  |
| --- | --- | --- |
| ***版本号*** | ***修订时间*** | ***修订内容*** |
| *v1.0* | *2018-11-* | *初版修订* |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

***\* Release Copyleft©free \****

1. postgresql安装:

|  |
| --- |
| [root@mon3 ~]# yum -y install python-devel python-virtualenv postgresql postgresql-contrib postgresql-server-dev-all supervisor postgresql-server postgresql-devel  [root@mon3 postgresql\_data]# vim /usr/lib/systemd/system/postgresql.service  …  Environment=PGDATA=/data/postgresql\_data  …  [root@mon3 ~]# su - postgres  -bash-4.2$ vim .bash\_profile  …  PGDATA=/data/postgresql\_data  …  # 关闭selinux  [root@mon3 data]# getenforce  Enforcing  [root@mon3 ~]# setenforce 0 # 临时关闭  [root@mon3 ~]# vim /etc/selinux/config # 永久关闭，重启机器  …  SELINUX=disabled  ….  -bash-4.2$ postgresql-setup initdb  Initializing database ... OK  -bash-4.2$ vim /data/postgresql\_data/pg\_hba.conf  # 添加该行  host all postgres 127.0.0.1/8 md5  -bash-4.2$ exit  logout  [root@mon3 postgresql\_data]# systemctl daemon-reload  [root@mon3 ~]# systemctl enable postgresql  [root@mon3 postgresql\_data]# systemctl start postgresql  [root@mon3 postgresql\_data]# systemctl status postgresql  ● postgresql.service - PostgreSQL database server  Loaded: loaded (/usr/lib/systemd/system/postgresql.service; enabled; vendor preset: disabled)  Active: active (running) since Mon 2018-11-26 15:22:46 CST; 28s ago  [root@mon3 postgresql\_data]# su - postgres  # 如果不用systemctl进行管理，用以下命令启动也行  # -bash-4.2$ /usr/bin/pg\_ctl -D /data/postgresql\_data -l logfile start  -bash-4.2$ psql  postgres=# \passwd postgres  Invalid command \passwd. Try \? for help.  postgres=# \password postgres  Enter new password:  Enter it again:  postgres=# create database paddles;  postgres=# \l  List of databases  Name | Owner | Encoding | Collate | Ctype | Access privileges  -----------+----------+----------+-------------+-------------+-----------------------  paddles | postgres | UTF8 | en\_US.UTF-8 | en\_US.UTF-8 |  postgres | postgres | UTF8 | en\_US.UTF-8 | en\_US.UTF-8 |  template0 | postgres | UTF8 | en\_US.UTF-8 | en\_US.UTF-8 | =c/postgres +  | | | | | postgres=CTc/postgres  template1 | postgres | UTF8 | en\_US.UTF-8 | en\_US.UTF-8 | =c/postgres +  | | | | | postgres=CTc/postgres  (4 rows)  postgres=# \q  -bash-4.2$ |

2. paddles配置:

|  |
| --- |
| [root@mon3 ~]# useradd paddles  [root@mon3 ~]# echo paddles:paddles|chpasswd  [root@mon3 ~]# su - paddles  [paddles@mon3 ~]$ git clone https://github.com/ceph/paddles.git  [paddles@mon3 ~]$ cd paddles/  [paddles@mon3 paddles]$ virtualenv ./virtualenv  New python executable in /home/paddles/paddles/virtualenv/bin/python  Installing setuptools, pip, wheel...done.  [paddles@mon3 paddles]$ cp config.py.in config.py  [paddles@mon3 paddles]$ vim config.py  …  server = {  'port': '8081',  'host': '10.129.4.103'  }  …  'url': 'postgresql+psycopg2://postgres:postgres@localhost/paddles',  …  [paddles@mon3 paddles]$ source virtualenv/bin/activate  (virtualenv) [paddles@mon3 paddles]$ pip install -r requirements.txt  (virtualenv) [paddles@mon3 paddles]$ python setup.py develop  (virtualenv) [paddles@mon3 paddles]$ pecan populate config.py  ==> LOADING ENVIRONMENT  ==> BUILDING SCHEMA  ==> STARTING A TRANSACTION...  No handlers could be found for logger "sqlalchemy.engine.base.Engine"  ==> COMMITING...  (virtualenv) [paddles@mon3 paddles]$ cp alembic.ini.in alembic.ini  (virtualenv) [paddles@mon3 paddles]$ vim alembic.ini  …  sqlalchemy.url = postgresql+psycopg2://postgres:postgres@127.0.0.1/paddles  …  (virtualenv) [paddles@mon3 paddles]$ alembic stamp head  INFO [alembic.runtime.migration] Context impl PostgresqlImpl.  INFO [alembic.runtime.migration] Will assume transactional DDL.  INFO [alembic.runtime.migration] Running stamp\_revision -> 11e2594da07b  # 启动测试  (virtualenv) [paddles@mon3 paddles]$ pecan serve config.py  Starting server in PID 21250  serving on http://10.129.4.103:8081 |

3. pulpito配置:

|  |
| --- |
| [root@mon3 ~]# useradd pulpit  [root@mon3 ~]# echo pulpito:pulpito|chpasswd  [root@mon3 ~]# su - pulpito  [pulpito@mon3 ~]$ git clone https://github.com/ceph/pulpito.git  [pulpito@mon3 ~]$ cd pulpito/  [pulpito@mon3 pulpito]$ virtualenv ./virtualenv  [pulpito@mon3 pulpito]$ cp config.py.in prod.py  [pulpito@mon3 pulpito]$ vim prod.py  …  server = {  'port': 8080,  'host': '10.129.4.103'  }  paddles\_address = 'http://10.129.4.103:8081'  …  'debug': False,  …  [pulpito@mon3 pulpito]$ source virtualenv/bin/activate  (virtualenv) [pulpito@mon3 pulpito]$ pip install -r requirements.txt  # 启动测试  # 先启动paddles  (virtualenv) [pulpito@mon3 pulpito]$ python run.py |

