

Linux服务器构建与运维管理

从基础到实战 (基于 openEuler)

第5章：使用Nginx实现代理服务

阮晓龙

13938213680 / ruanxiaolong@hactcm.edu.cn

<https://internet.hactcm.edu.cn>

<http://www.51xueweb.cn>

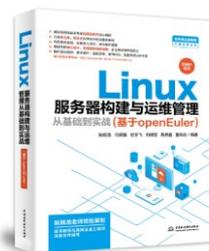
河南中医药大学信息技术学院互联网技术教学团队
河南中医药大学医疗健康信息技术研究所

2024.10

1

提纲

- 代理服务
- 了解Nginx
 - Nginx Open Source
 - F5 Nginx Plus
- 使用Nginx实现代理服务
 - 多种方式发布业务
- 使用Nginx实现负载均衡
 - 负载均衡与容灾
- 实现Nginx可视化管理
 - 运行监控
 - 可视化管理 (Web Console)
 - 集中管理



2

1. 代理服务

1.1 代理服务器

- 代理服务器是介于客户端浏览器和服务端Web服务器之间的一台服务器。
- 当用户通过代理服务器访问网站时，浏览器不是直接到Web服务器去取回网页，而是向代理服务器发出请求，由代理服务器负责取回所请求的网页内容并传回用户浏览器。
- 使用代理服务器进行网站发布也是最为常见的提升网站安全的重要措施之一。



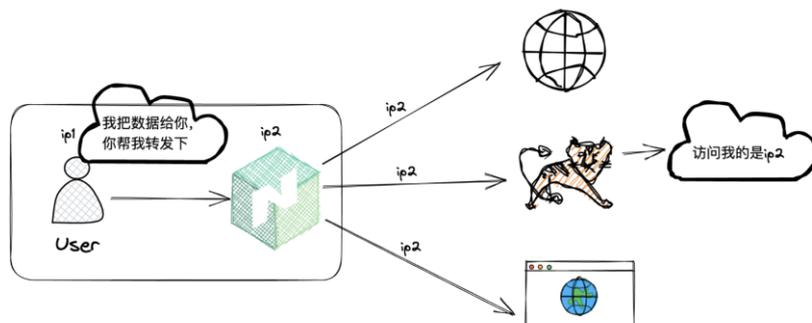
1. 代理服务

1.2 代理服务

- 代理服务可以实现互联网与局域网之间的通信，分为正向代理和反向代理两种。
 - 正向代理
 - 正向代理服务器是位于客户端与互联网上的网站服务器之间的服务器。
 - 当客户端无法访问外部资源时，为了从互联网上的网站服务器获取内容，客户端可发送请求到正向代理服务器，然后正向代理服务器从互联网上的网站服务器中获取内容并返回给客户端。
 - 正向代理的典型应用就是为内部客户端访问外网提供方便，比如企业网/校园网内部用户通过代理访问外部网站等。
 - 使用正向代理服务器时，客户端必须专门配置正向代理服务器的信息，如在浏览器中配置代理服务器等。代理服务器能够使用缓存来缓解网站服务器的负载情况，提升响应速度。



正向代理的基本工作流程



Nginx快速上手: https://blog.csdn.net/qq_24950043/category_12521098.html

5

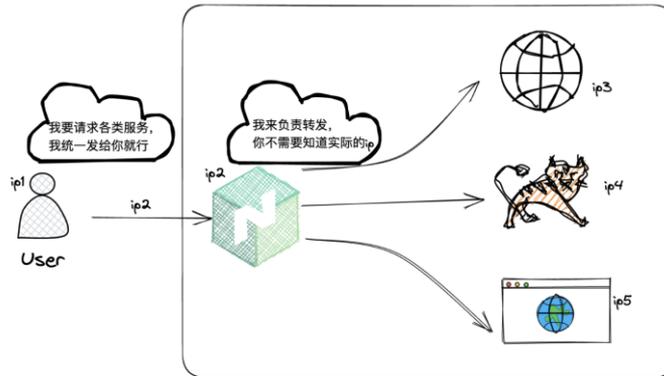
1. 代理服务

1.2 代理服务

- 代理服务可以实现互联网与局域网之间的通信，分为正向代理和反向代理两种。
 - 反向代理
 - 反向代理与正向代理相反，在客户端看来它就像是一个普通的网站服务器，客户端不需做任何配置。
 - 客户端发送请求到代理服务器，代理服务器决定将这些请求转发到何处。
 - 反向代理的主要作用如下。
 - 隐藏服务器真实IP，客户端只能看到代理服务器地址。
 - 实现业务负载均衡，代理服务器可根据网站服务器的负载情况，将客户端请求分发到不同的网站服务器。
 - 提高业务访问速度，代理服务器提供缓存服务、提高网站等业务的访问速度。
 - 提供安全保障，代理服务器可作为应用层防火墙，为内部网站提供安全防护。

