©2022 F5

F5 NGINX

NGINX 102

OLF 2022

Tim Quinlan

Technical Account Manager
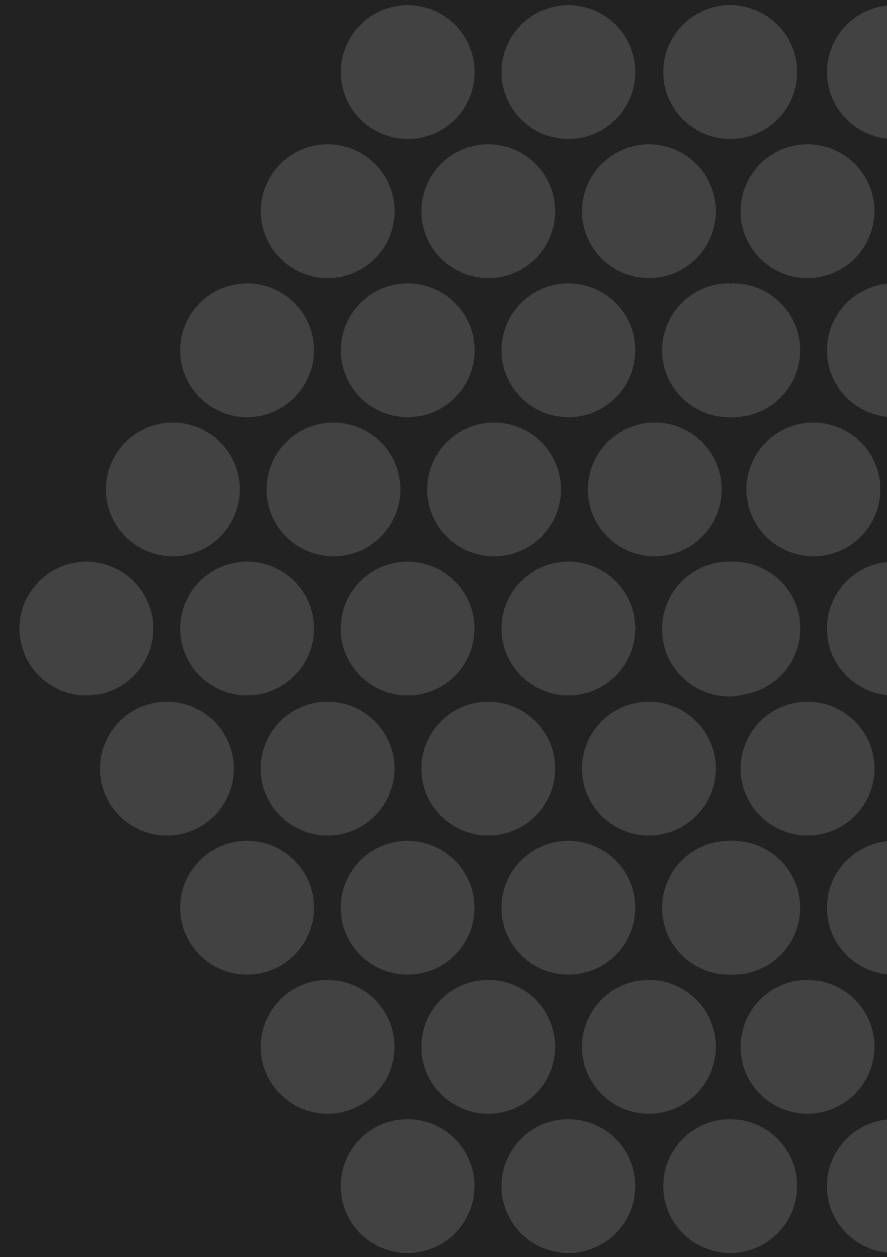
NGINX
Part of F5

