NGINX
Part of F5

nginx.org.cn

NGINX开源社区入门实践课

NGINX开源力作之NJS初体验

7/15 8:00PM
9:30PM (四)

扫码进入Zoom直播间

扫码进入bilibili直播间

扫码进入Zoom直播间/B站直播间
或者打开Zoom APP
网络研讨会ID：987 5477 2952
课程口令密码：66666 (5个6)

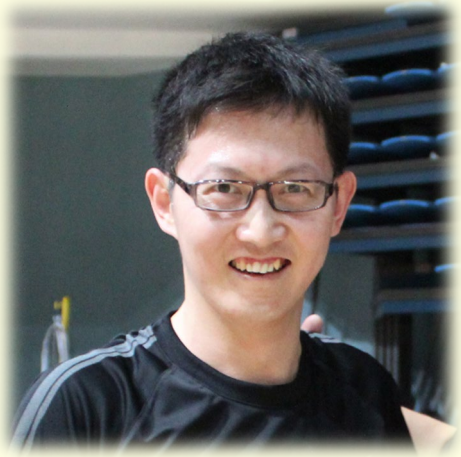
讲师：宗兆伟

F5Networks资深研发工程师

- 负责F5产品在中国的本地化工作
- NGINX和NJS在中国市场使用推广
- 新产品、新技术、开源技术的调研

NJS允许用户用脚本的方式给复杂业务场景下的流量负载添加处理逻辑，是扩展NGINX应用的一把利刃。本课程将讲解NJS的技术背景，并通过动手实践体会NJS的典型应用。

NGINX
Part of F5



主讲人：宗兆伟

F5资深研发工程师

作为F5中国本地化研发工程师，主要负责F5产品在中国的本地化工作，包括NGINX 和NJS 技术层面的中国市场使用推广；F5产品在OpenStack多云环境中的适配；F5新产品新技术、开源技术的调研；数据可视与分析能力的落地以及其他产品相关自动化工作等等。对云生态中用到的各种技术有深入的理解，对端到端自动化的理念有独到的见解。

课前准备：

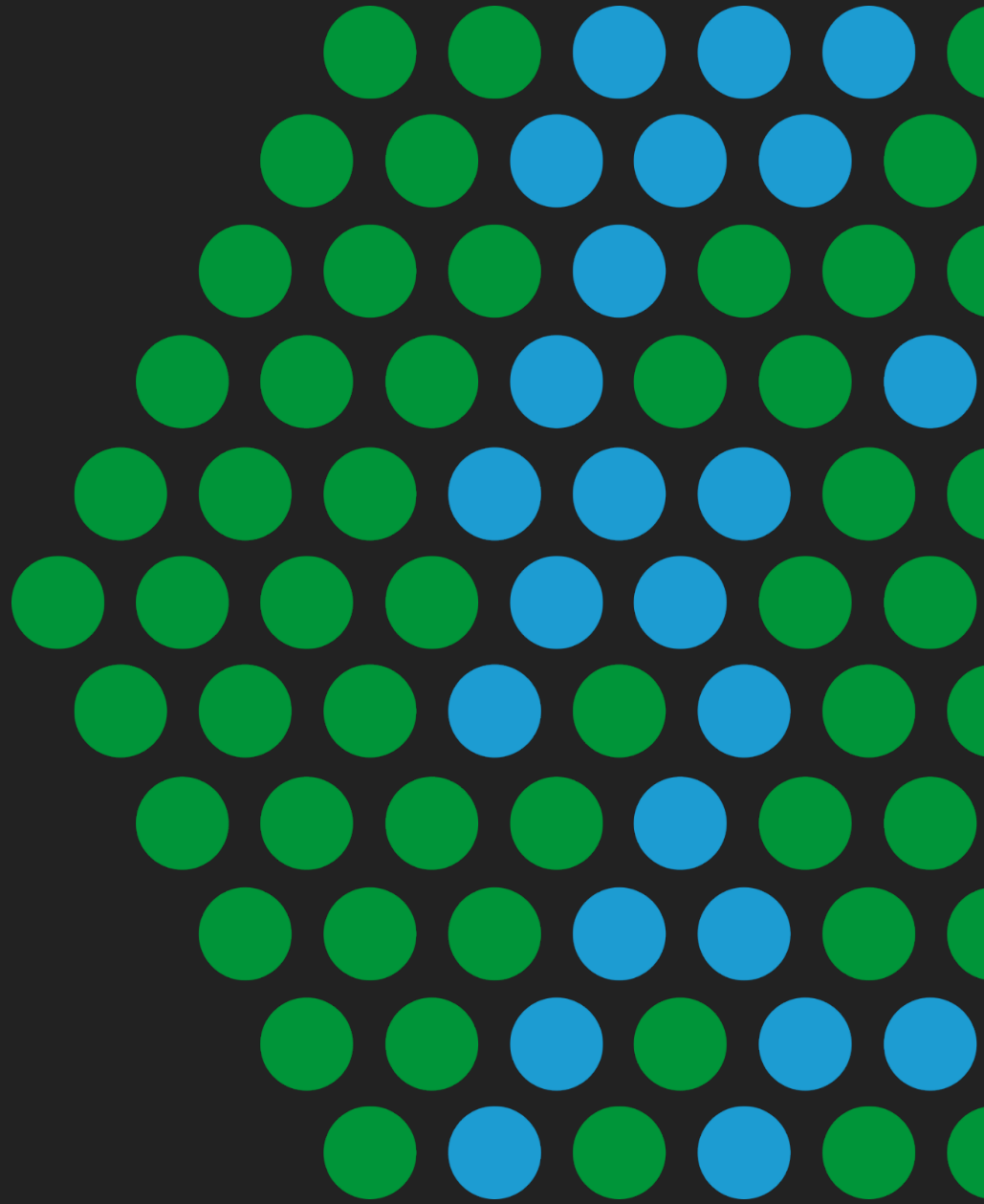
- 提前关注 https://github.com/f5devcentral/nginx-njs-usecases
- 按照其中的运行环境准备docker运行环境（docker & docker-compose）
- javascript基本代码阅读（或者编码）能力
- 命令行操作能力

