

Linux高级运维

NSD OPERATION

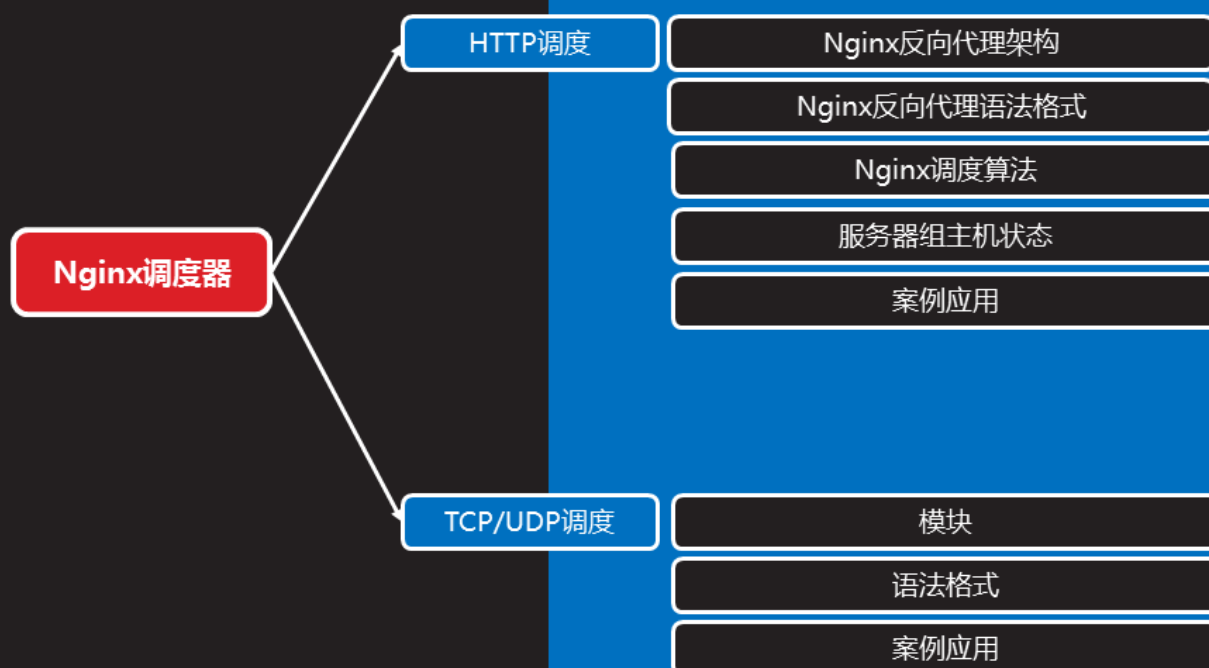
DAY03

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	Nginx调度器
	10:30 ~ 11:20	
	11:30 ~ 12:20	
下午	14:00 ~ 14:50	Nginx常见问题
	15:00 ~ 15:50	
	16:00 ~ 16:50	
	17:00 ~ 17:30	总结和答疑



Nginx调度器

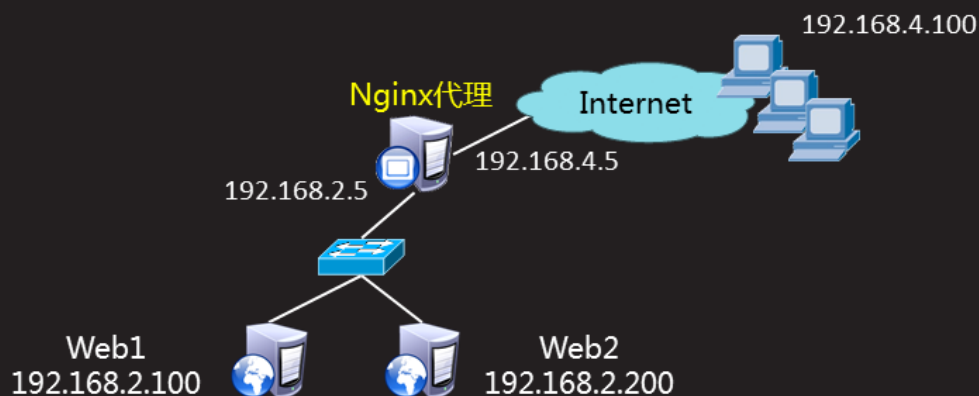


HTTP调度

Nginx反向代理架构

- 实验拓扑

知识讲解



Nginx反向代理语法格式

知识讲解

```
http {  
    upstream sergrp {  
        server 192.168.8.5:80;  
        server 192.168.8.6:80;  
    }  
    server {  
        listen 80;  
        server_name www.tarena.com;  
        location / {  
            .....  
            proxy_pass http://sergrp;  
        }  
    }  
}
```