6

反向代理的基本工作流程



Nginx快速上手: https://blog.csdn.net/qq_24950043/category_12521098.html

7

1. 代理服务

1.2 代理服务

□ 代理服务可以实现互联网与局域网之间的通信，分为正向代理和反向代理两种。

■ 反向代理

□ 反向代理服务器可以**提高Web服务器的交付安全**。

- 采用反向代理的模式进行应用部署时，真实的数据服务器与用户之间由反向代理服务器分割，将真实的数据服务器隐藏。
- 当用户访问具体的业务时，需要先通过反向代理服务器，同样的当业务遭受攻击时，也会通过反向代理服务器。反向代理服务器可通过地址保护、访问范围限制等方式进行攻击拦截。

□ 反向代理服务器可以**实现负载均衡**。

- 反向代理服务器可以将请求转发给内部的Web服务器，基于此原理可以让代理服务器将请求均匀的转发给多台的Web服务器上，达到负载均衡。
- 使用反向代理进行负载均衡的实现时可以将负载均衡与代理服务器的高速缓存技术结合起来，提高性能和安全性。

□ 反向代理服务器可以**提升信息安全**。

- 反向代理服务器在访问业务时进行了请求转发，在请求转发的过程中，可以配置过滤规则，利用反向代理服务器有效提升信息安全。
- 通过设置规则实现防范DDOS攻击、资源盗链、SQL注入等威胁。

8

1. 代理服务

1.2 代理服务

- 代理服务可以实现互联网与局域网之间的通信，分为正向代理和反向代理两种。
 - **正向代理与反向代理的区别**
 - 在正向代理与反向代理的模式中，虽然代理服务器所处位置都是在客户端与网站服务器之间，都是将客户端的请求转发给网站服务器，但两者之间存在一定的差异。
 - 正向代理是客户端的代理，反向代理是服务器的代理。
 - 正向代理一般是为客户端架设的，反向代理一般是为服务器架设的。
 - 正向代理中网站服务器无法获知客户端的真正地址，反向代理中客户端无法获知网站服务器的真正地址。
 - 正向代理主要是解决内部访问外部网络受到限制的问题，反向代理主要是提供更为安全稳定的网站服务。



2. NGINX

2.1 认识NGINX

- Nginx是一个高性能的HTTP和反向代理服务器，也是开源Web服务器软件。
 - 由俄罗斯程序员Igor Sysoev开发的，最初是为了解决在高并发场景下的C10k问题（即一个服务器进程只能处理 10,000 个并发连接）的限制。
 - Nginx以高性能、稳定性、丰富的功能集、简单配置及低资源消耗而被广泛使用。
 - **Nginx为什么能支持大并发？**
 - Nginx使用事件驱动的异步非阻塞I/O模型，默认使用epoll模型，这使得它在处理大量并发连接时表现非常出色。
 - Nginx还支持worker进程，可以有效地分配资源并提高性能。



Technologies > Web Servers

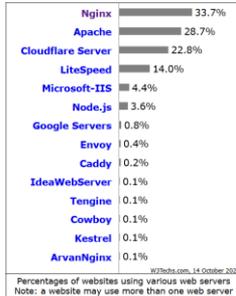
Usage statistics and market shares of web servers

Request an **extensive web servers market report**.

[Learn more](#)

This diagram shows the percentages of websites using various web servers. See [technologies overview](#) for explanations on the methodologies used in the surveys. Our reports are updated daily.

How to read the diagram:
Nginx is used by 33.7% of all the websites whose web server we know.



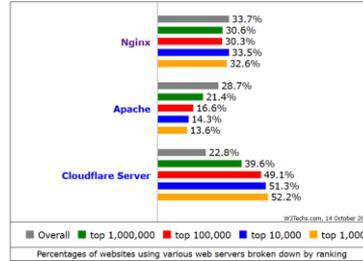
Usage of web servers broken down by ranking

This diagram shows the percentages of websites using various web servers broken down by ranking. Cross-technology reports only include technologies with more than 1% usage to ensure statistical significance of the results. See [technologies overview](#) for explanations on the methodologies used in the surveys.

