NGINX 102

OLF 2022

Tim Quinlan
Technical Account Manager
Agenda

Very brief history of NGINX
Configuration and operation
Reverse proxy discussion and demo
Scripting discussion and demo
High availability discussion (demo time permitting)
Q&A
History
Very Brief History of NGINX

2004 NGINX open sourced
   C10K Problem
   Outperform Apache

2011 NGINX, Inc. formed
   NGINX Plus
   Software subscription and support

2019 NGINX acquired by F5
   Foundational software for Modern Applications
License and Support

BSD 2-Clause

NGINX Open Source freely available
   Community support or software subscription

NGINX Plus based on NGINX Open Source
   Available only as software subscription
Popularity

Web server developers: Market share of all sites

- Apache
- Microsoft
- Sun
- nginx
- Google
- Cloudflare
- NCSA
- LiteSpeed
- OpenResty
- Other
## Sources

<table>
<thead>
<tr>
<th>F5</th>
<th>Community</th>
</tr>
</thead>
<tbody>
<tr>
<td>Official Docker Image (mainline/stable)</td>
<td>NGINX Open Source (mainline/stable)</td>
</tr>
<tr>
<td>NGINX Open Source Packages *</td>
<td>OS Distros</td>
</tr>
<tr>
<td>NGINX Plus Packages *</td>
<td>Docker community</td>
</tr>
</tbody>
</table>

* Subscription available (https://www.nginx.com/support/)
Open Source & Free Projects
The Swiss Army Knife of reliable, high-performance traffic management

NGINX Open Source
Simplify traffic management with a proxy that can handle everything from web serving to API traffic

NGINX Unit
Run web applications and APIs, serve static content, and proxy to backends with a universal web app server

NGINX JavaScript
Extend NGINX with sophisticated configuration solutions for server-side use cases and per-request processing

Agent
Manage NGINX deployments with companion software that provides observability and a config API

NGINX Amplify
Run real-time diagnostics for NGINX Open Source and NGINX Plus

Use anywhere

Kubernetes-native

NGINX Ingress Controller
Manage app connectivity at the edge of a Kubernetes cluster with API gateway, identity, and observability features

NGINX Kubernetes Gateway
Experiment with the new Gateway API using NGINX as the data plane

NGINX Service Mesh
Implement developer-friendly service-to-service connectivity, security, orchestration, and observability
An Architecture for Modern Apps

<table>
<thead>
<tr>
<th>Data Plane</th>
<th>Enterprise</th>
<th>Free</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Plane</strong></td>
<td>NGINX Plus</td>
<td>NGINX App Protect</td>
</tr>
<tr>
<td>Load Balancing</td>
<td>Web Application Gateway</td>
<td>Web Application Delete</td>
</tr>
<tr>
<td>API Gateway</td>
<td>Content Cache</td>
<td>Denial of Service</td>
</tr>
<tr>
<td>NGINX Open Source</td>
<td>Reverse Proxy</td>
<td>Reverse Proxy</td>
</tr>
<tr>
<td>NGINX Unit</td>
<td>App Server</td>
<td>App Server</td>
</tr>
</tbody>
</table>

The **Data Plane** houses and transports application and data traffic.

All policies, service-level agreements (SLAs), and scaling or behavior triggers (such as retries, keepalives, and horizontal scaling) are executed at this tier.

<table>
<thead>
<tr>
<th>Control Plane</th>
<th>NGINX Management Suite</th>
<th>NGINX Ingress Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Control Plane</strong></td>
<td>Instance Manager</td>
<td>Kubernetes Ingress</td>
</tr>
<tr>
<td>Instance Discovery</td>
<td>Configuration Management</td>
<td>Lightweight Mesh</td>
</tr>
</tbody>
</table>

The **Control Plane** configures rules for the data plane.

It resides above the data plane, as a separate entity, and enforces rules for the data plane. In Kubernetes, it oversees orchestration and coordination of containers, nodes, pods, and clusters and makes global decisions about the cluster (e.g. scheduling), while detecting and responding to cluster events.

<table>
<thead>
<tr>
<th>Management Plane</th>
<th>NGINX Management Suite</th>
<th>NGINX Amplify</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Management Plane</strong></td>
<td>API Connectivity Manager</td>
<td>Monitoring</td>
</tr>
<tr>
<td>Developer Portal</td>
<td>API Gateway Management</td>
<td></td>
</tr>
</tbody>
</table>

The **Management Plane** sets guardrails for the data plane and control plane.

