

NGINX从入门到精通进阶系列培训

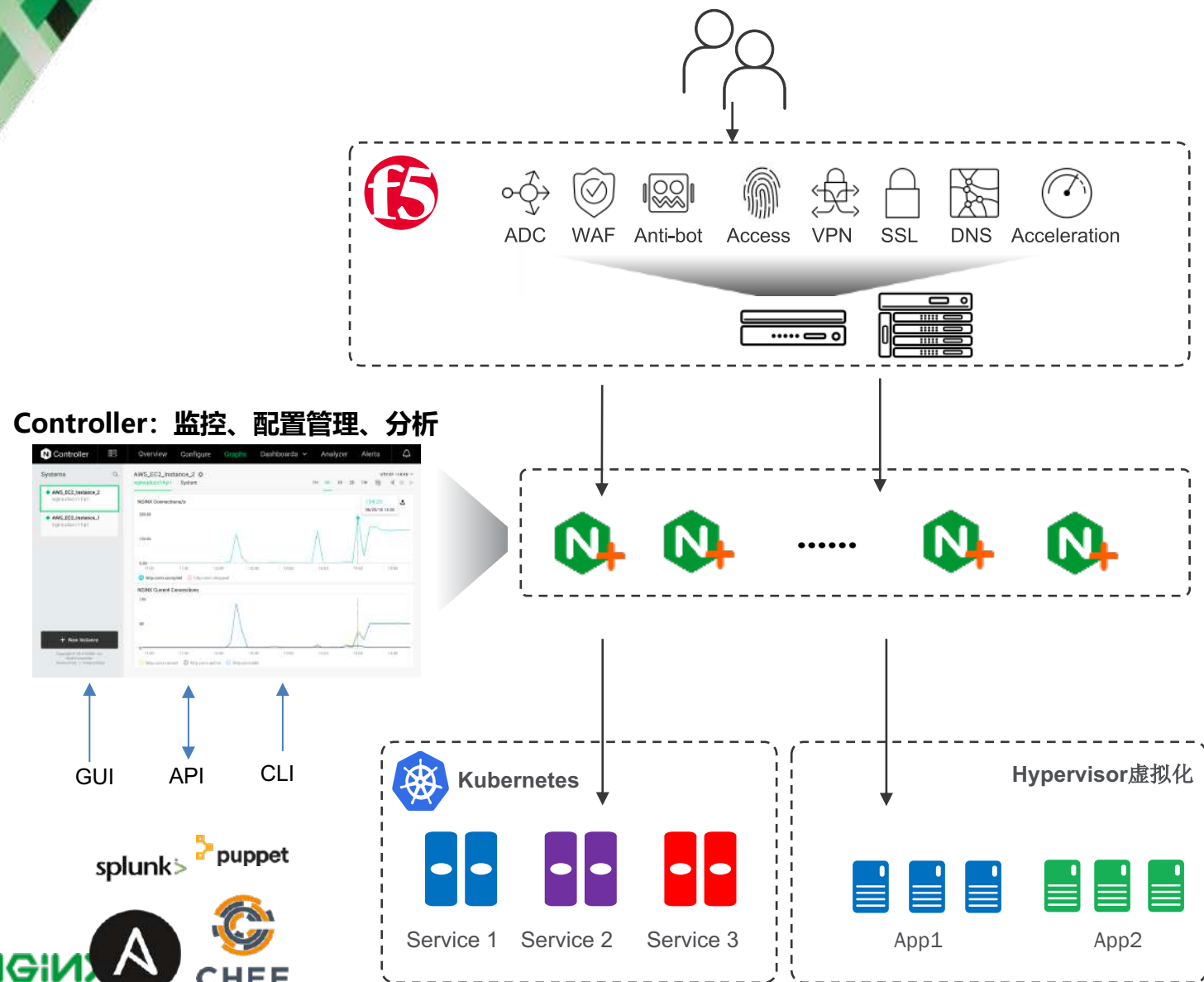
实践篇：灰度、A/B、限流等



廖健雄 j.liao@f5.com



NGINX常见功能场景



反向代理/Web Server

URL分流

SSL证书卸载

Cache

服务发现

A/B测试, 灰度发布

限流

应急流量切换

集群间配置/状态同步



NGINX高级指令简介 - split_clients

```
split_clients $remote_addr $upstream {  
    5% appversion2;  
    *   appversion1;  
}  
  
upstream appversion1 {  
    # ...  
}  
  
upstream appversion2 {  
    # ...  
}  
  
server {  
    listen 80;  
    location / {  
        proxy_pass http://$upstream;  
    }  
}
```