Detailed statistics in our extensive web servers market report.

[Learn more](#)

How to read the diagram:
Nginx is used by 30.6% of all the websites whose web server we know.
Nginx is used by 30.6% of all the websites whose web server we know and that rank in the top 1,000,000.



https://w3techs.com/technologies/overview/web_server

2002-2003

Igor Sysoev enhances Apache web server

NGINX development begins

Pilot deployments start

2014-2018

Series B&C funding and international expansion

NGINX becomes a multi-product organization

HTTP/2 support added

2004-2008

First open source release of NGINX

Early adopters Community formed

Adopted by cloud-native startups

2019-2023

FS, Inc. acquires NGINX

Programmability features expand

HTTP/3 support added

2009-2013

NGINX 1.0.0 released!

Offices open in Moscow and San Francisco

First commercial product released

2024

NGINX Open Source moves to GitHub

NGINX One released

We celebrate 20 years!

<https://blog.nginx.org/blog/celebrating-20-years-of-nginx>



13

14

nginx ("engine x") is an HTTP web server, reverse proxy, content cache, load balancer, TCP/UDP proxy server, and mail proxy server. Originally written by Igor Sysoev and distributed under the **2-clause BSD License**. **Enterprise distributions**, commercial support and training are available from **F5, Inc.**

↙ ↘

nginx: open Source
<https://nginx.org/>

nginx: enterprise
<https://www.nginx-cn.net/>

 **NGINX**
Part of F5



14

2. NGINX

2.2 NGINX的特性

- Nginx的主要特性
 - **Basic HTTP server features**
 - 基本HTTP服务器功能
 - Other HTTP server features
 - 其他HTTP服务器功能
 - Mail proxy server features
 - 邮件代理服务器功能
 - TCP/UDP proxy server features
 - TCP/UDP代理服务器功能
 - Architecture and scalability
 - 架构和可扩展性
 - Tested OS and platforms
 - 经过测试的操作系统和平台
- Serving static and index files, autoindexing; open file descriptor cache;
- **Accelerated reverse proxying** with caching; load balancing and fault tolerance;
- **Accelerated support** with caching of FastCGI, uwsgi, SCGI, and memcached servers; load balancing and fault tolerance;
- **Modular architecture.** Filters include **gzipping**, byte ranges, chunked responses, XSLT, SSI, and image transformation filter. Multiple SSI inclusions within a single page can be processed in parallel if they are handled by proxied or FastCGI/uwsgi/SCGI servers;
- **SSL and TLS SNI support;**
- **Support for HTTP/2** with weighted and dependency-based prioritization;
- **Support for HTTP/3.**

2. NGINX

- Nginx的主要特性
 - Basic HTTP server features
 - 基本HTTP服务器功能
 - **Other HTTP server features**
 - 其他HTTP服务器功能
 - Mail proxy server features
 - 邮件代理服务器功能
 - TCP/UDP proxy server features
 - TCP/UDP代理服务器功能
 - Architecture and scalability
 - 架构和可扩展性
 - Tested OS and platforms
 - 经过测试的操作系统和平台
- **Name-based and IP-based virtual servers;**
- **Keep-alive and pipelined connections support;**
- **Access log formats, buffered log writing, fast log rotation, and syslog logging;**
- **3xx-5xx error codes redirection;**
- **The rewrite module: URI changing using regular expressions;**
- **Executing different functions depending on the client address;**
- **Access control based on client IP address, by password (HTTP Basic authentication) and by the result of subrequest;**
- **Validation of HTTP referer;**
- **The PUT, DELETE, MKCOL, COPY, and MOVE methods;**
- **FLV and MP4 streaming;**
- **Response rate limiting;**
- **Limiting the number of simultaneous connections or requests coming from one address;**
- **IP-based geolocation;**
- **A/B testing;**
- **Request mirroring;**
- **Embedded Perl;**
- **njs scripting language.**

2. NGINX

□ Nginx的主要特性

- Basic HTTP server features
 - 基本HTTP服务器功能
- Other HTTP server features
 - 其他HTTP服务器功能
- Mail proxy server features
 - 邮件代理服务器功能
- TCP/UDP proxy server features
 - TCP/UDP代理服务器功能
- Architecture and scalability
 - 架构和可扩展性
- Tested OS and platforms
 - 经过测试的操作系统和平台