4. supervisor配置:

|  |
| --- |
| [root@mon3 supervisord.d]# vim paddles.ini  [root@mon3 supervisord.d]# cat paddles.ini  [program:paddles]  user=paddles  environment=HOME="/home/paddles",USER="paddles"  directory=/home/paddles/paddles  command=/home/paddles/paddles/virtualenv/bin/gunicorn\_pecan -c gunicorn\_config.py config.py  autostart=true  autorestart=true  redirect\_stderr=true  stdout\_logfile = /home/paddles/paddles/paddles.out.log  stderr\_logfile = /home/paddles/paddles/paddles.err.log  [root@mon3 supervisord.d]# vim pulpito.ini  [root@mon3 supervisord.d]# cat pulpito.ini  [program:pulpito]  user=pulpito  directory=/home/pulpito/pulpito  command=/home/pulpito/pulpito/virtualenv/bin/python run.py  environment=HOME="/home/pulpito",USER="pulpito"  autostart=true  autorestart=true  redirect\_stderr=true  stdout\_logfile = /home/pulpito/pulpito/pulpito.out.log  stderr\_logfile = /home/pulpito/pulpito/pulpito.err.log  [root@mon3 supervisord.d]# systemctl enable supervisord  Created symlink from /etc/systemd/system/multi-user.target.wants/supervisord.service to /usr/lib/systemd/system/supervisord.service.  [root@mon3 supervisord.d]# systemctl start supervisord  [root@mon3 supervisord.d]# systemctl status supervisord  ● supervisord.service - Process Monitoring and Control Daemon  Loaded: loaded (/usr/lib/systemd/system/supervisord.service; enabled; vendor preset: disabled)  Active: active (running) since Tue 2018-11-27 09:56:21 CST; 6s ago  [root@mon3 supervisord.d]# supervisorctl start all  [root@mon3 supervisord.d]# supervisorctl status  paddles RUNNING pid 21987, uptime 0:00:08  pulpito RUNNING pid 21988, uptime 0:00:08 |

5. ntp服务配置:

|  |
| --- |
| [root@mon3 ~]# yum -y install ntp ntpdate  [root@mon3 ~]# vim /etc/ntp.conf  …  restrict 10.129.4.103 mask 255.255.255.0 nomodify  server 127.127.1.0  fudge 127.127.1.0 stratum 10  …  [root@mon3 ~]# systemctl restart ntpd  [root@mon3 ~]# ntpq -np  remote refid st t when poll reach delay offset jitter  ==============================================================================  \*127.127.1.0 .LOCL. 10 l 24 64 7 0.000 0.000 0.000  …  [root@mon3 ~]# hwclock -w |

6. git服务器配置:

|  |
| --- |
| [root@mon3 ~]# useradd git  [root@mon3 ~]# echo git:git | chpasswd  [git@mon3 ~]$ cd /data/git/  [git@mon3 git]$ git init --bare sample.git  # 此时客户端可以进行克隆了  root@admin:~# git clone git@10.129.4.103:/data/git/sample.git  # git无密钥配置  # 客户端命令  root@admin:~# ssh-keygen  root@admin:~# ssh-copy-id git@10.129.4.103 #输入git用户密码即可  # 如果不想让开发者使用git用户登录到系统，则/etc/passwd文档中设置git用户的shell  [root@mon3 ~]# cat /etc/passwd  …  git:x:1004:1004::/home/git:/usr/bin/git-shell  …  # git-daemon配置  [root@mon3 ~]# vim /etc/supervisord.d/git-daemon.ini  [root@mon3 ~]# cat /etc/supervisord.d/git-daemon.ini  [program:git-daemon]  user=git  environment=HOME="/home/git",USER="git"  directory=/home/git  command=/usr/libexec/git-core/git-daemon --base-path=/data/git --export-all --enable=upload-pack --enable=upload-archive --enable=receive-pack --syslog --inetd --verbose  autostart=true  autorestart=true  redirect\_stderr=true  stdout\_logfile = /home/git/git-daemon/git-daemon.out.log  stderr\_logfile = /home/git/git-daemon/git-daemon.err.log  [root@mon3 ~]# supervisorctl reload  [root@mon3 ~]# supervisorctl status  git-daemon RUNNING pid 25459, uptime 0:02:56  # HTTP安装  [root@mon3 httpd]# wget http://mirrors.tuna.tsinghua.edu.cn/apache//apr/apr-1.6.5.tar.gz  [root@mon3 httpd]# tar -zxvf apr-1.6.5.tar.gz  [root@mon3 httpd]# mkdir apr  [root@mon3 httpd]# cd apr-1.6.5  [root@mon3 apr-1.6.5]# ./configure --prefix=/data/httpd/apr/  [root@mon3 apr-1.6.5]# make && make install  [root@mon3 httpd]# wget http://mirrors.tuna.tsinghua.edu.cn/apache//apr/apr-util-1.6.1.tar.gz  [root@mon3 httpd]# tar -zxvf apr-util-1.6.1.tar.gz  [root@mon3 httpd]# mkdir apr-util  [root@mon3 httpd]# cd apr-util-1.6.1  [root@mon3 apr-util-1.6.1]# ./configure --prefix=/data/httpd/apr-util/ --with-apr=/data/httpd/apr/  [root@mon3 apr-util-1.6.1]# make && make install  [root@mon3 httpd]# wget https://sourceforge.net/projects/pcre/files/pcre/8.38/pcre-8.38.tar.gz  [root@mon3 httpd]# tar -zxvf pcre-8.38.tar.gz  [root@mon3 httpd]# mkdir pcre  [root@mon3 httpd]# cd pcre-8.38  [root@mon3 httpd-2.4.37]# ./configure --prefix=/data/httpd/pcre/  [root@mon3 httpd-2.4.37]# make && make install  # 如果httpd编译有问题，则将apr-util换成1.5版本  [root@mon3 httpd]# wget http://mirrors.shu.edu.cn/apache//httpd/httpd-2.4.37.tar.gz  tar -zxvf httpd-2.4.37.tar.gz  mkdir httpd  cd httpd-2.4.37  ./configure --prefix=/data/httpd/httpd --with-apr=/data/httpd/apr --with-apr-util=/data/httpd/apr-util --with-pcre=/data/httpd/pcre --enable-module=shared  make && make install  # gitweb配置  [root@mon3 ~]# yum search gitweb  [root@mon3 ~]# vim /etc/gitweb.conf  …  $projectroot = "/data/git"  …  # HTTP配置  [root@mon3 conf]# vim httpd.conf  …  LoadModule cgid\_module modules/mod\_cgid.so  …  ServerName 10.129.4.103:80  …  # 修改DocumentRoot  DocumentRoot "/var/www/git"  <Directory "/var/www/git">  Options +FollowSymLinks +ExecCGI  AllowOverride All  Require all granted  DirectoryIndex gitweb.cgi  AddHandler cgi-script .cgi  </Directory>  # 启动httpd  [root@mon3 bin]# ./apachectl -k start  # 新gitweb UI  [root@mon3 data]# git clone https://github.com/kogakure/gitweb-theme.git  [root@mon3 data]# cd gitweb-theme/  [root@mon3 gitweb-theme]# ./setup -vi -t /var/www/git/ --install # 注意备份 |