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
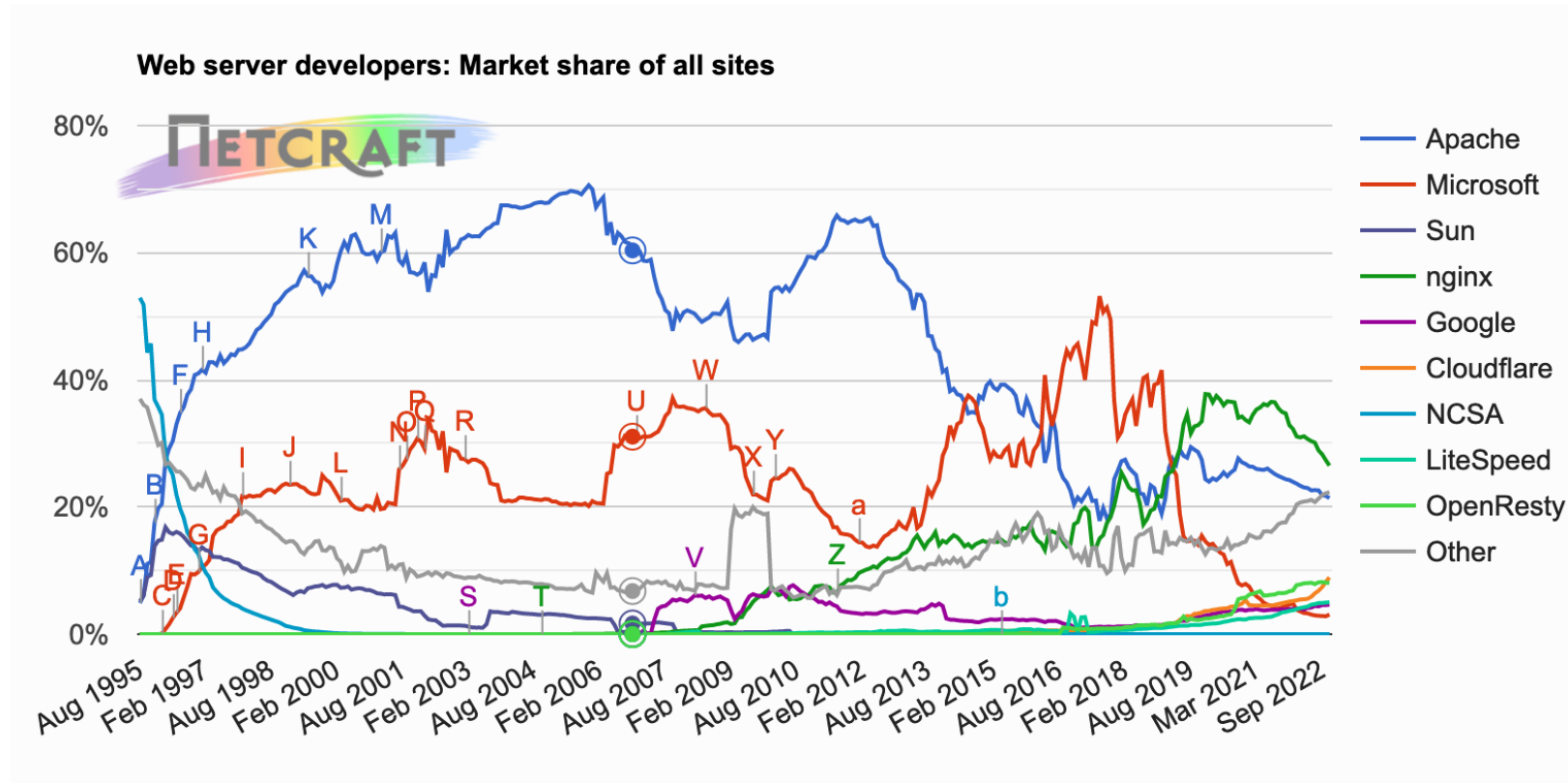
License and Support

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

©2022 F5

Popularity

Web server developers: Market share of all sites

Sources

F5	Community
Official Docker Image (mainline/stable)	NGINX Open Source (mainline/stable)
NGINX Open Source Packages *	OS Distros
NGINX Plus Packages *	Docker community

* Subscription available (https://www.nginx.com/support/)