- One master and several worker processes; worker processes run under an unprivileged user;
- Flexible configuration;
- Reconfiguration and upgrade of an executable without interruption of the client servicing;
- Support for kqueue (FreeBSD 4.1+), epoll (Linux 2.6+), /dev/poll (Solaris 7 11/99+), event ports (Solaris 10), select, and poll;
- The support of the various kqueue features including EV_CLEAR, EV_DISABLE (to temporarily disable events), NOTE_LOWAT, EV_EOF, number of available data, error codes;
- The support of various epoll features including EPOLLRDHUP (Linux 2.6.17+, glibc 2.8+) and EPOLLEXCLUSIVE (Linux 4.5+, glibc 2.24+);
- sendfile (FreeBSD 3.1+, Linux 2.2+, macOS 10.5+), sendfile64 (Linux 2.4.21+), and sendfilev (Solaris 8 7/01+) support;
- File AIO (FreeBSD 4.3+, Linux 2.6.22+);
- DIRECTIO (FreeBSD 4.4+, Linux 2.4+, Solaris 2.6+, macOS);
- Accept-filters (FreeBSD 4.1+, NetBSD 5.0+) and TCP_DEFER_ACCEPT (Linux 2.4+) support;
- 10,000 inactive HTTP keep-alive connections take about 2.5M memory; (C10K)
- Data copy operations are kept to a minimum.

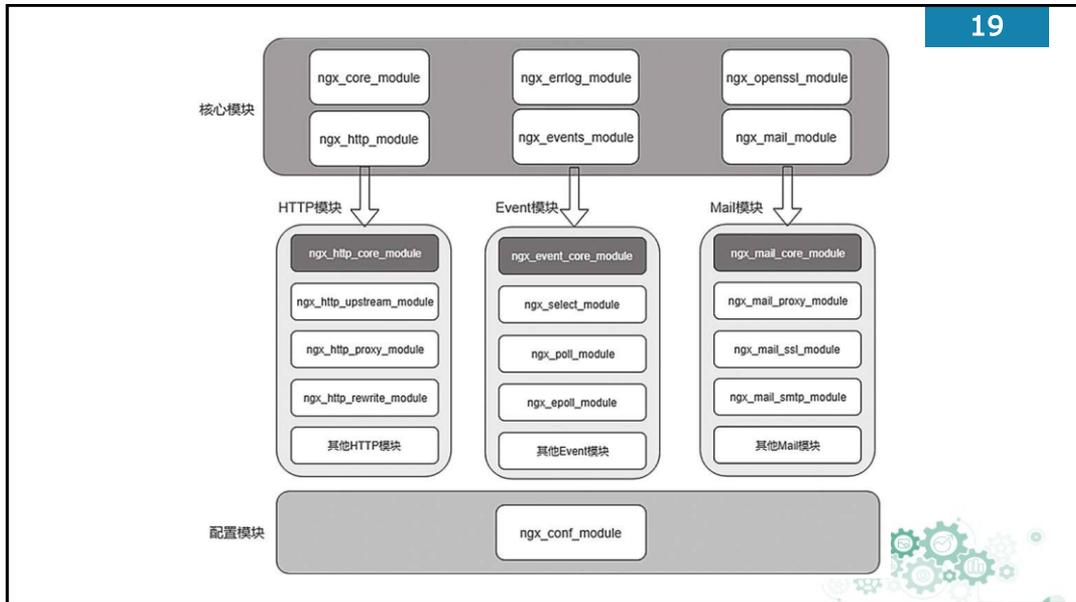
网络与信息系统智能运维课程体系 <https://internet.hactcm.edu.cn> 棒棒堂智能运维 <http://www.51xueweb.cn>

2. NGINX

2.3 NGINX模块

- Nginx是由内核和模块组成的，内核主要通过查找配置文件将客户端请求映射到location block，然后通过location block配置的指令启动不同的模块完成相应的工作。
 - Nginx的模块从结构上分为以下3种。
 - 核心模块：
 - 包含HTTP模块、EVENT模块和MAIL模块等
 - 基础模块：
 - 包含HTTP Access模块、HTTP FastCGI模块、HTTP Proxy模块及HTTP Rewrite模块等，主要用于提供基于http协议的基本服务，如访问控制、FastCGI支持、反向代理和URL重写等功能
 - 第三方模块：
 - 包含HTTP Upstream Request Hash模块、Notice模块和HTTP Access Key模块等，主要由Nginx社区或第三方开发并提供，用于实现更丰富和特定的功能