7. gitlab配置:

|  |
| --- |
| # x86  [root@mon3 ~]# curl -s https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.rpm.sh | sudo bash  [root@mon3 ~]# yum -y install gitlab-ce  [root@mon3 ~]# gitlab-ctl start  # arcch64  # gitlab源码安装http://blog.51cto.com/dongsong/1706335  [root@mon3 ~]# yum -y install libicu-devel patch gcc-c++ readline-devel zlib-devel libffi-devel openssl-devel make autoconf automake libtool bison libxml2-devel libxslt-devel libyaml-devel zlib-devel openssl-devel cpio expat-devel gettext-devel curl-devel perl-ExtUtils-CBuilder perl-ExtUtils-MakeMaker cmake pcre-devel kernel-devel bzip2 yarn  # git安装, 安装包地址换为https://mirrors.edge.kernel.org/pub/software/scm/git/git-2.19.1.tar.gz  [root@mon3 gitlab]# wget https://mirrors.edge.kernel.org/pub/software/scm/git/git-2.9.4.tar.gz  [root@mon3 gitlab]# tar -zxvf git-2.9.4.tar.gz  [root@mon3 gitlab]# mkdir git  [root@mon3 gitlab]# cd git-2.9.4  [root@mon3 git-2.9.4]# ./configure --prefix=/data/gitlab/git/  [root@mon3 git-2.9.4]# make -j48 && make install  # 安装ruby  [root@mon3 gitlab]# wget https://cache.ruby-lang.org/pub/ruby/2.5/ruby-2.5.3.tar.gz  [root@mon3 gitlab]# tar -zxvf ruby-2.5.3.tar.gz  [root@mon3 gitlab]# mkdir ruby  [root@mon3 gitlab]# cd ruby-2.5.3  [root@mon3 ruby-2.5.3]# ./configure --prefix=/data/gitlab/ruby --disable-install-rdoc  [root@mon3 ruby-2.5.3]# make -j40  [root@mon3 ruby-2.5.3]# make install  # 源码安装nodejs  [root@mon3 gitlab]# wget https://nodejs.org/dist/v10.14.0/node-v10.14.0.tar.gz  [root@mon3 gitlab]# tar -zxvf node-v10.14.0.tar.gz  [root@mon3 gitlab]# cd node-v10.14.0  [root@mon3 node-v10.14.0]# ./configure --prefix=/data/gitlab/nodejsc/  [root@mon3 node-v10.14.0]# make -j48  [root@mon3 node-v10.14.0]# make install  # 二进制安装nodejs  # [root@mon3 gitlab]# wget https://nodejs.org/dist/v10.14.0/node-v10.14.0-linux-arm64.tar.xz  [root@mon3 gitlab]# xz -dkv node-v10.14.0-linux-arm64.tar.xz  [root@mon3 gitlab]# tar -xvf node-v10.14.0-linux-arm64.tar  # 源码安装go  [root@mon3 gitlab]# wget https://studygolang.com/dl/golang/go1.11.src.tar.gz  [root@mon3 gitlab]# mkdir goc  [root@mon3 gitlab]# tar -zxvf go1.11.src.tar.gz  # 二进制安装go  [root@mon3 gitlab]# wget https://studygolang.com/dl/golang/go1.11.linux-arm64.tar.gz  [root@mon3 gitlab]# tar -zxvf go1.11.linux-arm64.tar.gz  # 安装redis  [root@mon3 gitlab]# wget http://download.redis.io/releases/redis-5.0.2.tar.gz  [root@mon3 gitlab]# tar -zxvf redis-5.0.2.tar.gz  [root@mon3 gitlab]# mkdir redis  [root@mon3 gitlab]# cd redis-5.0.2  [root@mon3 redis-5.0.2]# make  [root@mon3 redis-5.0.2]# cd src/  [root@mon3 src]# make install PREFIX=/data/gitlab/redis  [root@mon3 src]# cd ..  [root@mon3 redis-5.0.2]# mkdir /data/gitlab/redis/etc/  [root@mon3 redis-5.0.2]# cp redis.conf /data/gitlab/redis/etc/  # 进行redies配置， 查看附录部分  # 增加supervisor配置脚本redis.ini  [root@mon3 ~]# vim /etc/supervisord.d/redis.ini  [root@mon3 ~]# cat /etc/supervisord.d/redis.ini  [program:redis]  user=root  directory=/data/gitlab/redis  command=/data/gitlab/redis/bin/redis-server /data/gitlab/redis/etc/redis.conf  environment=HOME="root",USER="root"  autostart=true  autorestart=true  redirect\_stderr=true  stdout\_logfile = /data/gitlab/redis/log/redis.out.log  stderr\_logfile = /data/gitlab/redis/log/redis.err.log  [root@mon3 ~]# supervisorctl start redis  [root@mon3 ~]# supervisorctl status redis  redis RUNNING pid 46890, uptime 0:02:08  # mariadb安装  [root@mon3 gitlab]# wget https://downloads.mariadb.org/interstitial/mariadb-10.3.11/source/mariadb-10.3.11.tar.gz  [root@mon3 gitlab]# tar -zxvf mariadb-10.3.11.tar.gz  [root@mon3 gitlab]# mkdir -p mariadb/etc  [root@mon3 gitlab]# mkdir -p mariadb/datadir  [root@mon3 mariadb-10.3.11]# cmake . -DCMAKE\_INSTALL\_PREFIX=/data/gitlab/mariadb/ -DMYSQL\_DATADIR=/data/gitlab/mariadb/datadir -DSYSCONFDIR=/data/gitlab/mariadb/etc -DWITHOUT\_TOKUDB=1 -DWITH\_INNOBASE\_STORAGE\_ENGINE=1 -DWITH\_ARCHIVE\_STPRAGE\_ENGINE=1 -DWITH\_BLACKHOLE\_STORAGE\_ENGINE=1 -DWIYH\_READLINE=1 -DWIYH\_SSL=system -DVITH\_ZLIB=system -DWITH\_LOBWRAP=0 -DMYSQL\_UNIX\_ADDR=/tmp/mysql.sock -DDEFAULT\_CHARSET=utf8 -DDEFAULT\_COLLATION=utf8\_general\_ci  [root@mon3 mariadb-10.3.11]# make -j48  [root@mon3 mariadb-10.3.11]# make install  [root@mon3 mariadb-10.3.11]# cp support-files/mysql.server /etc/init.d/mariadb  [root@mon3 mariadb-10.3.11]# chmod u+x /etc/init.d/mariadb  [root@mon3 mariadb-10.3.11]# /etc/init.d/mariadb configtest  Testing MariaDB configuration syntax SUCCESS! Syntax OK  [root@mon3 ~]# useradd mysql  [root@mon3 ~]# echo mysql:mysql |chpasswd  [root@mon3 ~]# chown -R mysql:mysql /data/gitlab/mariadb  [root@mon3 ~]# cd /data/gitlab/mariadb/scripts  [root@mon3 scripts]# ./mysql\_install\_db --basedir=/data/gitlab/mariadb --datadir=/data/gitlab/mariadb/datadir/ --user=mysql  # 启动数据库  [root@mon3 scripts]# /etc/init.d/mariadb start  Reloading systemd: [ OK ]  Starting mariadb (via systemctl): [ OK ]  [root@mon3 scripts]# /etc/init.d/mariadb status  SUCCESS! MariaDB running (47677)  # 配置数据库  [mysql@mon3 ~]$ mysql -u root -p  Enter password:# 密码为空，直接回车即可  MariaDB [(none)]> grant all privileges on \*.