//定义源服务器组

//Web主机名

//调用服务组



Nginx调度算法

知识讲解

- nginx目前支持的调度算法
 - 轮询（默认的）：逐一循环调度
 - weight：指定轮询几率，权重值和访问比率成正比
 - ip_hash：根据客户端IP分配固定的后端服务器



服务器组主机状态

知识讲解

- 状态类型
 - down : 表示当前server暂时不参与负载
 - max_fails : 允许请求失败的次数 (默认为1)
 - fail_timeout : max_fails次失败后, 暂停提供服务的时间



Web案例应用

知识讲解

```
upstream sergrp {  
    #ip_hash;  
    #server 192.168.8.5:80 weight=2;  
    server 192.168.8.5:80 down;  
    server 192.168.8.4:80;  
    server 192.168.8.6:80 backup;  
    server 192.168.8.3:80 max_fails=2 fail_timeout=30;  
}
```



案例1：Nginx反向代理

课堂练习

- 任务要求
 - 改用Nginx实现Web反向代理
 - 使用加权的轮询调度算法，调度后端两台Web服务器
 - 后端服务器最大失败错误为2次，宕机后暂停30秒服务
 - 客户端检测多个源站的效果

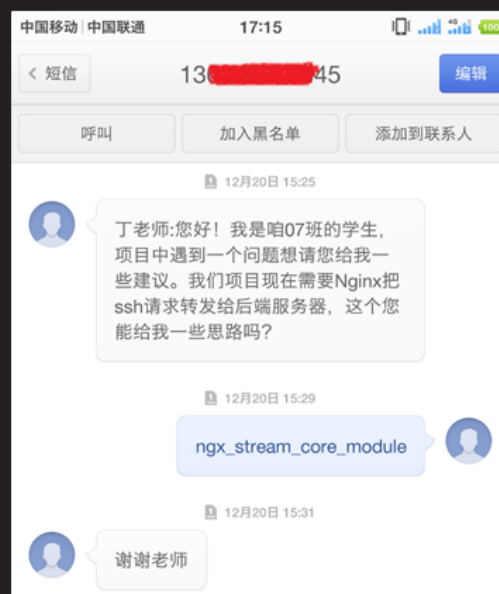


TCP/UDP调度

模块

- ngx_stream_core_module模块
- 使用--with-stream开启该模块
- 注意：
 - nginx从1.9版本才开始支持该功能

知识讲解



语法格式

```
stream {  
    upstream backend {  
        server backend1.example.com:12345 weight=5;  
        server 127.0.0.1:22 max_fails=3 fail_timeout=30s;  
    }  
    server {  
        listen 12345;  
        proxy_connect_timeout 1s;  
        proxy_timeout 3s;  
        proxy_pass backend;  
    }  
}  
  
http{  
    ... ..  
}
```