- ngx_http_split_clients_module 模块允许变量匹配用于A/B 测试等场景。
- hash的机制

NGINX高级指令简介 - map

```
map variable-from-request variable-to-set {  
    regex1 value-to-set-if-match;  
    regex2 value-to-set-if-match;  
    #...  
    regexN value-to-set-if-match;  
    default value-to-set-if-no-match;  
}
```

```
map $uri $isphp {  
    ~\.php$ 1;  
    default 0;  
}
```

```
map $http_user_agent $mobile {  
    default 0;  
    "~Opera Mini" 1;  
}
```

- NGINX map 指令使用正则表达式进行变量的配置

NGINX高级指令简介 - keyval

```
http {
    keyval_zone zone=blacklist:64k timeout=2h sync
state=/etc/nginx/state_files/blacklist.json;
    keyval $remote_addr $blacklist_status zone=blacklist;
    ...
    server {
        ...
        location / {
            if ($blacklist_status) {
                proxy_pass http://blacklist_webserver;
            }
        }
    }

    http://nginx_ip/api/3/http/keyvals/blacklist
    -----
    {
        "10.1.1.1":1
    }
}
```

- NGINX PLUS 内置的Key-value 对，可以被API以及njs进行管理配置

一个简单的灰度发布案例



张总提了个需求，要根据浏览器类型来做灰度发布。

老大，没问题



这么快，小伙子可以哦！

老大，已经实现了



张总又提了个需求，再帮他多加一个灰度策略

好的，我试试



张总说还要能生产上快速回退的方案

这个。。走变更流程么？



走流程哪能快得起来？



张总的需求老变化，你要不想办法给他个自服务的方式吧。

老大你给我3个月时间，再给我3-5个开发



。。。你自己想一下办法



```
http {
    upstream webserver {
        # ...
    }
    upstream grayserver {
        # ...
    }
    map $http_user_agent $isGray {
        "~Opera Mini" 1;
        default 0;
    }
    server {
        location / {
            if ($isGray) {
                proxy_pass http://grayserver;
            }
            proxy_pass http://webserver;
        }
    }
}
```

NGINX PLUS 指令/模块增强

新增商业模块	实现功能
<u>ngx_http_api_module</u>	API 接口
<u>ngx_http_auth_jwt_module</u>	JWT 认证
<u>ngx_http_f4f_module</u>	Adobe HTTP Dynamic Streaming (HDS)支持
<u>ngx_http_hls_module</u>	HTTP Live Streaming (HLS) server的支持
<u>ngx_http_keyval_module</u>	http KV存储
<u>ngx_http_session_log_module</u>	日志记录会话（即多个HTTP请求的聚合），而不是单个HTTP请求。
<u>ngx_http_status_module</u>	收集运行状态信息，90+ metric
<u>ngx_http_upstream_conf_module</u>	API 配置upstream模块
<u>ngx_http_upstream_hc_module</u>	主动健康检查
<u>ngx_stream_keyval_module</u>	Stream KV 存储
<u>ngx_stream_upstream_hc_module</u>	主动健康检查
<u>ngx_stream_zone_sync_module</u>	集群间状态同步

开源模块增强	新增指令实现功能
<u>ngx_http_core_module</u>	报错友好页面可以自定义
<u>ngx_http_fastcgi_module</u>	fastcgi_cache的操控
<u>ngx_http_limit_req_module</u>	Sync 集群间同步指令
<u>ngx_http_mp4_module</u>	mp4_limit_rate限流指令
<u>ngx_http_proxy_module</u>	Proxy cache的操控
<u>ngx_http_scgi_module</u>	Scgi_cache的操控
<u>ngx_http_upstream_module</u>	主动健康检查的配置
<u>ngx_http_uwsgi_module</u>	Uwsgi cache的操控
<u>ngx_stream_upstream_module</u>	主动健康检查的配置

