## 配置管理平台部署文档

环境:

CentOS 6/7 x64

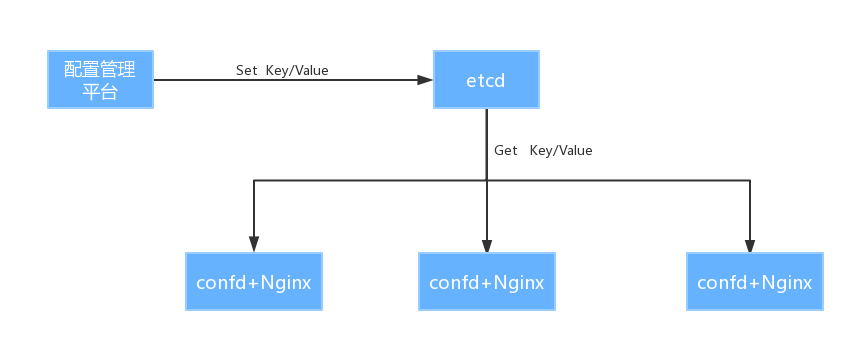
Python: 2.7.6

Etcd： 3.2.18

Confd: 0.16.0

Nginx: 1.12.1

#### 拓扑图:



#### 二．涉及软件

**etcd：**分布式KV存储系统，一般用于共享配置和服务注册与发现。是CoreOS公司发起的一个开源项目。 ETCD存储格式类似于文件系统，以根"/"开始下面一级级目录，最后一个是Key，一个key对应一个Value。

**etcd集群：**使用Raft协议保证每个节点数据一致，由多个节点对外提供服务。这里只用单台。

**confd：**管理本地应用配置文件，使用etcd或consul存储的数据渲染模板，还支持redis、zookeeper等。

confd有一个watch功能，通过HTTP API定期监测对应的etcd中目录变化，获取最新的Value，然后渲染模板

**Nginx:**  Nginx是一款轻量级的Web服务器/反向代理服务器以及电子邮件代理服务器，并在一个BSD-like协议下发行。由俄罗斯的程序设计师lgor Sysoev所开发，供俄国大型的入口网站及搜索引擎Rambler使用。其特点是占有内存少，并发能力强，事实上nginx的并发能力确实在同类型的网页服务器中表现较好。

#### 三．软件部署

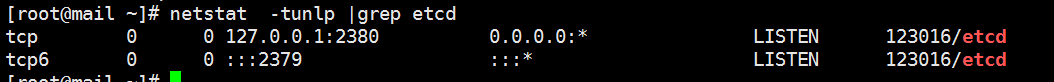
环境说明: 建议使用 Cento7.X X64

1. **安装etcd(这里安装的单机,集群环境根据自己的需求选取)**

# yum install etcd -y

# sed -i 's/localhost/0.0.0.0/g' /etc/etcd/etcd.conf **#配置监听地址**

# systemctl start etcd && systemctl enable etcd **#启动服务设置开机动**



1. **安装nginx**

#cd /usr/local/src

#wget [http://nginx.org/download/nginx-1.12.1.tar.gz](http://nginx.org/download/nginx-1.14.0.tar.gz)

#git clone <https://github.com/yaoweibin/nginx_upstream_check_module.git>

#tar -zxvf nginx-1.12.1.tar.gz

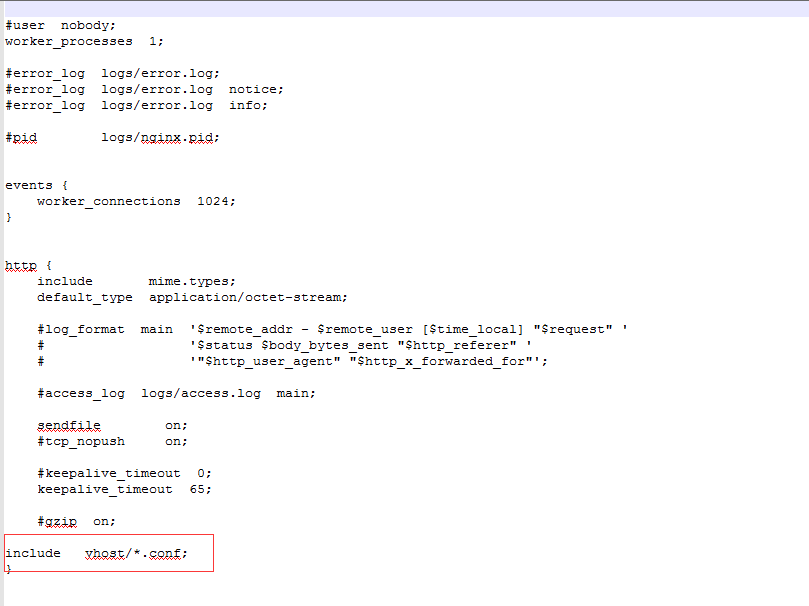
#cd nginx-1.12.1

#patch -p1 </usr/local/src/nginx\_upstream\_check\_module/check\_1.12.1+.patch

#./configure --prefix=/usr/local/nginx --add-module=/usr/local/src/nginx\_upstream\_check\_module/

