

# Dnsmasq 部署 DNS 服务(CentOS)

## Dnsmasq 部署 DNS 服务(CentOS)

### 软件部署

安装dnsmasq

检查端口占用

自定义配置dnsmasq配置

启动dnsmasq

检查状态

修改docker-compose文件

重新生成docker容器服务

### Centos开放端口

重启后操作

检查dns服务器

检查IP是否改变

启动nginx服务

## 软件部署

### 安装dnsmasq

```
$ yum install dnsmasq
```

### 检查端口占用

```
# 查看53端口是否被占用  
$ netstat -tunlp | grep 53
```

### 自定义配置dnsmasq配置

```
$ vi /etc/dnsmasq.d/address.conf

# 指定dnsmasq默认查询的上游服务器
server=114.114.114.114

# 把test.dns.com解析到特定的IP(下面是举例的测试域名和IP)
address=/test.commander.com/119.27.188.153
address=/test.qy.form.com/119.27.188.153
address=/test.wx.form.com/119.27.188.153
```

保存退出，确保本机53端口可对外访问

## 启动dnsmasq

```
$ service dnsmasq start
```

## 检查状态

```
$ netstat -tlunp | grep 53
```

如下图所示，开启成功。

```
[root@VM_16_12_centos ~]# netstat -tlunp | grep 53
tcp        0      0 0.0.0.0:53          0.0.0.0:*          LISTEN      24293/dnsmasq
tcp6       0      0 :::53              :::*                LISTEN      24293/dnsmasq
udp        0      0 0.0.0.0:53          0.0.0.0:*          24293/dnsmasq
udp6       0      0 :::53              :::*                24293/dnsmasq
```

## 修改docker-compose文件

在docker-compose文件中找到所有设置dns的services，添加dns服务器的IP（119.27.188.153为举例dns服务器IP），**自定义的dns域名服务要放在第一位，否则可能无法解析**，没有设置dns的services不用管。docker-compose文件有2个，在aistreamserver和formserver目录下。

```
12
13     zookeeper:
14         image: wurstmeister/zookeeper:latest
15         restart: always
16         container_name: aistream_zookeeper
17         dns:
18         - 119.27.188.153
19         - 114.114.114.114
20
21     kafka:
```

## 重新生成docker容器服务

```
# 进入对应的docker-compose目录下, 执行下面命令(aistreamserver和formserver目录都要执行)
$ sudo docker-compose up -d
```

## Centos开放端口

开放dns服务器访问端口（53）、网站服务端口（80、443）、grpc服务端口（12345）

```
firewall-cmd --zone=public --add-port=53/tcp --permanent
firewall-cmd --zone=public --add-port=53/udp --permanent
firewall-cmd --zone=public --add-port=80/tcp --permanent
firewall-cmd --zone=public --add-port=80/udp --permanent
firewall-cmd --zone=public --add-port=443/tcp --permanent
firewall-cmd --zone=public --add-port=443/udp --permanent
firewall-cmd --zone=public --add-port=12345/tcp --permanent
firewall-cmd --zone=public --add-port=12345/udp --permanent
firewall-cmd --reload
```

```
kill -9 30030
service dnsmasq start
service dnsmasq status
```

## 重启后操作

### 检查dns服务器

```
# 检查53端口情况
netstat -tunlp | grep 53
```

如图，表示被占用端口：

```
[root@localhost ~]# netstat -tunlp | grep 53
tcp        0      0 0.0.0.0:53 0.0.0.0:*        LISTEN      9889/dnsmasq
udp        0      0 0.0.0.0:53 0.0.0.0:*        8695/avahi-daemon:
udp        0      0 192.168.122.1:53 0.0.0.0:*        9889/dnsmasq
```

# 杀掉进程

```
kill -9 9889
```

# 开启dns服务

```
service dnsmasq start
```

# 查看服务状态, (如下图表示成功)

```
service dnsmasq status
```

```
[root@localhost ~]# service dnsmasq status
Redirecting to /bin/systemctl status dnsmasq.service
● dnsmasq.service - DNS caching server.
   Loaded: loaded (/usr/lib/systemd/system/dnsmasq.service; disabled; vendor preset: disabled)
   Active: active (running) since Wed 2019-01-02 11:51:06 CST; 7s ago
     Main PID: 27220 (dnsmasq)
        Tasks: 1
      Memory: 428.0K
     CGroup: /system.slice/dnsmasq.service
            └─27220 /usr/sbin/dnsmasq -k
```

# 查看端口情况

```
netstat -tunlp | grep 53
```

如图表示成功:

```
[root@localhost ~]# netstat -tunlp | grep 53
tcp        0      0 0.0.0.0:53          0.0.0.0:*          LISTEN     27220/dnsmasq
tcp6       0      0 :::53              :::*                LISTEN     27220/dnsmasq
udp        0      0 0.0.0.0:5353       0.0.0.0:*          8695/avahi-daemon:
udp        0      0 0.0.0.0:53         0.0.0.0:*          27220/dnsmasq
udp6       0      0 :::53              :::*                27220/dnsmasq
```

检查IP是否改变

# 检查本机网卡

```
ifconfig ens33
```

```
[root@localhost ~]# ifconfig ens33
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.2.6  netmask 255.255.255.0  broadcast 192.168.2.255
    inet6 fe80::5bf3:17cc:c977:9e85  prefixlen 64  scopeid 0x20<link>
    ether 00:0c:29:e8:e7:f9  txqueuelen 1000  (Ethernet)
    RX packets 142  bytes 20188 (19.7 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 101  bytes 11287 (11.0 KiB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

ip改变则修改相应的文

件, /etc/dnsmasq.d/address.conf, /etc/resolv.conf, /data1/aistreamserver/docker-compose.yml, /data1/formserver/docker-compose.yml, 修改相应的ip

修改成功后,

```
# 重启dns服务
service dnsmasq restart

# 进入/data1/aistreamserver
docker-compose up -d

# 进入/data1/formserver
docker-compose up -d
```

## 启动nginx服务

```
# 检查配置文件，找到配置文件路径
nginx -t
```

```
[root@localhost ~]# nginx -t
nginx: the configuration file /data1/server/nginx-1.15.4//conf/nginx.conf syntax is ok
nginx: configuration file /data1/server/nginx-1.15.4//conf/nginx.conf test is successful
```

```
# 执行配置文件
nginx -c /data1/server/nginx-1.15.4//conf/nginx.conf
```

如出现如图所示情况，先杀掉nginx进程

```
[root@localhost ~]# nginx -t
nginx: the configuration file /data1/server/nginx-1.15.4//conf/nginx.conf syntax is ok
nginx: configuration file /data1/server/nginx-1.15.4//conf/nginx.conf test is successful
[root@localhost ~]# nginx -c /data1/server/nginx-1.15.4//conf/nginx.conf
[root@localhost ~]# nginx -c /data1/server/nginx-1.15.4//conf/nginx.conf
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)
nginx: [emerg] still could not bind()
```

```
# 重新加载配置文件
nginx -s reload
```