©2022 F5

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 JavaScript

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

Agent
An NGINX Project

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

Unit
An NGINX Project

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

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
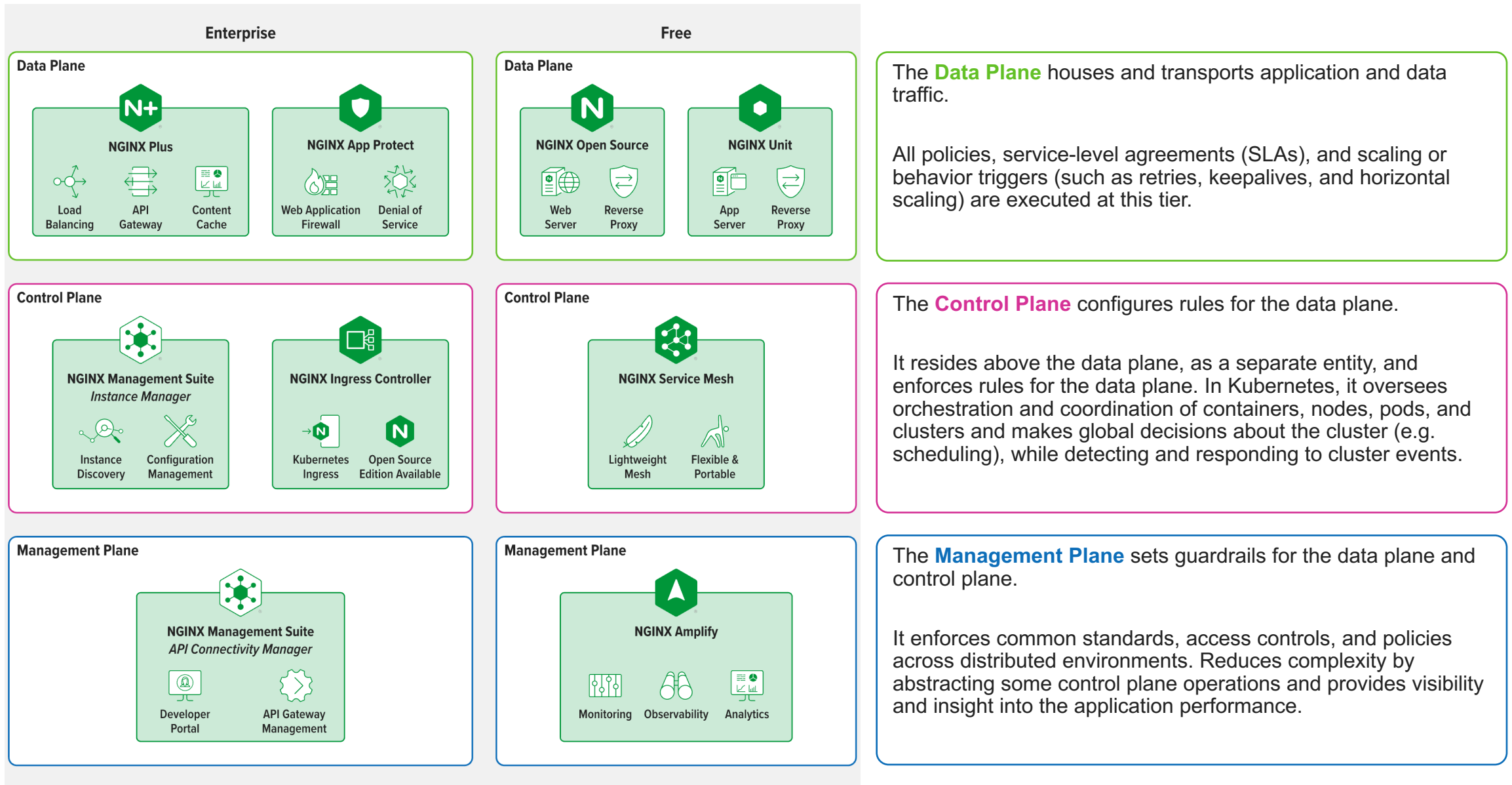
NGINX Kubernetes Gateway

Experiment with the new Gateway API using NGINX as the data plane
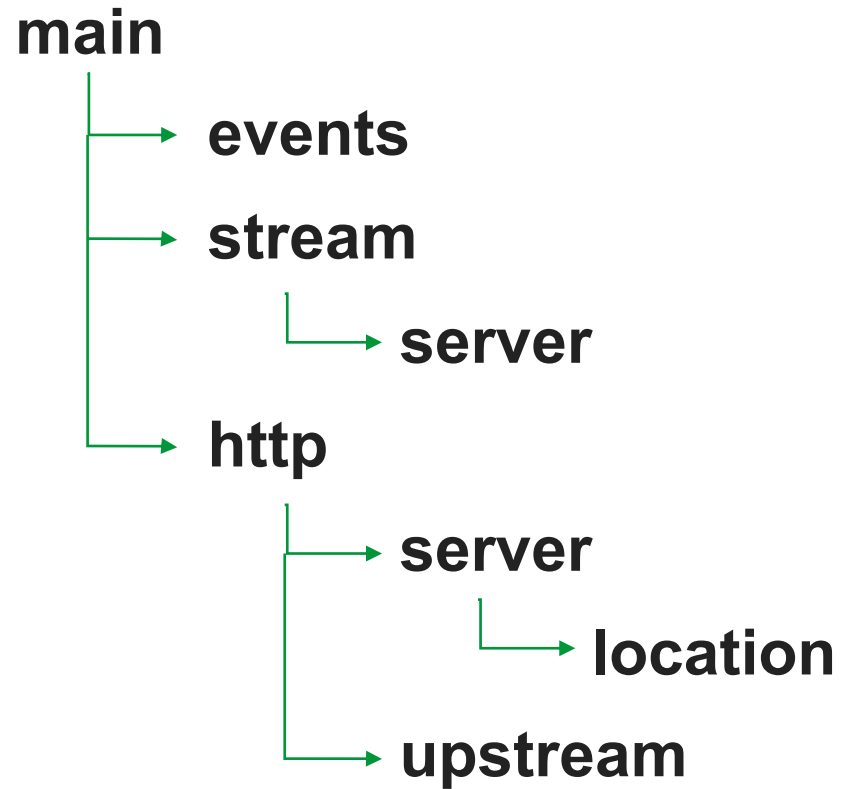
NGINX Service Mesh