\* to 'root'@'%' identified by 'root' with grant option;  Query OK, 0 rows affected (0.001 sec) # 更新root密码  MariaDB [(none)]> grant all privileges on \*.\* to 'root'@'localhost' identified by 'root' with grant option;  Query OK, 0 rows affected (0.001 sec)  MariaDB [(none)]> flush privileges;  Query OK, 0 rows affected (0.001 sec)  # 创建数据库  MariaDB [(none)]> CREATE DATABASE gitlabhq\_production;  Query OK, 1 row affected (0.001 sec)  # 创建用户和授权  MariaDB [(none)]> grant all on gitlabhq\_production.\* to 'git'@'localhost' identified by 'git';  Query OK, 0 rows affected (0.001 sec)  MariaDB [(none)]> grant all on gitlabhq\_production.\* to 'git'@'%' identified by 'git';  Query OK, 0 rows affected (0.001 sec)  MariaDB [(none)]> grant all on gitlabhq\_production.\* to 'git'@'127.0.0.1' identified by 'git';  Query OK, 0 rows affected (0.001 sec)  MariaDB [(none)]> flush privileges;  Query OK, 0 rows affected (0.001 sec)  MariaDB [(none)]> quit;  Bye  # 增加/etc/profile配置  [root@mon3 ~]# vim /etc/profile  …  PATH=/data/gitlab/git/bin:$PATH  PATH=/data/gitlab/redis/bin:$PATH  PATH=/data/gitlab/nodejs/bin:$PATH  export GOROOT=/data/gitlab/go  export GOARCH=arm64  export GOOS=linux  export GOBIN=$GOROOT/bin  export GOPATH=/data/gitlab/go/gopath  PATH=$GOBIN:$PATH  PATH=/data/gitlab/ruby/bin:$PATH  PATH=/data/gitlab/mariadb/bin:$PATH  export PATH  …  [root@mon3 ~]# source /etc/profile  # ruby源配置  [root@mon3 ~]# gem sources --add https://gems.ruby-china.com/ --remove https://rubygems.org/  https://gems.ruby-china.com/ added to sources  https://rubygems.org/ removed from sources  [root@mon3 ~]# gem sources -l  \*\*\* CURRENT SOURCES \*\*\*  https://gems.ruby-china.com/  [root@mon3 ~]# gem install bundler --no-ri --no-rdoc  Fetching: bundler-1.17.1.gem (100%)  Successfully installed bundler-1.17.1  1 gem installed  # 安装yarn  [root@mon3 ~]# curl --silent --location https://dl.yarnpkg.com/rpm/yarn.repo | sudo tee /etc/yum.repos.d/yarn.repo  [root@mon3 ~]# yum -y install yarn  # 配置gitlab  [root@mon3 gitlab]# git clone https://gitlab.com/gitlab-org/gitlab-ce.git  [root@mon3 gitlab]# chown -R git:git gitlab-ce/  [root@mon3 gitlab]# su git  [git@mon3 gitlab]$ cd gitlab-ce/  [git@mon3 gitlab-ce]$ cd config  [git@mon3 config]$ cp gitlab.yml.example gitlab.yml  [git@mon3 config]$ vim gitlab.yml  …  host: 10.129.4.103  storages: # You must have at least a `default` storage path.  default:  path: /data/gitlab/gitlab-ce/repositories  git:  bin\_path: /data/gitlab/git/bin/git  …  [git@mon3 config]$ cp secrets.yml.example secrets.yml  [git@mon3 config]$ chmod 0600 secrets.yml  [git@mon3 config]$ cp unicorn.rb.example unicorn.rb  [root@mon3 config]# vim unicorn.rb  …  working\_directory "/data/gitlab/gitlab-ce"  listen "/data/gitlab/gitlab-ce/tmp/sockets/gitlab.socket"  pid "/data/gitlab/gitlab-ce/tmp/pids/unicorn.pid"  stderr\_path "/data/gitlab/gitlab-ce/log/unicorn.stderr.log"  stdout\_path "/data/gitlab/gitlab-ce/log/unicorn.stdout.log"  require\_relative "/data/gitlab/gitlab-ce/lib/gitlab/cluster/lifecycle\_events":  …  [git@mon3 config]$ cp initializers/rack\_attack.rb.example initializers/rack\_attack.rb  [git@mon3 config]$ cd ..  [git@mon3 gitlab-ce]$ git config --global core.autocrlf input  [git@mon3 gitlab-ce]$ cp config/resque.yml.example config/resque.yml  [git@mon3 gitlab-ce]$ vim config/resque.yml  …  production:  # Redis (single instance)  # url: unix:/var/run/redis/redis.sock  url: redis://127.0.0.1:6379  …  [git@mon3 gitlab-ce]$ cp config/database.yml.mysql config/database.yml  [git@mon3 gitlab-ce]$ vim config/database.yml  production:  ...  