nginx.org.cn

NGINX
Part of F5

NGINX开源力作之NJS初体验

Andrew Zong - 宗兆伟
F5Networks Senior Software Engineer
July 19, 2021



议题纲要

OPTIONAL SUBHEAD PLACEHOLDER

技术背景

发展历程

设计理念

核心价值

场景实践

使用方法

应用场景

技术背景

扩展NGINX功能的四种方式

C-MODULE、PERL、LUA、NJS

NJS 发展历程

NGINX增加脚本能力的又一次尝试

- Perl 模块 （2005 年 Igor）
  - 优势：**所有**现有Perl模块；**直接嵌入NGINX配置文件。**
  - **缺点**：**无异步支持**； Perl**全局解**释器。

- JS 模块（ 2015 年 Igor ）
  - **最流行的脚本**语言。
  - 类似C语言的语法。
  - **事件**驱动处理特性。

NJS核心价值

使用脚本的方式扩展应用服务能力

HARNESSING THE POWER AND CONVENIENCE OF JAVASCRIPT FOR EACH REQUEST

- **减少开**发投入：**减少用**户独立使用C语言开发特定场景的NGINX**模**块的可能性。
- **降低使用**难度：**将**JavaScript**代**码集成到NGINX HTTP**和流**（TCP / UDP）**模**块的事件处理模型中。
- **提高**产出效率：**使用**JavaScript**代**码扩展NGINX**配置**语法，**以**实现复杂的配置解决方案。

初识

安装NJS模块

RHEL/CENTOS为例

从源码安装模块

```
$ hg clone http://hg.nginx.org/njs  # or git clone https://github.com/nginx/njs

$ ./configure --add-dynamic-module=path-to-njs/nginx
```

安装预编译模块

```
$ sudo yum install yum-utils

$ vim /etc/yum.repos.d/nginx.repo

$ sudo yum install nginx

$ sudo yum install nginx-module-njs
```

```
[nginx-stable]
name=nginx stable repo
baseurl=http://nginx.org/packages/centos/$releasever/$basearch/
gpgcheck=1
enabled=1
gpgkey=https://nginx.org/keys/nginx_signing.key
module_hotfixes=true

[nginx-mainline]
name=nginx mainline repo
baseurl=http://nginx.org/packages/mainline/centos/$releasever/$basearch/
gpgcheck=1
enabled=0
gpgkey=https://nginx.org/keys/nginx_signing.key
module_hotfixes=true
```

https://nginx.org/en/linux_packages.html#RHEL-CentOS

NJS基本使用入门

nginx.conf

```
load_module modules/ngx_http_js_module.so;

events {}

http {
    js_import http.js;

    server {
        listen 8000;

        location / {
            js_content http.hello;
        }
    }
}
```

http.js

```
function hello(r) {
    r.return(200, "Hello world!");
}

export default {hello};
```

NJS参考文档：
http://nginx.org/en/docs/njs/

应用示例：
https://github.com/xeioex/njs-examples
https://github.com/f5devcentral/nginx-njs-usecases

博客：
https://nginx.com/blog/tag/javascript

源代码：
https://github.com/nginx/njs

视频资料：
NGINX JavaScript and NJS deepDive - From 0 to Hero

NJS异步Promise应用框架

```
// initialzation of connection
ngx.fetch('http://nginx.org/en/docs/njs', opts)

// sometime later, when the host replies
.then(reply => reply.text())

// sometime later, when the body reading is done
.then(body => r.return(200, body.toString()))

// if something goes wrong, the execution goes here.
.catch(e => r.return(501, e.message))
```

ngx.fetch

Promise.then()

resolve()/reject()