make && make install

#mkdir /usr/local/nginx/conf/vhost/



Nginx主配置文件修改为这个样子,增加include目录配置

1. **安装confd**

下载地址<https://github.com/kelseyhightower/confd/releases>

下载完毕丢到系统里面

# cp confd /usr/bin/confd

# which confd

/usr/bin/confd

1. **创建配置文件目录**

# mkdir -p /etc/confd/{conf.d,templates}

conf.d    **# 资源模板，下面文件必须以toml后缀**

templates  **# 配置文件模板，下面文件必须以tmpl后缀**

1. **创建confd配置文件**

# vi /etc/confd/conf.d/app01.conf.toml

[template]

src = "app01.conf.tmpl" # 默认在/etc/confd/templates目录下

dest = "/usr/local/nginx/conf/vhost/app01.conf" **#要更新的配置文件**

keys = [

"/Shopping", **#监测的key**

]

reload\_cmd ="/usr/local/nginx/sbin/nginx -s reload" **#最后执行的命令**

1. **创建confd模板**

# vi /etc/confd/templates/app01.conf.tmpl

upstream {{getv "/Shopping/nginx/cluster1/proxy\_name"}} {

{{range getvs "/Shopping/nginx/cluster1/upstream/\*"}}

server {{.}};

{{end}}

check interval=5000 rise=1 fall=5 timeout=4000 type=http;

check\_http\_send "HEAD / HTTP/1.0\r\n\r\n";

check\_http\_expect\_alive http\_2xx http\_3xx;

}

server {

server\_name {{range getvs "/Shopping/nginx/cluster1/server\_name/\*"}} {{.}} {{end}};

location / {

proxy\_pass http://{{getv "/Shopping/nginx/cluster1/proxy\_name"}};

proxy\_redirect off;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

}

location /status {

check\_status;

access\_log off;

