

服务自动注册/发现

又拍云科技

基于文本的入口发现

text + shell

```

#!/bin/sh
echo 1 > /proc/sys/net/ipv4/ip_forward
RULES=""
wasu-ap1.c.upyun.com#http#fair#8089#192.168.14.121:8089@192.168.14.122:8089

xx=$IFS
STRING="stream {\n"
XX="\tfair;\n"

rm -rf /opt/nginx/conf/conf.d/*.conf
for rules in $RULES;do
    IFS=#;read -r name protocol weight port real <<< "$rules"
    IFS=$xx
    if [ $protocol != "http" ];then
        STRING+="#####\n\tupstream $name {\n"
        STRING_HTTP=
    fi
    if [ $protocol = "http" ];then
        STRING_HTTP="upstream $name {\n"
    fi

    [ $weight = "ip_hash" ] && XX="\tip_hash;\n"
    [ $weight = "least_conn" ] && XX="\tleast_conn;\n"
    [ $protocol = "http" ] && STRING_HTTP+="$XX"

    for i in $(echo $real|tr '@' ' ');do
        [ $protocol != "http" ] && STRING+="\t\tserver $i weight=5 max_fails=2 fail_timeout=3s;\n"
        [ $protocol = "http" ] && STRING_HTTP+="\tserver $i weight=5 max_fails=2 fail_timeout=3s;\n"
    done
    [ $protocol != "http" ] && STRING+="\t}\n"
    [ $protocol = "http" ] && STRING_HTTP+="}\n\n"

    if [ $protocol = "udp" ];then
        STRING+="\tserver {\n\t\tlisten\t$port udp;\n\t\tproxy_pass\t$name;\n\t\tproxy_timeout\t3s;\n\t\telit [ $protocol = "http" ];then
        STRING_HTTP+="server {\n\t\tlisten\t$port;\n\t\tserver_name $name;\n\t\terror_log\t/opt/nginx/logs.
proxy_set_header Host \$host;\n\t\tproxy_set_header X-Real-IP \$remote_addr;\n\t\tproxy_set_header X-Forward
else
        STRING+="\tserver {\n\t\tlisten\t$port;\n\t\tproxy_pass\t$name;\n\t\tproxy_timeout\t3s;\n\t\t\
    fi
    if [ ! -z "$STRING_HTTP" ];then
        echo -e "$STRING_HTTP\n}" > /opt/nginx/conf/conf.d/$name.conf
    fi
done
echo -e "$STRING\n}" > /opt/nginx/conf/stream.d/stream.conf

```

服务名 - 协议 - 类型 - 均衡方式 - 外端口 - 后端服务池

proxy_rules.sh

HTTP 服务

TCP/UDP 服务

利用inotify-tools

监控文件变化触发事件

```
#!/bin/sh
CMD="/usr/bin/inotifywait"
DEST_DIR="/root/"
OPTS="close_write,delete,create,attrib"
nginx_load() {
    /opt/nginx/sbin/nginx -t && /opt/nginx/sbin/nginx -s reload
}

if [ -z $1 ];then
    $CMD -mrq -e $OPTS --fromfile /root/xxx --format "%w %e %f"|while read DIR EVENT FILE;do
        echo $DIR $EVENT $FILE
        /root/proxy_rules.sh && nginx_load
    done
elif [ $1 = "stop" ];then
    PID=$(ps -ef|grep $CMD|grep -v grep|awk '{print $2}')
    kill -9 $PID
    exit 0
fi
```

需要监听的事件类型

触发的事件脚本

基于API的服务发现

DR CoN: Scalable Architecture using
Docker, Registrator, Consul, Consul Template and Nginx

构建Consul三节点集群

IPADDR="192.168.13.167"

