ECS服务器CentOS 安装Nginx

# yum安装nginx

yum install nginx -y

# 检查配置

检测文件 nginx.conf

nginx -t

提示中出现successfull就表明成功。

# 启动nginx

sudo systemctl start nginx.service

# 首页访问测试

curl 127.0.0.1

# 设置开机启动

sudo systemctl enable nginx.service

# 修改配置

进入nginx配置文件

vim /etc/nginx/nginx.conf

也可以用“FileZilla”工具拉取文件下来，本地修改，在上传。

# 重启nginx

systemctl restart nginx.service

# nginx常用命令

启动：systemctl start nginx.service

停止：systemctl stop nginx.service

重启：systemctl reload nginx.service

状态：systemctl status nginx.service

# 安装mysql

# 安装加速器

rpm -Uvh <http://dev.mysql.com/get/mysql57-community-release-el7-9.noarch.rpm>

# 查看

yum repolist all |grep mysql

默认下载5.7版本的MySQL数据库，而8.0版本的默认是关闭的，先把5.7的禁用，启用MySQL 8.0数据库。

# 打开文件, 修改设置

vi /etc/yum.repos.d/mysql-community.repo

输入命令后显示如下代码，每一个模块代表一个版本数据库。

其中可以看到5.7版本, enabled=1, 改为enabled=0, 禁用掉  
另外可以看到8.0版本, enabled=0, 改为enabled=1, 开启

# 修改完之后查看可以用的MySQL版本

# yum repolist enabled | grep mysql

# 正式开始安装MySQL

 yum install -y mysql-community-server

Linux CentOS7.6常用命令：

查看端口：netstat -ntlp

查看进程（如查看nginx进程）：ps -ef |grep nginx

# ps -ef |grep nginx

root 12969 1 0 Apr17 ? 00:00:00 nginx: master process nginx

nginx 12970 12969 0 Apr17 ? 00:00:00 nginx: worker process

nginx 12971 12969 0 Apr17 ? 00:00:00 nginx: worker process

之后，可以直接杀掉进程，命令：

# sudo kill -quit 12969

CentOS7安装chrome

cd /etc/yum.repos.d

sudo vim google-chrome.repo

在google-chrome.repo中输入：

[google-chrome]

name=google-chrome

baseurl=http://dl.google.com/linux/chrome/rpm/stable/$basearch

enabled=1

gpgcheck=1

gpgkey=https://dl-ssl.google.com/linux/linux\_signing\_key.pub

wget -q -O -https://dl-ssl.google.com/linux/linux\_signing\_key.pub | sudo apt-key add -

>>> sudo sh -c 'echo "deb <http://dl.google.com/linux/chrome/deb/> stable main" >> /etc/apt/sources.list.d/google-chrome.list'

Docker

1. 环境要求，3.10以上

查看环境命令：uname -a

1. yum源更新

yum update

1. docker安装nginx

创建挂载目录

mkdir /usr/local/nginx/conf

mkdir /usr/local/nginx/html

mkdir /usr/local/nginx/log

运行：

docker run --name nginx-service -p 80:80 -v /usr/local/nginx/conf/nginx.conf:/etc/nginx/nginx.conf -v /usr/local/nginx/html:/usr/share/nginx/html -v /usr/local/nginx/log:/var/log/nginx --restart always --privileged=true -d nginx

--web容器：

docker run --name nginx-web -p 81:80 -v /usr/local/nginx/conf/web.conf:/etc/nginx/nginx.conf -v /usr/local/nginx/html/web:/usr/share/nginx/html -v /usr/local/nginx/log/web:/var/log/nginx --restart always --privileged=true -d nginx

--admin容器：

docker run --name nginx-admin -p 82:80 -v /usr/local/nginx/conf/admin.conf:/etc/nginx/nginx.conf -v /usr/local/nginx/html/admin:/usr/share/nginx/html -v /usr/local/nginx/log/admin:/var/log/nginx --restart always --privileged=true -d nginx

--resource容器：

docker run --name nginx-resource -p 83:80 -v /usr/local/nginx/conf/resource.conf:/etc/nginx/nginx.conf -v /usr/local/nginx/html/resource:/usr/share/nginx/html -v /usr/local/nginx/log/resource:/var/log/nginx --restart always --privileged=true -d nginx

--api容器：

docker run --name nginx-api -p 84:80 -v /usr/local/nginx/conf/api.conf:/etc/nginx/nginx.conf -v /usr/local/nginx/log/api:/var/log/nginx --restart always --privileged=true -d nginx

nginx挂载配置：请注意root指向。

#工作进程数 1 ,不要超过计算机的核数,四核配置4,八核配置8

worker\_processes 1;

#工作连接数,也就是线程,一个进程有1024个线程,

events {

worker\_connections 1024;

}

#http请求配置

http {

default\_type application/octet-stream;

#sendfile为发送文件,要on开启

sendfile on;

#keepalive\_timeout超时时间

keepalive\_timeout 65;

server {

#监听的端口,这里为80

listen 80;

#server\_name就是域名,

server\_name localhost;

#location域名代理地址

# / 代表所有请求路径

location / {

#指向docker的地址

root /usr/share/nginx/html;

index index.html;

}

}

}

重启docker容器内的服务：

sudo docker exec -it nginx-api /etc/init.d/nginx reload

docker安装mysql

mkdir /usr/local/mysql/conf

mkdir /usr/local/mysql/data

mkdir /usr/local/mysql/log

在conf下编辑文件my.cnf

# Copyright (c) 2017, Oracle and/or its affiliates. All rights reserved.

#

# This program is free software; you can redistribute it and/or modify

# it under the terms of the GNU General Public License as published by

# the Free Software Foundation; version 2 of the License.

#

# This program is distributed in the hope that it will be useful,

# but WITHOUT ANY WARRANTY; without even the implied warranty of

# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

# GNU General Public License for more details.

#

# You should have received a copy of the GNU General Public License

# along with this program; if not, write to the Free Software

# Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

#

# The MySQL Server configuration file.

#

# For explanations see

# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

[mysqld]

pid-file = /var/run/mysqld/mysqld.pid

socket = /var/run/mysqld/mysqld.sock

datadir = /var/lib/mysql

secure-file-priv= NULL

# Disabling symbolic-links is recommended to prevent assorted security risks

symbolic-links=0

# Custom config should go here

!includedir /etc/mysql/conf.d/

docker run --restart=always -d -v /usr/local/mysql/conf/my.cnf:/etc/mysql/my.cnf -v /usr/local/mysql/logs:/logs -v /usr/local/mysql/data/mysql:/var/lib/mysql -p 3306:3306 --name mysql -e MYSQL\_ROOT\_PASSWORD=ndqazwsx2020 mysql:latest

nd2020qazwsx

nohup java -jar api-service-0.0.1-SNAPSHOT.jar >log.log &

<http://localhost:84/web/course?size=8>

<http://localhost:8081/web/course?size=8>