网络与信息系统智能运维课程体系 <https://internet.hactcm.edu.cn> 棒棒堂智能运维 <http://www.51xueweb.cn>



19

2. NGINX

2.3 NGINX模块

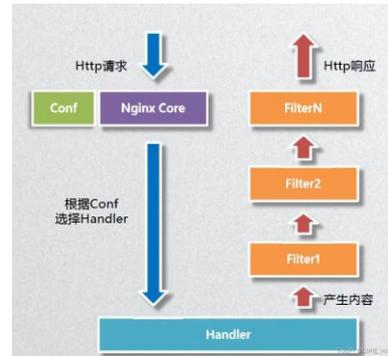
- Nginx是由内核和模块组成的，内核主要通过查找配置文件将客户端请求映射到location block，然后通过location block配置的指令启动不同的模块完成相应的工作。
 - Nginx的模块从功能上分为以下几种。
 - Core（核心模块）：构建Nginx基础服务，管理其他模块
 - Handlers（处理器模块）：
 - 此类模块直接处理请求，进行输出内容和修改headers信息等操作
 - Proxies（代理类模块）：
 - 此类模块是Nginx的HTTP Upstream之类的模块，这些模块主要与后端一些服务（比如FastCGI等）进行交互，实现服务代理和负载均衡等功能
 - Filters（过滤器模块）：
 - 此类模块主要对其他处理器模块输出的内容进行修改操作，最后由Nginx输出

20

2. NGINX

2.3 NGINX模块

- Nginx是由内核和模块组成的，内核主要通过查找配置文件将客户端请求映射到location block，然后通过location block配置的指令启动不同的模块完成相应的工作。
 - Nginx的核心模块主要负责建立Nginx服务模型、管理网络层和应用层协议以及启动针对特定应用的一系列模块，其他模块负责服务器的实际工作。
 - Nginx发送文件或转发请求到其他服务器时，由Handlers、Proxies模块提供服务。
 - Nginx把输出压缩或者增加一些数据时，由Filters模块提供服务。