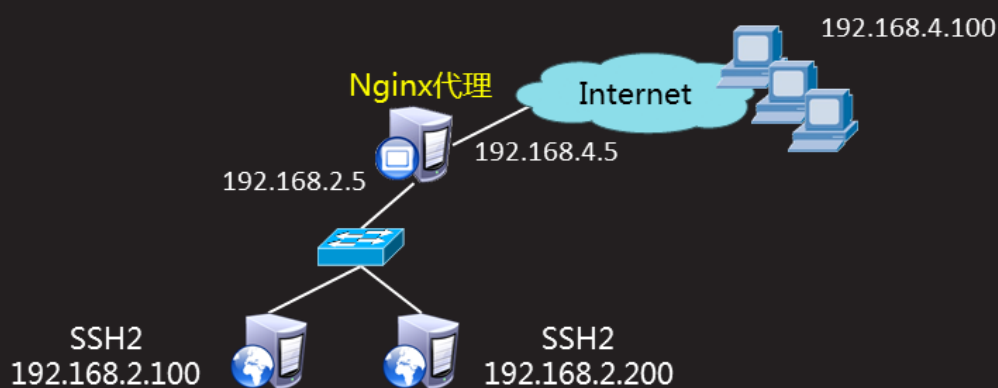
知识讲解



SSH案例应用

- 使用Nginx调度后端SSH服务

知识讲解



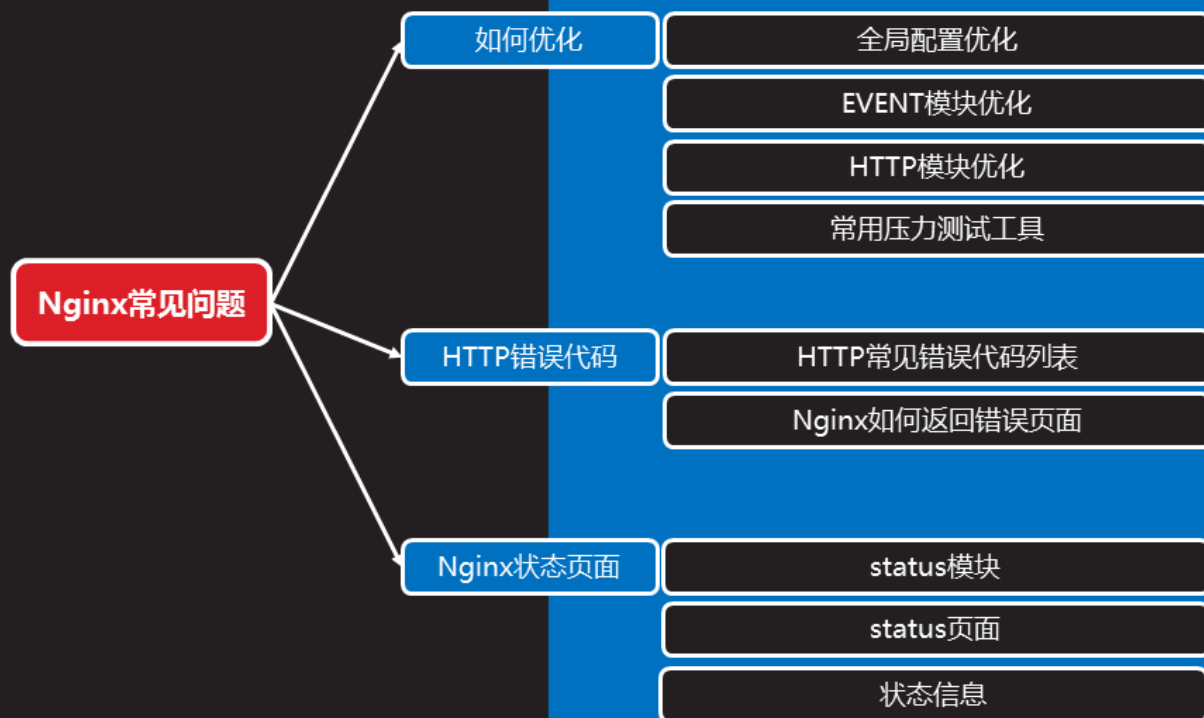
案例2：Nginx的TCP/UDP调度器

- 任务要求
 - Nginx实现SSH代理
 - 使用加权的轮询调度算法，调度后端两台SSH服务器

课堂练习



Nginx常见问题



如何优化

全局配置优化

- 调整进程数量

知识讲解

```
#user nobody;  
worker_processes 2;  
error_log /var/log/nginx/error_log info;  
# [ debug | info | notice | warn | error | crit ]
```

//与CPU核心数量一致
//定义日志级别



EVENT模块优化

- $\text{max_clients} = \text{worker_processes} * \text{worker_connections}$
- 注意修改系统ulimit限制/etc/security/limits.conf

知识讲解

```
events {  
    worker_connections 10000;  
    use epoll;  
}
```

//每个worker最大并发连接数



HTTP模块优化

知识讲解

```
http {
    keepalive_timeout 10;           //保持连接的超时时间
    gzip on;
    gzip_min_length 1000;
    gzip_comp_level 4;
    gzip_types text/plain text/css application/json application/x-javascript text/xml
    application/xml application/xml+rss text/javascript;
    client_header_buffer_size 1k;   //默认请求包头信息的缓存
    large_client_header_buffers 4 4k; //大请求包头部信息的缓存个数与容量
    //先根据client_header_buffer分配，如果不够，再根据large值分配
}
```



HTTP模块优化（续1）

- 如果需要处理大量静态文件，需要保持这些文件句柄为打开状态，避免后续再次打开

知识讲解

```
http {
    open_file_cache      max=2000 inactive=20s;
        open_file_cache_valid 60s;
        open_file_cache_min_uses 5;
        open_file_cache_errors off;
    //设置服务器最大缓存2000个文件句柄，关闭20秒内无请求的文件句柄
    //文件句柄的有效时间是60秒，60秒后过期
    //只有访问次数超过5次会被缓存
}
```