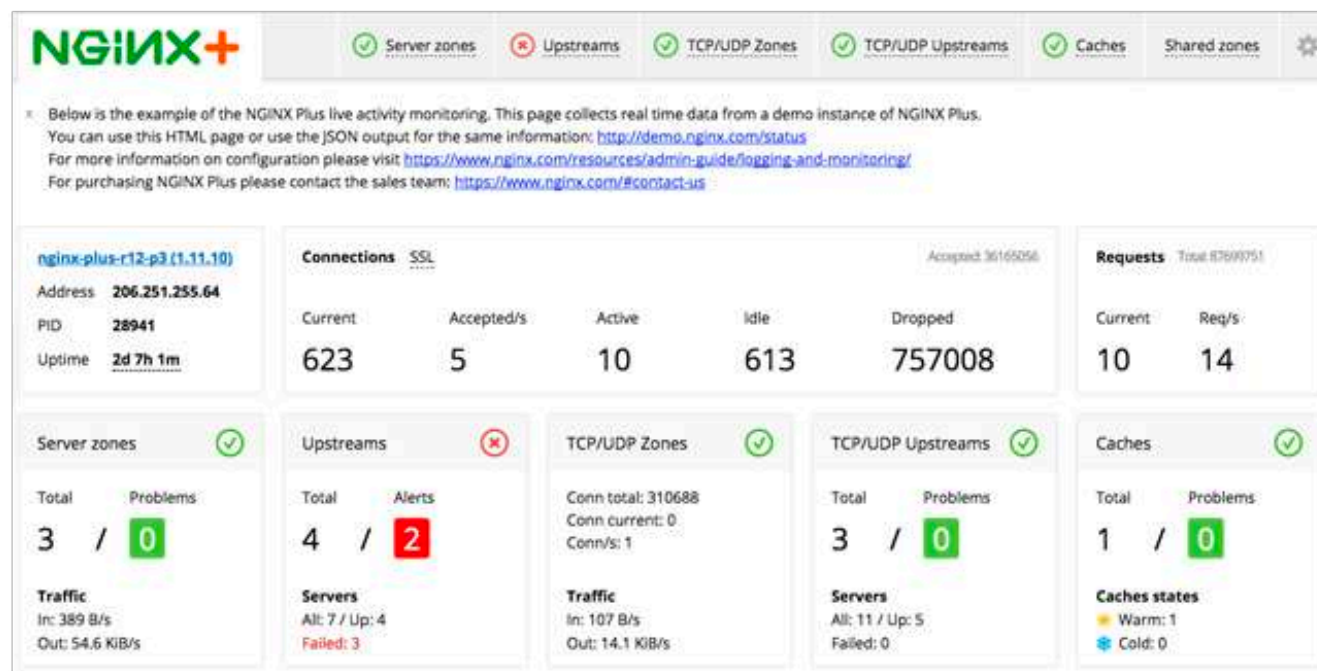
<https://www.nginx.com/products/nginx/#compare-versions>

NGINX PLUS 监控

- 内建实时的图形dashboard
- 高达100多个实时性能参数指标
- 支持以JSON和HTML输出格式集成到客户监控平台

← → ↻ ⓘ 不安全 | 10.1.10.143/stub_status

```
Active connections: 2
server accepts handled requests
23438 23438 82520
Reading: 0 Writing: 1 Waiting: 1
```



NGINX PLUS API接口

- 管理upstream servers, key-value 存储, 以及获取实时的运行状态信息。
- 动态修改配置, 无需进程reload。

server { //可单独创建一个server用于api、dashboard的访问, 避免和业务server造成冲突。

```
listen 8081;
```

```
location /api {  
    api write=on;  
}
```

```
location = /dashboard.html {  
    root /usr/share/nginx/html;
```

```
}
```

```
location /swagger-ui {  
    root /usr/share/nginx/html;  
}
```