2. NGINX

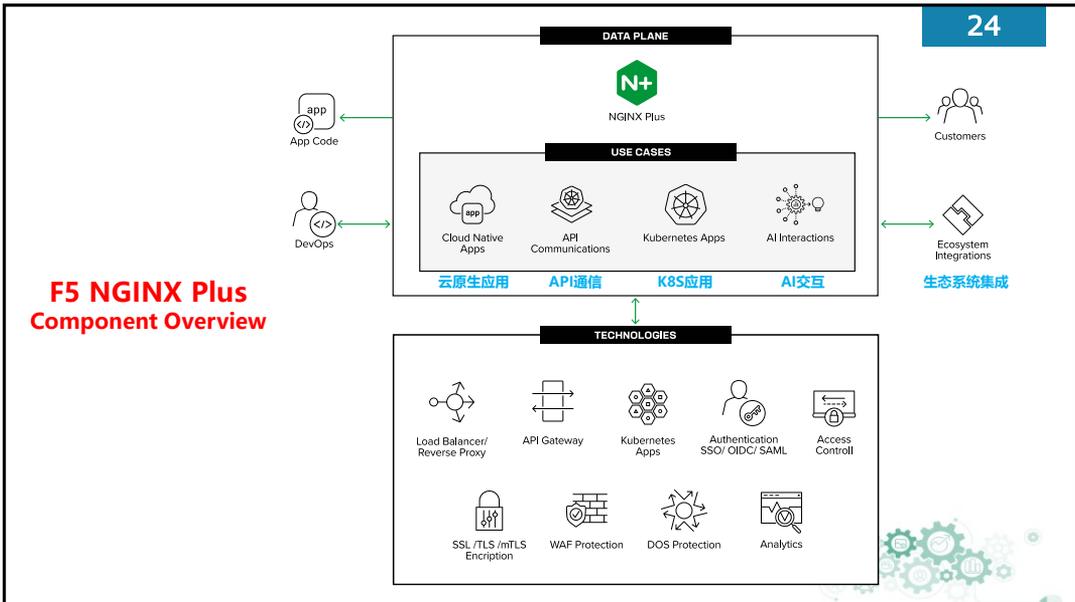
2.5 F5 NGINX Plus

- F5 NGINX Plus
 - The software load balancer, reverse proxy, web server, and content cache with the enterprise features and support you expect.
 - 软件负载均衡器、反向代理、Web 服务器和内容缓存，提供企业级功能和支持服务。
 - F5 NGINX Plus不仅仅是速度最快的Web服务器，还具备F5 NGINX开源版全套功能，并增加了企业级功能，例如高可用性、主动运行状况检查、DNS系统发现、会话持久性和RESTful API等。
 - Web服务器/反向代理：NGINX 是Web服务器和反向代理。
 - Kubernetes Ingress Controller：F5 NGINX Ingress Controller可在K8S平台提供企业级负载均衡。
 - API 网关：利用最值得信赖的API网关，提供实时API。
 - 负载均衡器：高性能、轻量级的负载均衡，满足NetOps和DevOps的需求。
 - 内容交付网络（CDN）：在不影响性能的情况下，减少网络流量。
 - 生态系统集成：
 - NGINX维护着一个第三方模块库，其模块经过全面的测试和认证，与NGINX Plus有良好的互操作性。



23

F5 NGINX Plus Component Overview



24

3. 使用NGINX实现代理服务



使用NGINX实现代理服务

任务目标:

- 使用NGINX实现反向代理服务
- 通过多种方式发布内部网站

操作步骤:

- 部署nginx
- 通过端口进行业务发布
- 通过虚拟目录进行业务发布
- 通过域名进行业务发布
- 配置nginx的日志
- 本地主机进行测试

操作演示:



3. 使用NGINX实现代理服务

- Nginx日志存放在/var/log/nginx目录下，包含访问日志access.log和错误日志error.log。
- nginx日志通过/etc/nginx/nginx.conf主配置文件进行设置，查看nginx.conf配置文件可获知日志的默认设置信息。

配置文件: /etc/nginx/nginx.conf

```

1. #nginx.conf 配置文件内容较多，本部分仅显示与日志配置有关的内容
2. #错误日志存放位置、记录等级等信息，默认为记录 error 等级日志
3. error_log /var/log/nginx/error.log;
4. #访问日志设置信息
5. http {
6.     #定义了名为 "main" 的日志格式
7.     log_format main '$remote_addr - $remote_user [$time_local] "$request"
8.                   '$status $body_bytes_sent "$http_referer"
9.                   "$http_user_agent" "$http_x_forwarded_for";
10.    #访问日志文件存储路径为/var/log/nginx/access.log，日志记录格式为 main 定义的格式
11.    access_log /var/log/nginx/access.log main;
12. }
```

3. 使用NGINX实现代理服务

3.2 NGINX的日志

表 4-2-5 日志格式中常用字符串含义

变量	含义
\$bytes_sent	发送给客户端的总字节数
\$connection	连接的序列号
\$connection_requests	当前通过一个连接获得的请求数量
\$msec	日志写入时间，单位为秒，精度是毫秒
\$pipe	如果请求是通过 HTTP 流水线（pipelined）发送的，值为“p”，否则为“.”
\$request_length	请求长度，包括请求头和请求正文
\$request_time	请求处理时间（从读入客户端的第一个字节开始，到最后一个字符发送给客户端后进行日志写入为止）单位为秒，精度是毫秒
\$status	响应状态
\$time_iso8601	IOS8601 标准下的服务器本地时间
\$time_local	通用日志格式下的服务器本地时间