docker run -d --name consul -h node1 \

-p \$IPADDR:8300:8300 \

-p \$IPADDR:8301:8301 \

-p \$IPADDR:8301:8301/udp \

-p \$IPADDR:8302:8302 \

-p \$IPADDR:8302:8302/udp \

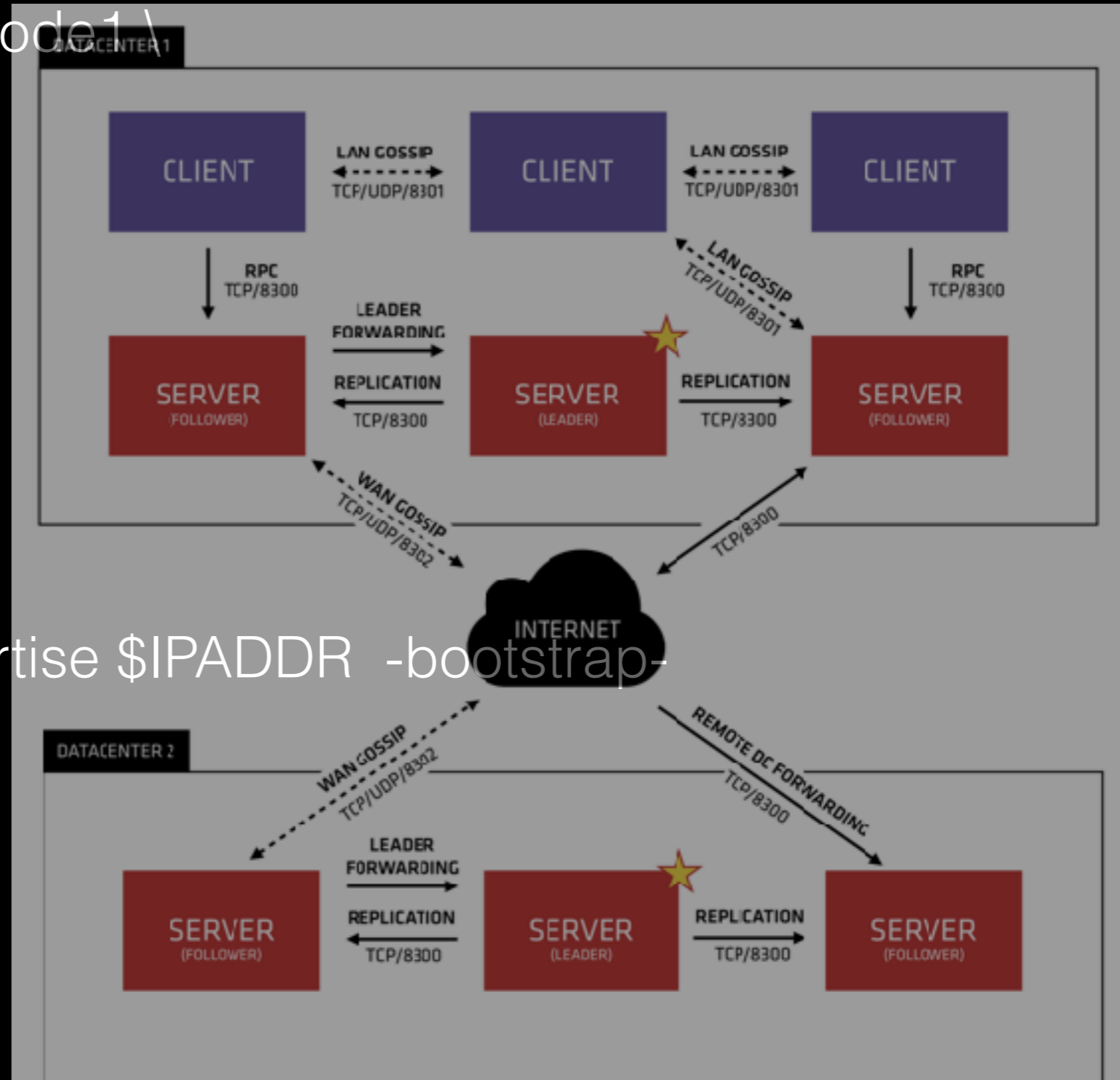
-p \$IPADDR:8400:8400 \

-p \$IPADDR:8500:8500 \

-p \$IPADDR:8600:53/udp \

progrum/consul -server -advertise \$IPADDR -bootstrap-

expect 3



部署registrator自动注册

```
IPADDR="192.168.13.167"
```

```
docker run -itd \  
-v /var/run/docker.sock:/tmp/docker.sock \  
-h $IPADDR gliderlabs/registrator \  
consul://$IPADDR:8500
```

每个consul agent的节点都要运行，打通docker api和consul 8500之间联系

启动Docker微服务容器

```
docker run -d --name service2
```

```
-P jlordiales/python-micro-service
```

Service Name	Status
consul	3 passing
consul-53	3 passing
consul-8300	3 passing
consul-8301	6 passing
consul-8302	6 passing
consul-8400	3 passing
consul-8500	3 passing
nginx-consul-80	2 passing
python-micro-service	3 passing
simple	1 passing

Node Name	IP Address	Health Status
node1	192.168.13.167	serfHealth
node2	192.168.13.168	serfHealth
node3	192.168.13.169	serfHealth

模板配置 Consul-template

```
{{range service "python-micro-service"}}
upstream {{.Name}} {
  ip_hash;
  server {{.Address}}:{{.Port}} max_fails=3 fail_timeout=60 weight=1;
}

server {
  listen 80 default_server;
  charset utf-8;
  location / {
    proxy_pass http://{{.Name}};
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
  }
}
{{end}}
```

语法测试:

```
consul-template -consul-addr 192.168.13.169:8500 \
  -template /root/test.ctmpl:/tmp/consul.result -dry -once
```

运行consul-template的nginx容器

```
docker run -p 8080:80 -d --name nginx \  
  --volume /root/test.ctmpl:/templates/service.ctmpl \  
  --link consul:consul jlordiales/nginx-consul
```

<http://192.168.13.167:8080/>