It enforces common standards, access controls, and policies across distributed environments. Reduces complexity by abstracting some control plane operations and provides visibility and insight into the application performance.
Configuration and Operation
Configuration Context Hierarchy

- main
  - events
  - stream
    - server
  - http
    - server
      - location
    - upstream
Context matters for directives and inheritance

**Syntax:** `limit_req_zone key zone=name:size rate=rate [sync];`
**Default:** —
**Context:** http

**Syntax:** `limit_req zone=name [burst=number] [nodelay | delay=number];`
**Default:** —
**Context:** http, server, location
Managing Configuration

Standard configuration management options:

- Monolithic
- Conf.d
- Includes
Managing Configuration

Check config:
    nginx -t

Print config:
    nginx -T

Apply config:
    nginx -s reload
Docker Image
Quick Start

$ docker run -p 80:80 -v $(pwd)/html:/usr/share/nginx/html nginx:latest
Reverse Proxy
Simple Reverse Proxy
Reverse Proxy Use Cases

Caching
Load Balancing
SSL Termination
API Gateway
Reverse Proxy Setup

**Upstreams:** http context

**Keepalive:** upstream context

**Load balancing algorithms:** upstream context

**Persistence:** upstream context

Reverse Proxy Demo
Rate Limiting
Rate Limiting

Limit Request Zone: http context

Limit Request: http, server or location context

Rate Limiting Demo
Mirroring
Reverse Proxy with Mirror
Rate Limiting

**Mirror Location:** server context  
**Mirror Directive:** location context

Mirroring Demo
Extending NGINX
Extending NGINX

Modules

C

Extensive community (https://www.nginx.com/resources/wiki/modules/)

Many NGINX features are modules, we’ve already seen:

Upstream
Mirror
Rate limit
Extending NGINX with NJS

NJS

Subset of JavaScript
ECMAScript 5.1 compatible (some ECMAScript 6 extensions)
Extending NGINX with NJS

NJS
Subset of JavaScript
ECMAScript 5.1 compatible (some ECMAScript 6 extensions)
Implemented with ngx_http_js_module
Extending NGINX with NJS

Load NJS Module: main context
Include the JS: http context
Import the JS: http, server or location context
Call the JS functions:
  js_content to set a location handler
  js_body_filter to filter response
NJS Demo
High Availability
NGINX Plus Features

Shared Zones
Active Health Checks
Dynamic Upstreams

(https://www.nginx.com/products/nginx/compare-models)
HA Reverse Proxy with NGINX Plus

Goals
Redundancy
1000 r/s

Reverse Proxy Tier
(1000 r/s per cluster)

Upstream Tier
HA Reverse Proxy with NGINX Plus

Goals
Redundancy
1000 r/s

Reverse Proxy Tier
(1000 r/s per cluster)

Upstream Tier
HA Reverse Proxy with NGINX Plus

Active Health Check, proxy stops sending requests to unhealthy upstreams

Goals

Redundancy
1000 r/s

Reverse Proxy Tier
(1000 r/s per cluster)

Upstream Tier
HA Reverse Proxy with NGINX Plus

Goals
Redundancy
1000 r/s

Reverse Proxy Tier
(1000 r/s per cluster)

Upstream Tier
HA Reverse Proxy with NGINX Plus

Shared rate limiting maintains r/s across cluster

Goals
Redundancy
1000 r/s

Reverse Proxy Tier
(1000 r/s per cluster)

Upstream Tier
High Availability with NGINX Plus Demo
(Time permitting)
High Availability NGINX Plus Info

License Required

Sign up ([https://www.nginx.com/free-trial-request](https://www.nginx.com/free-trial-request))

Or see me at the booth (this way supports OLF!!)

Example Material

[https://github.com/timquinlan/nginxplus_api_gw](https://github.com/timquinlan/nginxplus_api_gw)
**Links**

NGINX Material: [https://github.com/timquinlan/nginx102](https://github.com/timquinlan/nginx102)

NGINX Plus Material: [https://github.com/timquinlan/nginxplus_api_gw](https://github.com/timquinlan/nginxplus_api_gw)

NGINX Plus Trial: [https://www.nginx.com/free-trial-request](https://www.nginx.com/free-trial-request)

NGINX Docs: [https://nginx.org/en/docs](https://nginx.org/en/docs)

NGINX Plus Docs: [https://docs.nginx.com/](https://docs.nginx.com/)


NGINX Plus High Availability: [https://docs.nginx.com/nginx/admin-guide/high-availability/](https://docs.nginx.com/nginx/admin-guide/high-availability/)

NGINX Modules: [https://www.nginx.com/resources/wiki/modules/](https://www.nginx.com/resources/wiki/modules/)

NJS Examples: [https://github.com/nginx/njs-examples](https://github.com/nginx/njs-examples)
Contact:

Tim Quinlan

t.quinlan@f5.com

@trquacker