HTTP模块优化（续2）

- 客户端浏览器缓存数据

知识讲解

```
http {  
    location ~* \.(jpg|jpeg|gif|png|css|js|ico|xml)$ {  
        access_log      off;  
        expires          30d;  
    }  
}
```



常用压力测试工具

- ab
 - ab -c 并发数 -n 总请求数 URL
- 其他常见压力测试软件
 - http_load、webbench、siege

知识讲解



HTTP错误代码

HTTP常见错误代码列表

返回码	描述
200	一切正常
400	请求语法错误
401	访问被拒绝（账户或密码错误）
403	资源不可用，通常由于服务器上文件或目录的权限设置导致
403	禁止访问：客户端的IP地址被拒绝
404	无法找到指定位置的资源（Not Found）
414	请求URI头部太长
500	服务器内部错误
502	服务器作为网关或者代理时，为了完成请求访问下一个服务器，但该服务器返回了非法的应答（Bad Gateway）

Nginx如何返回错误页面

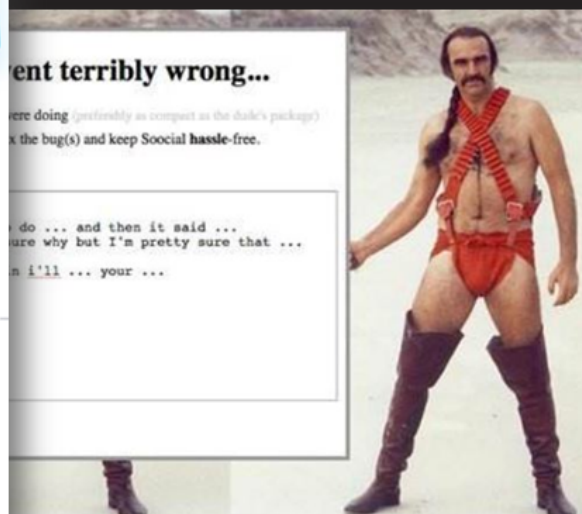
- 遇到访问错误时，处理错误代码你还有别的选择

知识讲解



Nginx如何返回错误页面（续1）

知识讲解



Nginx如何返回错误页面（续2）

知识讲解

```
http {  
    fastcgi_intercept_errors on;           //错误页面重定向  
    server {  
        error_page 404 /40x.html;         //自定义错误页面  
            location = /40x.html {  
                root html;  
            }  
        error_page 500 502 503 504 /50x.html;  
            location = /50x.html {  
                root html;  
            }  
    }  
}
```



Nginx状态页面

status模块

知识讲解

- status
 - --with-http_stub_status_module开启模块功能
 - 可以查看Nginx连接数等信息

```
[root@nginx-1.12.2]# ./configure --with-http_stub_status_module ... ..
```

注意：这里省略了其他选项...



status页面

知识讲解

- 修改配置文件nginx.conf

```
[root@~]# cat /usr/local/nginx/conf/nginx.conf
```

```
... ..  
location /status {  
    stub_status on;  
}  
... ..
```



status页面 (续1)

- 浏览器访问状态页面

知识讲解

```
[root@~]# curl http://192.168.4.5/status
Active connections: 1
server accepts handled requests
10 10 3
Reading: 0 Writing: 1 Waiting: 0
```



状态信息

- Active connections
 - 当前活动的连接数量
- Accepts
 - 已经接受客户端的连接总数量
- Handled
 - 已经处理客户端的连接总数量 (一般与accepts一致, 除非服务器限制了连接数量)
- Requests
 - 客户端发送的请求数量

知识讲解



状态信息（续1）

知识讲解

- Reading
 - 当前服务器正在读取客户端请求头的数量
- Writing
 - 当前服务器正在写响应信息的数量
- Waiting
 - 当前多少客户端在等待服务器的响应



案例3：Nginx常见问题处理

课堂练习

- 浏览网站时提示 “Too many open files” 如何解决
- 如何解决客户端访问头部信息过长的问题
- 如何让客户端浏览器缓存数据
- 如何自定义返回给客户端的404错误页面
- 如何查看服务器状态信息



总结和答疑

总结和答疑

调度器故障

问题现象

原因分析

调度器故障

问题现象

- 故障错误信息

[root@svr5 nginx-1.8.0]# firefox http://192.168.2.100
访问调度器后失败，提示50X错误或An error occurred

知识讲解



原因分析

- 分析故障
 - 查看nginx日志，看看本机失败还是连接后端主机失败
- 分析故障原因
 - Connection refused, while connecting to upstream
 - 原因：提示连接后端服务器被拒绝

知识讲解