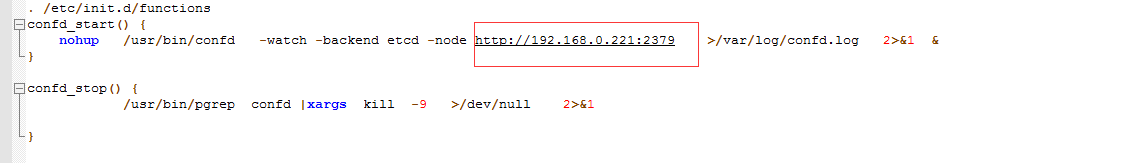
}

}

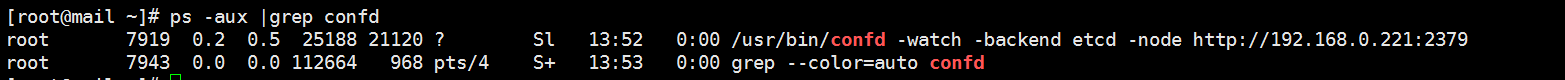
1. **启动confd并设置开机启动**

开机启动脚本会随文档附带

拷贝至/etc/init.d/confd ,只需要更改etcd 的连接地址即可



#/etc/init.d/confd start && chkconfig --add confd && chkconfig confd on



#### 配置平台部署

1. **Github 克隆平台代码安装平台依赖**

# git clone https://github.com/1032231418/Conf\_Web.git

# cd Conf\_Web/ospweb/

#virtualenv env  **#建议创建一个沙盒环境跑该平台**

# source env/bin/activate **#使用沙盒环境**

# pip install -r requirement.txt  **#安装相关软件**

1. **创建数据库并将表刷入数据库**

# vi opsweb/settings.py **#这里数据库信息改为自己的数据库信息**

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.mysql',

'NAME': 'confd',

'HOST': '192.168.8.114',

'USER': 'root',

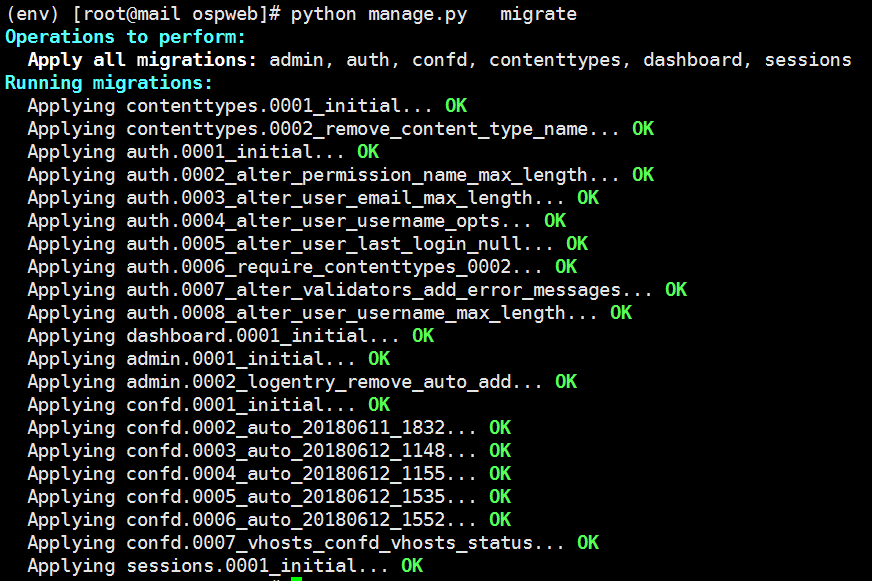
'PASSWORD': '123456',

'PORT': 3306,

}

}

# python manage.py migrate  **#提交迁移文件至数据库,将表刷入数据库**



1. **创建超级管理员账号**

# python manage.py createsuperuser



1. **运行平台**

**# python manage.py runserver 0:8000**

**访问地址就是 <http://ip/8000> 账号密码就是上一步创建的超级管理员账号密码**

1. **登录平台为nginx创建key/value**

**例子: Shopping 平台为例**

**项目创建:**

1. **创建商城项目 /Shopping**
2. **创建商城项目里面的 /Shopping/nginx nginx 服务**
3. **创建nginx 集群目录 /Shopping/nginx/cluster1**
4. **给我们的商城nginx集群1项目创建配置文件**

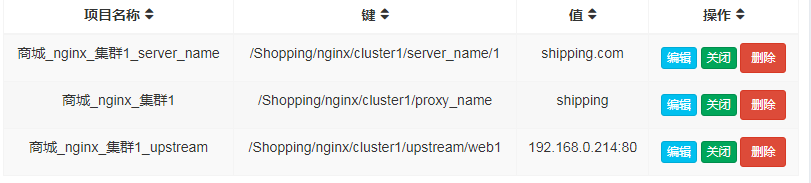
**域名 和 节点名称可能是多个，这里我们需要创建目录 /Shopping/nginx/cluster1/server\_name 和 /Shopping/nginx/cluster1/upstream**

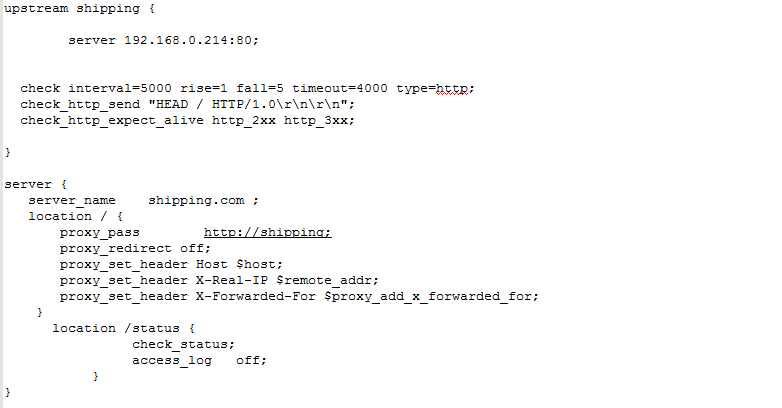


1. **给集群创建节点，和域名**

**配置创建:**

1. **反向代理 /Shopping/nginx/cluster1/proxy\_name**
2. **绑定一个域名 /Shopping/nginx/cluster1/server\_name/1**
3. **创建一个集群节点 /Shopping/nginx/cluster1/upstream/web1**





生成的配置文件

通过hosts 文件我们可以查看节点状态