网络与信息系统智能运维课程体系 <https://internet.hactcm.edu.cn>

棒棒堂智能运维 <http://www.51xueweb.cn>

27

4. 使用NGINX实现负载均衡

4.1 任务



使用NGINX实现负载均衡

任务目标：

- 使用NGINX实现负载均衡服务

操作步骤：

- 部署nginx
- 部署内部业务
- 通过负载均衡方式发布业务
- 配置业务安全防护
- 本地主机测试，验证负载均衡服务

操作演示：



网络与信息系统智能运维课程体系 <https://internet.hactcm.edu.cn>

棒棒堂智能运维 <http://www.51xueweb.cn>

28

4. 使用NGINX实现负载均衡

4.2 Nginx负载均衡模式

- Nginx主要通过ngx_http_upstream_module和ngx_http_proxy_module模块实现网站的负载均衡，实现方式有：
 - 轮询 (round-robin)：Nginx将客户端请求循环发送给各网站服务器节点，各网站服务器节点接收到的请求数量基本是一样的。Nginx默认为轮询模式。
 - 最少连接优先 (least-connected)：Nginx将避免把请求发送到繁忙的网站服务器节点，而是将请求发送给不太繁忙的网站服务器节点。
 - 持续会话 (ip-hash)：Nginx将客户端的会话一直保持在同一台网站服务器节点，直到该网站服务器节点不可用，一般用于需要维持session会话的网站业务。
 - 权重负载均衡 (Weighted load balancing)：Nginx根据设置的网站服务器权重信息，将客户端请求按照权重进行分发，权重值与访问比率成正比，一般用于服务器性能不均的情况。
 - 除了上述模式之外，Nginx还支持keepalive、least_time、random等方式。



4. 使用NGINX实现负载均衡

4.3 Nginx负载均衡健康检查

- Nginx通过两种方式对参与负载均衡的各网站服务器节点进行健康检查。
 - 被动模式：
 - 当接收到一个客户端请求时，Nginx会根据设置的负载均衡方式去请求相应的网站服务器节点，如果该节点连续失败多次（由max_fails设置值决定失败次数），则Nginx将其标记为失败状态，且在一段时间（由fail_timeout设置值决定时间段）内不再向其发送请求，继续请求下一个网站服务器节点。
 - 如果定义的所有网站服务器节点都请求失败，则返回给客户端Nginx定义的错误信息。
 - 设置时间过去后，Nginx会根据客户端请求探测该网站服务器节点，如果探测成功则将其标记为存活状态。
 - 主动模式：
 - Nginx会周期性地探测各网站服务器节点，同时对探测结果进行标记。
 - 主动模式是NGINX Plus版本的独有功能，NGINX版本仅支持被动检查模式。



5. 实现NGINX可视化

状态
监控

Web
控制台

集中
管理

5. 实现NGINX可视化

5.1 状态监控: stub_status

- ginx stub_status是 Nginx 的一个状态模块，可以提供关于 Nginx 服务器的一些基本运行状态信息。
- 通过访问配置了stub_status模块的 URL，可以获取到以下信息：
 - Active connections: 当前的活跃连接数。
 - server accepts handled requests:
 - 服务器接受的连接数、已处理的连接数和总请求数。
 - Reading: 当前正在读取请求头的连接数。
 - Writing: 当前正在写响应的连接数。
 - Waiting: 当前处于空闲状态，等待新请求的连接数。
- **注意:**
 - openEuler通过yum在线安装的nginx，缺失该模块。
 - 如果需要使用该模块，需要通过源代码编译安装。

33

```

Active connections: 2
server accepts handled requests
2 2 9
Reading: 0 Writing: 1 Waiting: 1