password: "git"  socket: /var/lib/mysql/mysql.sock  …  [git@mon3 gitlab-ce]$ chmod o-rwx config/database.yml  # 修改Gemfile  [git@mon3 gitlab-ce]$ vim Gemfile  …  source 'https://gems.ruby-china.com/'  …  [git@mon3 gitlab-ce]$ bundle install --deployment --without development test postgres aws Kerberos  # 安装gitlab-shell  [root@mon3 gitlab-ce]# bundle exec rake gitlab:shell:install REDIS\_URL=redis://127.0.0.1:6379 RAILS\_ENV=production  “”“  # gitlab-workhorse  [root@mon3 gitlab]# git clone https://gitlab.com/gitlab-org/gitlab-workhorse.git  root@mon3 gitlab]# mkdir gitlab-work  [root@mon3 gitlab]# cd gitlab-workhorse/  [root@mon3 gitlab-workhorse]# make install PREFIX=/data/gitlab/gitlab-work  [root@mon3 gitlab-work]# vim /etc/profile  …  PATH=/data/gitlab/gitlab-work/bin:$PATH  …  ““”“  # 安装gitlab-workhorse  [git@mon3 gitlab-ce]$ bundle exec rake "gitlab:workhorse:install[/home/git/gitlab-workhorse]" RAILS\_ENV=production  # 安装gitpages  [git@mon3 gitlab-ce]$ cd  [git@mon3 ~]$ git clone https://gitlab.com/gitlab-org/gitlab-pages.git  [git@mon3 ~]$ cd gitlab-pages/  [git@mon3 gitlab-pages]$ git checkout v$(</data/gitlab/gitlab-ce/VERSION)  [git@mon3 gitlab-pages]$ make  # 安装Gitaly  [git@mon3 gitlab-ce]$ bundle exec rake "gitlab:gitaly:install[/home/git/gitaly,/data/gitlab/gitlab-ce/repositories,https://gitlab.com/gitlab-org/gitaly.git]" RAILS\_ENV=production  [git@mon3 gitlab-ce]$ cd /home/git/gitaly/  [git@mon3 gitlab-ce]$ cd /home/git/gitaly/  [git@mon3 gitaly]$ chmod 0700 tmp/sockets/private  [git@mon3 gitaly]$ chown git tmp/sockets/private  # 初始化数据库和激活  [git@mon3 gitlab-ce]$ bundle exec rake gitlab:setup RAILS\_ENV=production GITLAB\_ROOT\_PASSWORD=root  # 初始化脚本配置  [root@mon3 gitlab-ce]# cp lib/support/init.d/gitlab /etc/init.d/gitlab  [root@mon3 gitlab-ce]# cp lib/support/init.d/gitlab.default.example /etc/default/gitlab  [root@mon3 gitlab-ce]# cp lib/support/logrotate/gitlab /etc/logrotate.d/gitlab  [root@mon3 gitlab-ce]# vim /etc/default/gitlab  …  app\_root=/data/gitlab/gitlab-ce  gitlab\_workhorse\_dir=/home/git/gitlab-workhorse  gitaly\_dir=/home/git/gitaly  …  # 检查环境配置  [git@mon3 gitlab-ce]$ bundle exec rake gitlab:env:info RAILS\_ENV=production  System information  System:  Current User: git  Using RVM: no  Ruby Version: 2.5.3p105  Gem Version: 2.7.6  Bundler Version:1.17.1  Rake Version: 12.3.1  Redis Version: 5.0.2  Git Version: 2.9.4  Sidekiq Version:5.2.3  Go Version: go1.11 linux/arm64  GitLab information  Version: 11.6.0-pre  Revision: e0e813b  Directory: /data/gitlab/gitlab-ce  DB Adapter: mysql2  URL: http://10.129.4.103  HTTP Clone URL: http://10.129.4.103/some-group/some-project.git  SSH Clone URL: git@10.129.4.103:some-group/some-project.git  Using LDAP: no  Using Omniauth: yes  Omniauth Providers:  GitLab Shell  Version: 8.4.1  Repository storage paths:  - default: /data/gitlab/gitlab-ce/repositories  Hooks: /home/git/gitlab-shell/hooks  Git: /data/gitlab/git/bin/git  [git@mon3 gitlab-ce]$  # 编译GetText PO  [git@mon3 gitlab-ce]$ bundle exec rake gettext:compile RAILS\_ENV=production  # 编译Assert  [git@mon3 gitlab-ce]$ yarn install --production --pure-lockfile  [git@mon3 gitlab-ce]$ bundle exec rake gitlab:assets:compile RAILS\_ENV=production NODE\_ENV=production  # nginx安装  [root@mon3 data]# mkdir nginx  [root@mon3 data]# cd nginx/  [root@mon3 nginx]# wget https://sourceforge.net/projects/pcre/files/pcre/8.40/pcre-8.40.tar.gz  [root@mon3 nginx]# tar -zxvf pcre-8.40.tar.gz  [root@mon3 nginx]# wget http://zlib.net/zlib-1.2.11.tar.gz  [root@mon3 nginx]# tar -zxvf zlib-1.2.11.tar.gz  [root@mon3 nginx]# cd zlib-1.2.11  [root@mon3 zlib-1.2.11]# ./configure  [root@mon3 zlib-1.2.11]# make && make install  [root@mon3 nginx]# wget https://www.openssl.org/source/openssl-1.0.1t.tar.gz  [root@mon3 nginx]# tar -zxvf openssl-1.0.1t.tar.gz  [root@mon3 nginx]# wget http://nginx.org/download/nginx-1.14.1.tar.gz  [root@mon3 nginx]# tar -zxvf nginx-1.14.1.tar.gz  [root@mon3 nginx]# mkdir nginx  [root@mon3 nginx]# cd nginx-1.14.1  [root@mon3 nginx-1.14.1]# [root@mon3 nginx-1.14.1]# ./configure --prefix=/data/nginx/nginx --with-pcre=/data/nginx/pcre-8.40 --with-openssl=/data/nginx/openssl-1.0.1t --with-http\_realip\_module  [root@mon3 nginx-1.14.1]# make -j48 && make install  # 配置ngxin.conf,参考附录nginx.conf  #启动gitlab  [root@mon3 conf]# service gitlab start  Starting GitLab Unicorn  The Sidekiq job dispatcher is already running with pid 16076, not restarting  The GitLab Workhorse is already running with pid 15998, not restarting  Gitaly is already running with pid 16008, not restarting  The GitLab Unicorn web server with pid 17775 is running.  The GitLab Sidekiq job dispatcher with pid 16076 is running.  The GitLab Workhorse with pid 15998 is running.  Gitaly with pid 16008 is running.  GitLab and all its components are up and running.  [git@mon3 gitlab-ce]$ bundle exec rake gitlab:check RAILS\_ENV=production |