() => {}

在NJS中使用npm包

NJS中应用npm包，需要解决的问题：

1. 多个文件相互依赖问题
2. 平台相关API调用
3. 语言规范的兼容性问题

“browser-specific shim”

+

/ browserify

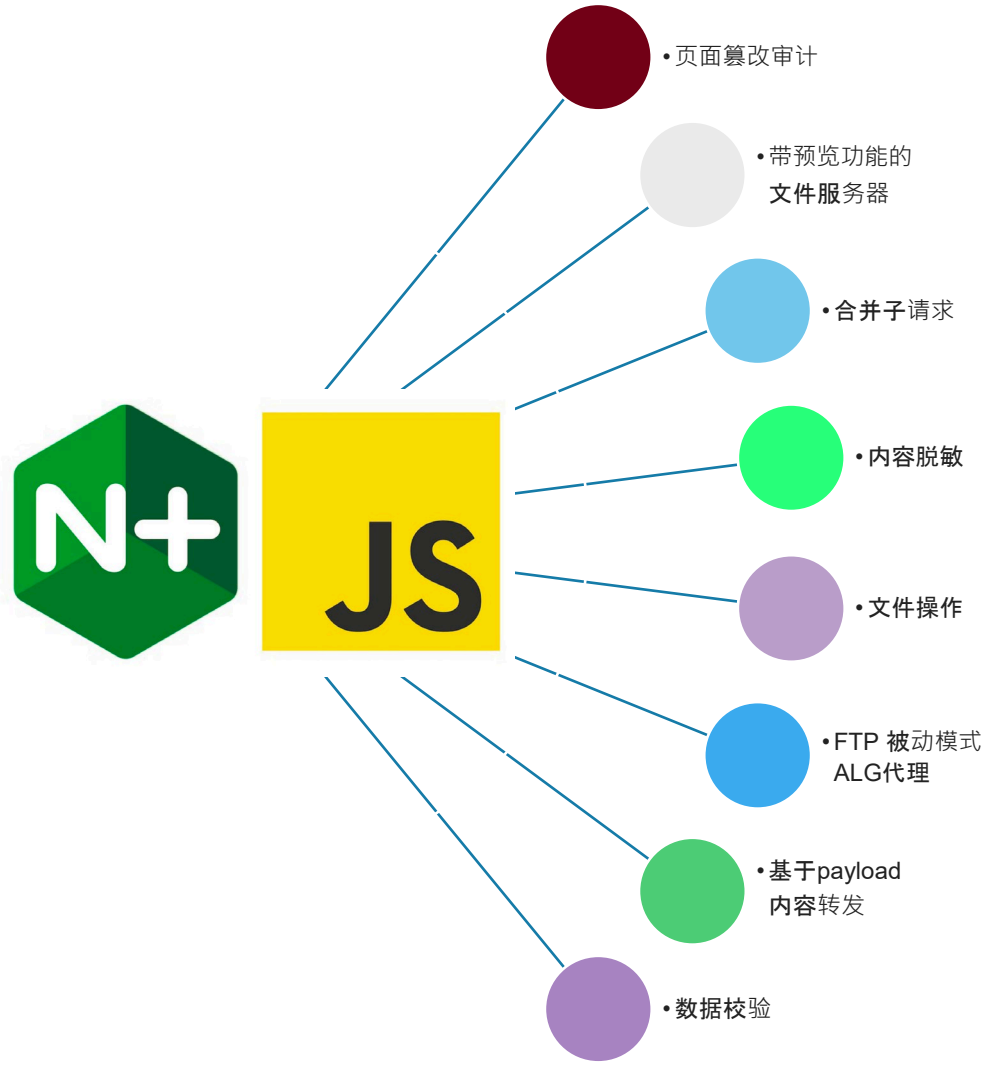
+

场景实践

实践准备

- Docker 运行环境（docker & docker-compose）
- Docker Images
  - node:14.17.3-alpine3.14
  - nginx:1.21.1
  - centos:7
- curl & <web browser>

场景介绍



NGINX的通用脚本框架

extend web services

Harnessing the Power and Convenience of JavaScript for Each Request

https://github.com/f5devcentral/nginx-njs-usecases

NGINX®

Part of F5

NGINX
Part of F5

NGINX开源社区是面向所有NGINX用户的开发者社区。

我们秉持**“开放，包容，沟通，贡献”**之宗旨，与业界共建开放、包容、活跃的“NGINX用户之家”；秉承开源的精神，在社区治理上高度开放，为所有NGINX的用户，开发者和技术爱好者，提供一个方便学习、讨论的场所。

期待您成为此社区中活跃的一员，贡献您的文章，博客，代码，踊跃讨论与回答问题，打造您个人品牌和影响力。

社区网站注册

社区公众号

加入微信群

nginx.org.cn