NGINX PLUS官方镜像仓库

^ Dynamic Modules

Dynamic Modules

Brotli

Cookie-Flag

Encrypted-Session

GeoIP

GeoIP2

Headers-More

HTTP Substitutions Filter

Image-Filter

Lua

njs Scripting Language

NGINX WAF

OpenTracing

Phusion Passenger Open Source

Perl

Prometheus-njs

RTMP

Set-Misc

XSLT

安装NGINX PLUS

```
sudo yum install nginx-plus
```

升级NGINX PLUS

```
sudo yum upgrade nginx-plus
```

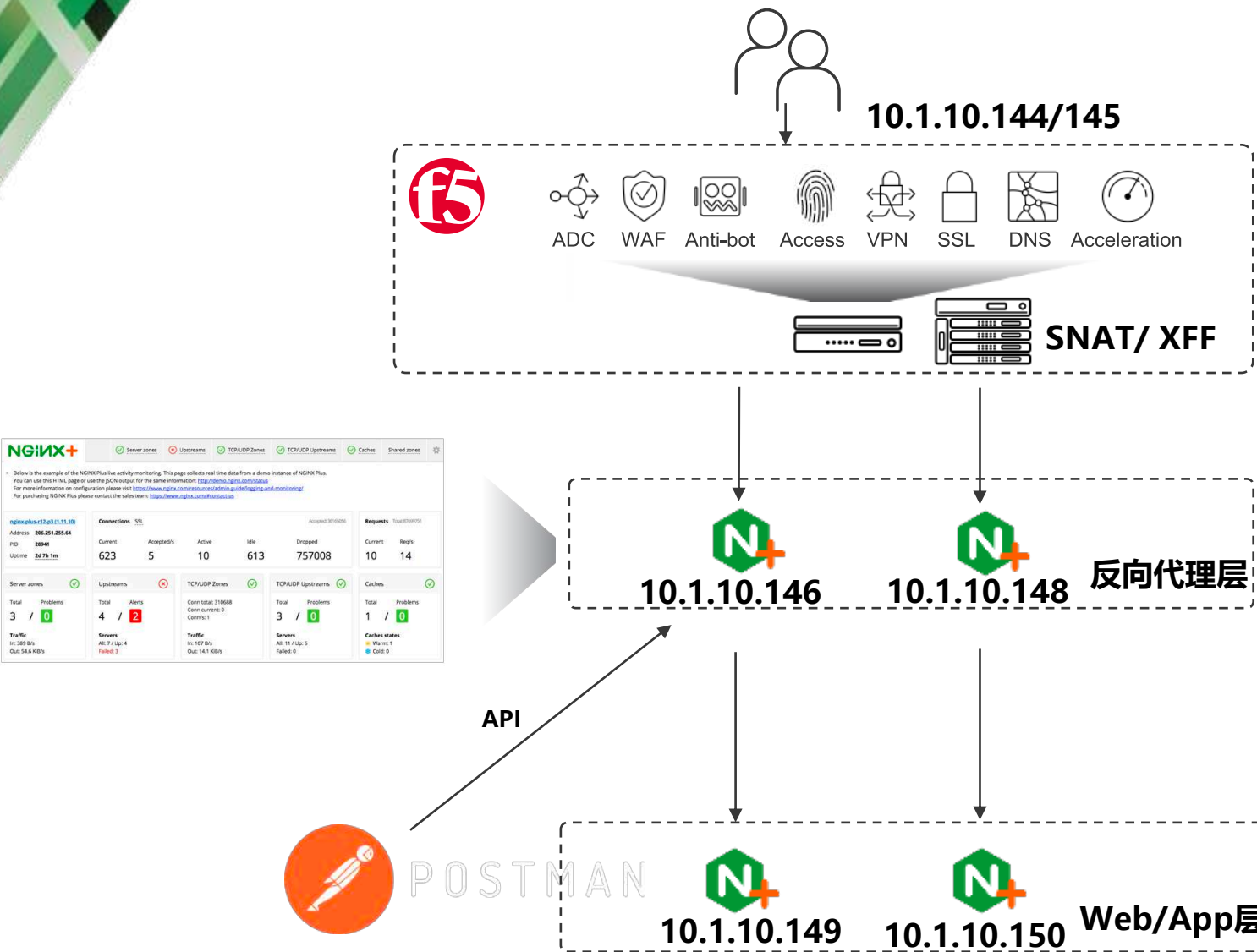
安装动态模块

```
sudo yum install nginx-plus-module-image-filter
```

加载动态模块

```
load_module modules/nginx_http_image_filter_module.so;
```

实践篇Demo架构图



反向代理/Web Server

URL分流

SSL证书卸载

Cache

服务发现

A/B测试, 灰度发布

限流

应急流量切换

集群间状态同步

关注我们

F5 官方微信
(新闻, 技术文章)



F5社区
(答疑, 吐槽, 分享, 互动)



加入F5社区：关注“F5社区”微信公众号，注册成为社员，随时参加meet up活动，代码共享，讨论，答疑等。只要你有想法，有创造，那么就快来大展身手吧，让我们在社区里尽情分享，交流，吐槽和互动，在这个自由的国度里，发现闪亮的自己。让我们一起来见证“一群有才能的人在一起做有梦想的事！”

NGINX技术群



操作步骤：

1. 扫描二维码并在“入群信息”栏填写姓名
2. 点击下方“我要入群”
3. 长按识别二维码进入群聊



Thank You