8. jenkins配置:

|  |
| --- |
| # 添加源  [cpu@mon3 jekins]$ sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo  [cpu@mon3 jekins]$ sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key  # 下载jenkins  [cpu@mon3 jekins]$ sudo yum -y install jenkins  # 运行jenkins  [cpu@mon3 jekins]$ rpm -ql jenkins  /etc/init.d/jenkins  /etc/logrotate.d/jenkins  /etc/sysconfig/jenkins  /usr/lib/jenkins  /usr/lib/jenkins/jenkins.war  /usr/sbin/rcjenkins  /var/cache/jenkins  /var/lib/jenkins  /var/log/Jenkins  [cpu@mon3 jenkins]$ sudo vim /etc/sysconfig/jenkins  …  JENKINS\_HOME=" /data/jenkins "  JENKINS\_PORT="9000"  JENKINS\_LISTEN\_ADDRESS="10.129.4.103"  …  [cpu@mon3 jenkins]$ sudo systemctl enable Jenkins  [cpu@mon3 jenkins]$ sudo systemctl start Jenkins  [cpu@mon3 jenkins]$ sudo systemctl status jenkins  ● jenkins.service - LSB: Jenkins Automation Server  Loaded: loaded (/etc/rc.d/init.d/jenkins; bad; vendor preset: disabled)  Active: active (running) since Mon 2018-11-26 10:24:25 CST; 5min ago  Docs: man:systemd-sysv-generator(8)  # 浏览器查看  # 浏览器输入http://10.129.4.103:9000/  # admin密码在相应的目录/home/cpu/jenkins/secrets/initialAdminPassword  # 安装推荐的插件即可  # 可选配置用户名和密码  # 确认Jenkins地址后安装完成  # 安装Jenkins插件PostBuildScript, Dynamic Axis, Conditional BuildStep Role-based Authorization Strategy |

9. ceph-build配置:

|  |
| --- |
| # 安装jenkins-job-builder  [cpu@mon3 pip]$ sudo pip install jenkins-job-builder  # 下载ceph-build  [cpu@mon3 ceph-build]$ git clone https://github.com/ceph/ceph-build.git |

10. jenkins-job-builder

|  |
| --- |
| # 安装jenkins-job-builder  [cpu@mon3 pip]$ sudo pip install jenkins-job-builder  # 配置  # 创建文件~/.config/jenkins\_jobs/jenkins\_jobs.ini 或/etc/jenkins\_jobs/jenkins\_jobs.ini  [cpu@mon3 ~]$ vim ~/.config/jenkins\_jobs/jenkins\_jobs.ini  [cpu@mon3 ~]$ cat ~/.config/jenkins\_jobs/jenkins\_jobs.ini  [job\_builder]  ignore\_cache=True  keep\_descriptions=False  include\_path=.:scripts:~/git/  recursive=False  exclude=.\*:manual:./development  allow\_duplicates=False  update=all  [jenkins]  user=jenkins  password=jenkins  url=http://10.129.4.103:9000  query\_plugins\_info=False  # 创建jenkins任务  [cpu@mon3 definitions]$ jenkins-jobs update /home/cpu/ceph-build/ceph-build/ceph-build/config/definitions/ceph-build.yml  # 下载jenkins-job-builder源码  [cpu@mon3 jenkins-job-builder]$ git clone https://git.openstack.org/openstack-infra/jenkins-job-builder  [cpu@mon3 jenkins-job-builder]$ cd jenkins-job-builder/  # 测试job配置  [cpu@mon3 jenkins-job-builder]$ jenkins-jobs test tests/yamlparser/fixtures/templates002.yaml  # 更新job配置 |

11. teuthology配置:

|  |
| --- |
| [root@mon3 ~]# yum -y install git python-devel python-pip python-virtualenv  libevent-devel libvirt-python beanstalkd redhat-lsb mariadb-devel libev-devel libvirt-devel  [root@mon3 ~]# useradd teuthology  [root@mon3 ~]# echo teuthology:teuthology|chpasswd  [root@mon3 ~]# useradd teuthworker  [root@mon3 ~]# echo teuthworker:teuthworker|chpasswd  [root@mon3 /]# vim /etc/sudoers  # 追加  teuthology ALL=(ALL) NOPASSWD:ALL  teuthworker ALL=(ALL) NOPASSWD:ALL  [root@mon3 ~]# su – teuthology  [teuthology@mon3 ~]$ vim .teuthology.yaml  lock\_server: 'http://10.129.4.103:8081'  results\_server: 'http://10.129.4.103:8081'  lab\_domain: ‘hxtarm.com'  queue\_host: 127.0.0.1  queue\_port: 11300  archive\_base: /data/cpu/ceph-test/archive  verify\_host\_keys: false  ceph\_git\_base\_url: http://10.129.4.103/storage  ceph\_qa\_suite\_git\_url: http://10.129.4.103/storage/ceph.git  ceph\_git\_url: http://10.129.4.103/storage/ceph.git  gitbuilder\_host: '10.129.4.103'  reserve\_machines: 1  archive\_server: http://10.129.4.103/teuthology/  max\_job\_time: 86400  [teuthology@mon3 ~]$ mkdir src  [teuthology@mon3 ~]$ cd src  [teuthology@mon3 src]$ git clone https://github.com/ceph/teuthology.git teuthology\_master  [teuthology@mon3 src]$ cd teuthology\_master/  [teuthology@mon3 teuthology\_master]$ ./bootstrap  [teuthology@mon3 teuthology\_master]$ source virtualenv/bin/activate  (virtualenv) [teuthology@mon3 teuthology\_master]$ wget https://raw.githubusercontent.com/ceph/teuthology/master/docs/\_static/create\_nodes.py  (virtualenv) [teuthology@mon3 teuthology\_master]$ vim create\_nodes.py  …  paddles\_url = 'http://10.129.4.103:8081'  machine\_type = 'plana'  lab\_domain = 'hxtarm.com'  user = 'ubuntu'  machine\_index\_range = range(3, 4) #设置添加的机器数量  …  # 免密配置  # 解析配置  (virtualenv) [teuthology@mon3 teuthology\_master]$ python create\_nodes.py  [root@mon3 sbin]# su - teuthworker  [teuthworker@mon3 ~]$ mkdir src  [teuthworker@mon3 ~]$ git clone https://github.com/ceph/teuthology.git src/teuthology\_master  [teuthworker@mon3 ~]$ cd src/teuthology\_master/  [teuthworker@mon3 teuthology\_master]$ ./bootstrap |

12. 附录：

* redis配置

|  |
| --- |
| daemonize no  tcp-backlog 60000  loglevel warning  logfile "/data/gitlab/redis/redis.log"  dir /data/gitlab/redis/  maxclients 60000 |

* my.cnf

|  |
| --- |
| [client]  port = 3306  socket = /var/lib/mysql/mysql.sock  [mysqld]  user = mysql  port = 3306  datadir=/data/gitlab/mariadb/datadir  socket=/var/lib/mysql/mysql.sock  log-bin = bin.log  log-error = error.log  pid-file = mysql.pid  long\_query\_time = 1  slow\_query\_log  slow\_query\_log\_file = slow.log  binlog\_cache\_size = 4M  binlog\_format = mixed  max\_binlog\_cache\_size = 16M  max\_binlog\_size = 1G  expire\_logs\_days = 30  ft\_min\_word\_len = 4  back\_log = 4096  max\_allowed\_packet = 64M  max\_connections = 65536  max\_connect\_errors = 100  join\_buffer\_size = 1M  read\_buffer\_size = 1M  read\_rnd\_buffer\_size = 1M  sort\_buffer\_size = 1M  query\_cache\_size = 64M  table\_open\_cache = 2048  thread\_cache\_size = 256  max\_heap\_table\_size = 64M  tmp\_table\_size = 64M  thread\_stack = 192K  connect\_timeout = 600  interactive\_timeout = 600  wait\_timeout = 600  thread\_concurrency = 8  local-infile = 0  skip-show-database  skip-name-resolve  skip-external-locking  #\*\*\* MyISAM  key\_buffer\_size = 512M  bulk\_insert\_buffer\_size = 64M  myisam\_sort\_buffer\_size = 32M  myisam\_max\_sort\_file\_size = 1G  myisam\_repair\_threads = 1  myisam\_recover  ##\*\*\* INNODB  innodb\_buffer\_pool\_size = 4G  # innodb\_additional\_mem\_pool\_size = 32M  innodb\_data\_file\_path = ibdata:32M:autoextend  innodb\_read\_io\_threads = 8  innodb\_write\_io\_threads = 8  innodb\_file\_per\_table = 1  innodb\_flush\_log\_at\_trx\_commit = 2  innodb\_lock\_wait\_timeout = 120  innodb\_log\_buffer\_size = 8M  innodb\_log\_file\_size = 256M  innodb\_log\_files\_in\_group = 3  innodb\_max\_dirty\_pages\_pct = 10  innodb\_thread\_concurrency = 16  #innodb\_force\_recovery = 4  #\*\*\* Replication Slave  ##read-only  ##skip-slave-start  relay-log = relay.log  log-slave-updates  # Disabling symbolic-links is recommended to prevent assorted security risks  symbolic-links=0  # Settings user and group are ignored when systemd is used.  # If you need to run mysqld under a different user or group,  # customize your systemd unit file for mariadb according to the  # instructions in http://fedoraproject.org/wiki/Systemd  [mysqld\_safe]  log-error=/var/log/mariadb/mariadb.log  pid-file=/var/run/mariadb/mariadb.pid  open-files-limit = 60000  [mysqldump]  quick  max\_allowed\_packet = 32M  [mysql]  no-auto-rehash  [myisamchk]  key\_buffer\_size = 512M  sort\_buffer\_size = 512M  read\_buffer = 8M  write\_buffer = 8M  [mysqlhotcopy]  interactive-timeout  #  # include all files from the config directory  #  !includedir /etc/my.cnf.d |