```

Active connections 当前客户端连接数

Server accepts handled requests

- 第1个数值: 接收的客户端连接总数
- 第2个数值: 已处理的客户端连接总数
- 第3个数值: 客户端请求总数

Reading Nginx正在读取请求信息的当前连接数

Writing Nginx将响应写回客户端的当前连接数

Waiting 当前等待请求的空闲客户端连接数

33

34

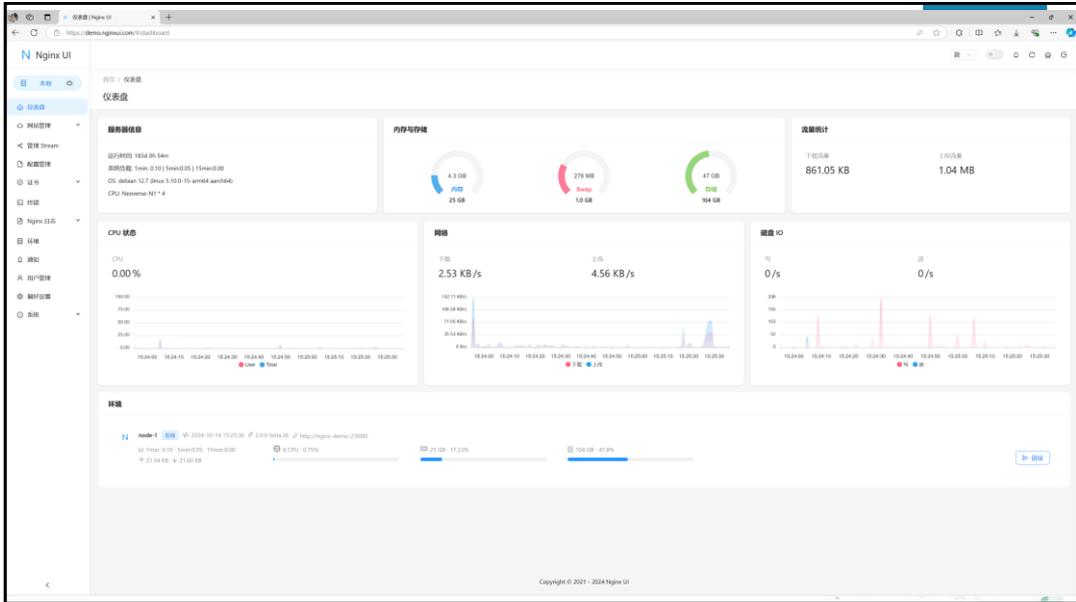
5. 实现NGINX可视化管理

5.2 Web控制台: Nginx UI

- Nginx UI是一个开源软件。
- Nginx UI是Nginx网络管理界面，旨在简化Nginx服务器的管理和配置。
 - 提供实时服务器统计数据、ChatGPT助手、一键部署、Let's Encrypt证书的自动续签以及用户友好的网站配置编辑工具。
 - 提供了在线访问Nginx日志、配置文件的自动测试和重载、网络终端、深色模式和自适应网页设计等功能。
 - Nginx UI采用Go和Vue.js构建，提供良好的体验。
- 官方地址: <https://nginxui.com>
- Github项目: <https://github.com/OxJacky/nginx-ui>

网络与信息系统智能运维课程体系 <https://internet.hactcm.edu.cn> 棒棒堂智能运维 <http://www.51xueweb.cn>

34



35

36

5. 实现NGINX可视化

5.3 集中管理: F5 NGINX Instance Manager

- F5 NGINX Instance Manager 可以简化对NGINX集群的配置和管理, 实现更加简单的监控、更新配置等。
 - 自动发现:
 - 能够自动发现所有的NGINX服务,
 - 并生成详细的列表。
 - 集中监控:
 - 实现增强的监控功能,
 - 包含更多的监控指标。
 - 管理配置:
 - 可以创建、修改、发布应用服务,
 - 以及进行安全配置。

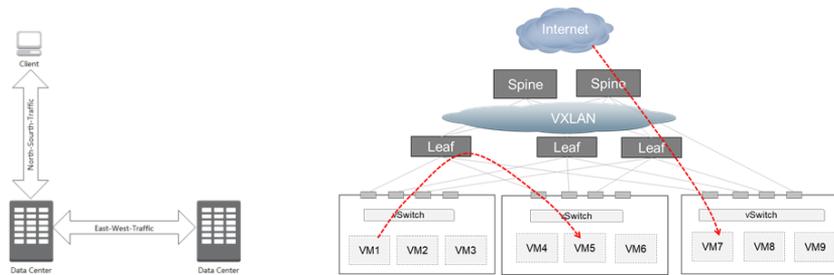
The diagram illustrates the F5 NGINX Instance Manager architecture. At the center is the 'NGINX Instance Manager' box, which is connected to three input boxes on the left: 'NGINX Open Source', 'NGINX Plus', and 'NGINX App Protect WAF'. The central box also contains the text 'Integrate into Workflows' and 'Boost Productivity through better information and greater fleet awareness'. To the right of the central box are four management capabilities, each with an icon and a brief description:

- Inventory:** Conduct an accurate count of your existing NGINX instances.
- Secure Apps and APIs at Scale:** Apply centralized security policies consistently from edge to cloud.
- Monitor:** Track the health of your fleet, gain ongoing visibility into deployments.
- Simplify Configurations:** Discover and resolve potential issues; identify systemic errors.

网络与信息系统智能运维课程体系 <https://internet.hactcm.edu.cn> 棒棒堂智能运维 <http://www.51xueweb.cn>

36

南北向流量 → 东西向流量



推荐阅读: <http://blog.nsfocus.net/east-west-flow-sum/>

37

网络与信息系统智能运维 课程体系学习平台

本课程体系由
河南中医药大学信息技术学院建设
课程体系学习平台由河南中医药大学医疗健康信息
工程技术研究所开发与保障
网络与信息系统智能运维课程体系学习平台
<https://internet.hactcm.edu.cn>
互联网运维管理工程应用丛书
<http://www.51xueweb.cn>

扫码学习
并获取课程资源



38