* nginx.conf

|  |
| --- |
| worker\_processes 1;  #error\_log logs/error.log;  #error\_log logs/error.log notice;  #error\_log logs/error.log info;  #pid logs/nginx.pid;  events {  worker\_connections 1024;  }  http {  include mime.types;  default\_type application/octet-stream;  #log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '  # '$status $body\_bytes\_sent "$http\_referer" '  # '"$http\_user\_agent" "$http\_x\_forwarded\_for"';  #access\_log logs/access.log main;  sendfile on;  #tcp\_nopush on;  #keepalive\_timeout 0;  keepalive\_timeout 65;  #gzip on;  upstream gitlab-workhorse {  # GitLab socket file,  # for Omnibus this would be: unix:/var/opt/gitlab/gitlab-workhorse/socket  server unix:/data/gitlab/gitlab-ce/tmp/sockets/gitlab-workhorse.socket fail\_timeout=0;  }  map $http\_upgrade $connection\_upgrade\_gitlab {  default upgrade;  '' close;  }  ## NGINX 'combined' log format with filtered query strings  log\_format gitlab\_access $remote\_addr - $remote\_user [$time\_local] "$request\_method $gitlab\_filtered\_request\_uri $server\_protocol" $status $body\_bytes\_sent "$gitlab\_filtered\_http\_referer" "$http\_user\_agent";  ## Remove private\_token from the request URI  # In: /foo?private\_token=unfiltered&authenticity\_token=unfiltered&feed\_token=unfiltered&...  # Out: /foo?private\_token=[FILTERED]&authenticity\_token=unfiltered&feed\_token=unfiltered&...  map $request\_uri $gitlab\_temp\_request\_uri\_1 {  default $request\_uri;  ~(?i)^(?<start>.\*)(?<temp>[\?&]private[\-\_]token)=[^&]\*(?<rest>.\*)$ "$start$temp=[FILTERED]$rest";  }  ## Remove authenticity\_token from the request URI  # In: /foo?private\_token=[FILTERED]&authenticity\_token=unfiltered&feed\_token=unfiltered&...  # Out: /foo?private\_token=[FILTERED]&authenticity\_token=[FILTERED]&feed\_token=unfiltered&...  map $gitlab\_temp\_request\_uri\_1 $gitlab\_temp\_request\_uri\_2 {  default $gitlab\_temp\_request\_uri\_1;  ~(?i)^(?<start>.\*)(?<temp>[\?&]authenticity[\-\_]token)=[^&]\*(?<rest>.\*)$ "$start$temp=[FILTERED]$rest";  }  ## Remove feed\_token from the request URI  # In: /foo?private\_token=[FILTERED]&authenticity\_token=[FILTERED]&feed\_token=unfiltered&...  # Out: /foo?private\_token=[FILTERED]&authenticity\_token=[FILTERED]&feed\_token=[FILTERED]&...  map $gitlab\_temp\_request\_uri\_2 $gitlab\_filtered\_request\_uri {  default $gitlab\_temp\_request\_uri\_2;  ~(?i)^(?<start>.\*)(?<temp>[\?&]feed[\-\_]token)=[^&]\*(?<rest>.\*)$ "$start$temp=[FILTERED]$rest";  }  ## A version of the referer without the query string  map $http\_referer $gitlab\_filtered\_http\_referer {  default $http\_referer;  ~^(?<temp>.\*)\? $temp;  }  ## Normal HTTP host  server {  ## Either remove "default\_server" from the listen line below,  ## or delete the /etc/nginx/sites-enabled/default file. This will cause gitlab  ## to be served if you visit any address that your server responds to, eg.  ## the ip address of the server (http://x.x.x.x/)n 0.0.0.0:80 default\_server;  listen 0.0.0.0:80 default\_server;  listen [::]:80 default\_server;  server\_name YOUR\_SERVER\_FQDN; ## Replace this with something like gitlab.example.com  server\_tokens off; ## Don't show the nginx version number, a security best practice  ## See app/controllers/application\_controller.rb for headers set  ## Real IP Module Config  ## http://nginx.org/en/docs/http/ngx\_http\_realip\_module.html  real\_ip\_header X-Real-IP; ## X-Real-IP or X-Forwarded-For or proxy\_protocol  real\_ip\_recursive off; ## If you enable 'on'  ## If you have a trusted IP address, uncomment it and set it  # set\_real\_ip\_from YOUR\_TRUSTED\_ADDRESS; ## Replace this with something like 192.168.1.0/24  ## Individual nginx logs for this GitLab vhost  access\_log /data/nginx/nginx/logs/gitlab\_access.log gitlab\_access;  error\_log /data/nginx/nginx/logs/gitlab\_error.log;  location / {  client\_max\_body\_size 0;  gzip off;  ## https://github.com/gitlabhq/gitlabhq/issues/694  ## Some requests take more than 30 seconds.  proxy\_read\_timeout 300;  proxy\_connect\_timeout 300;  proxy\_redirect off;  proxy\_http\_version 1.1;  proxy\_set\_header Host $http\_host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_set\_header X-Forwarded-Proto $scheme;  proxy\_set\_header Upgrade $http\_upgrade;  proxy\_set\_header Connection $connection\_upgrade\_gitlab;  proxy\_pass http://gitlab-workhorse;  }  error\_page 404 /404.html;  error\_page 422 /422.html;  error\_page 500 /500.html;  error\_page 502 /502.html;  error\_page 503 /503.html;  location ~ ^/(404|422|500|502|503)\.html$ {  # Location to the GitLab's public directory,  # for Omnibus this would be: /opt/gitlab/embedded/service/gitlab-rails/public.  root /data/gitlab/gitlab-ce/public;  internal;  }  }  } |

* 参考：

【1】在服务器上搭建 Git：https://git-scm.com/book/zh/v2/%E6%9C%8D%E5%8A%A1%E5%99%A8%E4%B8%8A%E7%9A%84-Git-%E5%9C%A8%E6%9C%8D%E5%8A%A1%E5%99%A8%E4%B8%8A%E6%90%AD%E5%BB%BA-Git

【2】gitlab源码安装：http://blog.51cto.com/dongsong/1706335

【3】go：https://studygolang.com/dl

【4】redis：https://www.cnblogs.com/limit1/p/9045183.html

【5】Installation from source：https://docs.gitlab.com/ce/install/installation.html

【6】yarn install：https://yarnpkg.com/en/docs/install#windows-stable

* 问题集：

|  |
| --- |
| Q：  unable to execute gcc: No such file or directory  error: command 'gcc' failed with exit status 1  A: gcc未安装  [root@mon3 ~]# yum -y install gcc gcc-c++  Q:  error: libpq-fe.h: No such file or directory  A:  [root@mon3 ~]# yum -y install postgresql-devel  Q:  FATAL: Ident authentication failed for user "postgres"  A:  参考：https://www.cyberciti.biz/faq/psql-fatal-ident-authentication-failed-for-user/和https://help.ubuntu.com/stable/serverguide/postgresql.html  -bash-4.2$ vim /data/postgresql\_data/pg\_hba.conf  # 添加该行  host all postgres 127.0.0.1/8 md5  [root@mon3 pg\_log]# systemctl restart postgresql # 更改后重启服务  Q:  In Gemfile:  re2  A:  [root@mon3 gitlab]# git clone https://github.com/google/re2.git  [root@mon3 gitlab]# cd re2/  [root@mon3 re2]# make  [root@mon3 re2]# make test  [root@mon3 re2]# make install  [root@mon3 re2]# make testinstall |