Implement developer-friendly service-to-service connectivity, security, orchestration, and observability

©2022 F5

An Architecture for Modern Apps

Enterprise

Data Plane

NGINX Plus
- Load Balancing
- API Gateway
- Content Cache

NGINX App Protect
- Web Application Firewall
- Denial of Service

Control Plane

NGINX Management Suite
Instance Manager
- Instance Discovery
- Configuration Management

NGINX Ingress Controller
- Kubernetes Ingress
- Open Source Edition Available

Management Plane

NGINX Management Suite
API Connectivity Manager
- Developer Portal
- API Gateway Management

Free

Data Plane

NGINX Open Source
- Web Server
- Reverse Proxy

NGINX Unit
- App Server
- Reverse Proxy

Control Plane

NGINX Service Mesh
- Lightweight Mesh
- Flexible & Portable

Management Plane

NGINX Amplify
- Monitoring
- Observability
- Analytics

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.

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.

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**

Documentation

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
```

©2022 F5

Managing Configuration

Standard configuration management options:

Monolithic

Conf.d

Includes

©2022 F5

Managing Configuration

Check config:

nginx -t

Print config:

ningx -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

©2022 F5

Simple Reverse Proxy

Upstream

Reverse Proxy

Upstream

©2022 F5
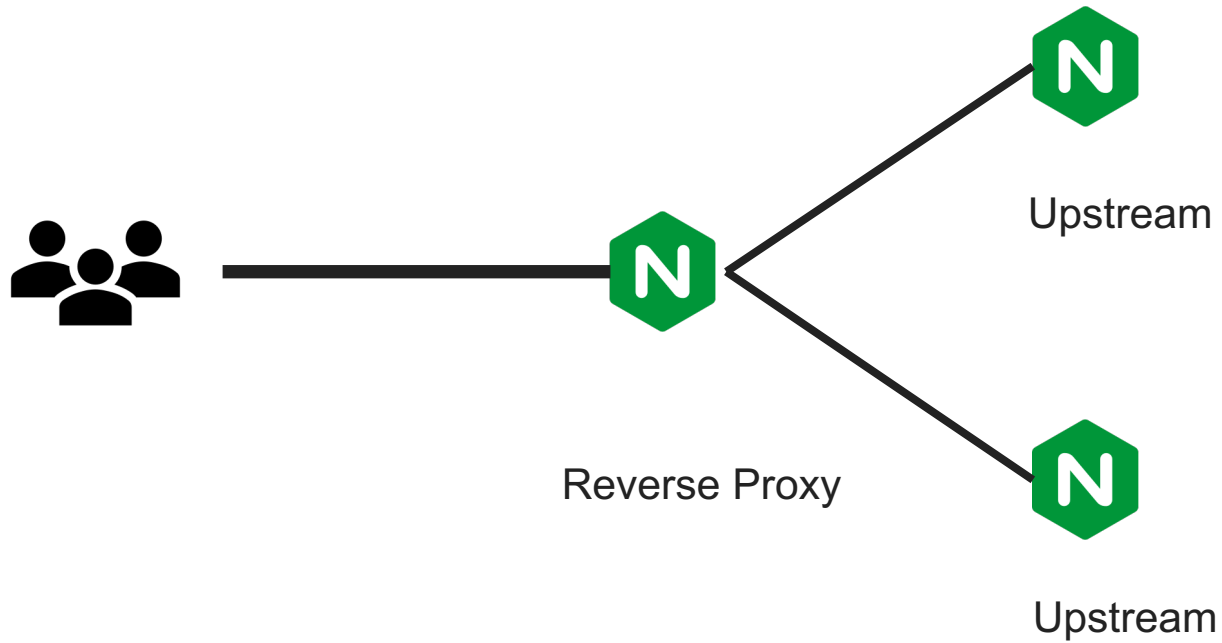
Reverse Proxy Use Cases

Caching

Load Balancing

SSL Termination

API Gateway

Reverse Proxy Setup

Upstreams: http context

Keepalive: upstream context

Load balancing algorithms: upstream context

Persistence: upstream context

(http://nginx.org/en/docs/http/ngx_http_upstream_module.html)

Reverse Proxy Demo

Rate Limiting

Rate Limiting

Limit Request Zone: http context

Limit Request: http, server or location context

(http://nginx.org/en/docs/http/ngx_http_limit_req_module.html)

Rate Limiting Demo

Mirroring

Reverse Proxy with Mirror

Upstream

Upstream

Mirror

©2022 F5
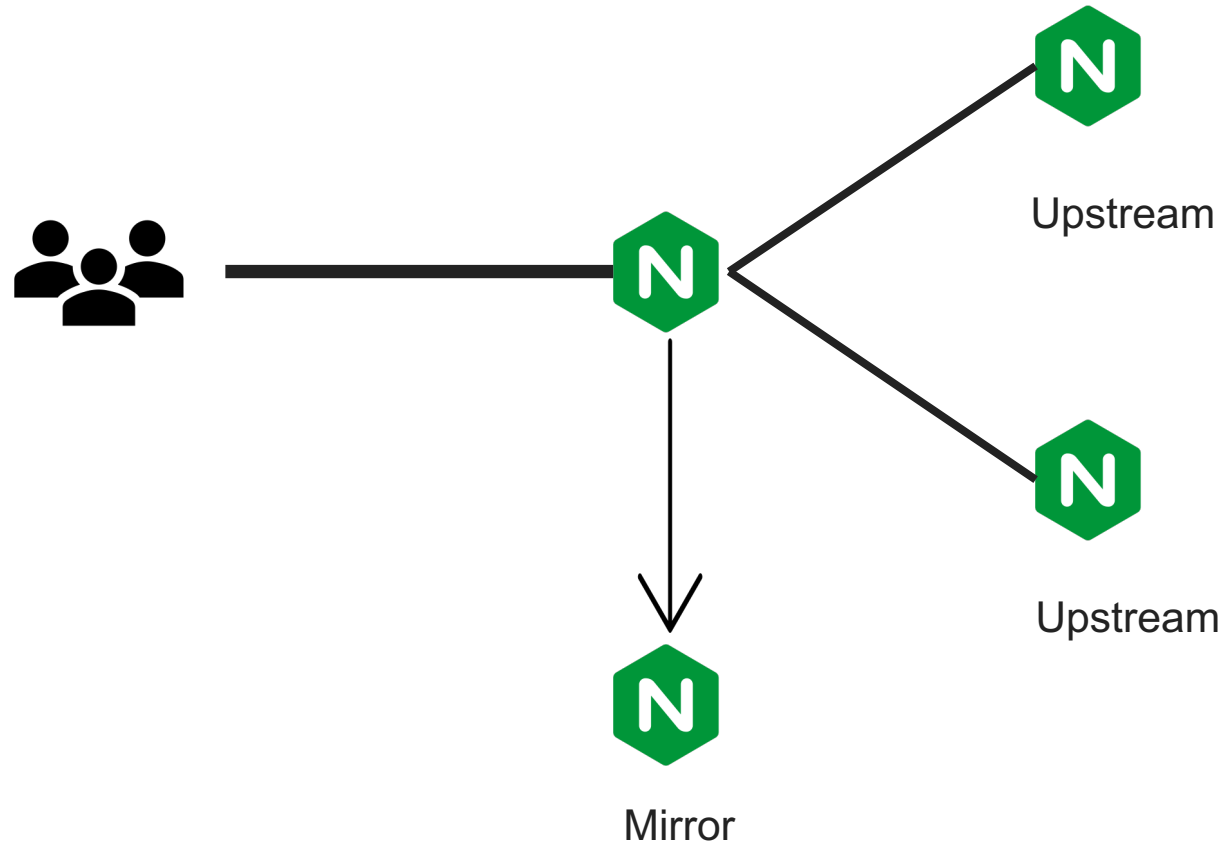
Rate Limiting

Mirror Location: server context

Mirror Directive: location context

(http://nginx.org/en/docs/http/ngx_http_mirror_module.html)

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

Full docs at (http://nginx.org/en/docs/http/ngx_http_js_module.html)

NJS Demo

©2022 F5

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

Reverse Proxy Tier

(1000 r/s per cluster)

Upstream Tier

Goals

Redundancy

1000 r/s

©2022 F5

HA Reverse Proxy with NGINX Plus

Reverse Proxy Tier

(1000 r/s per cluster)

Upstream Tier
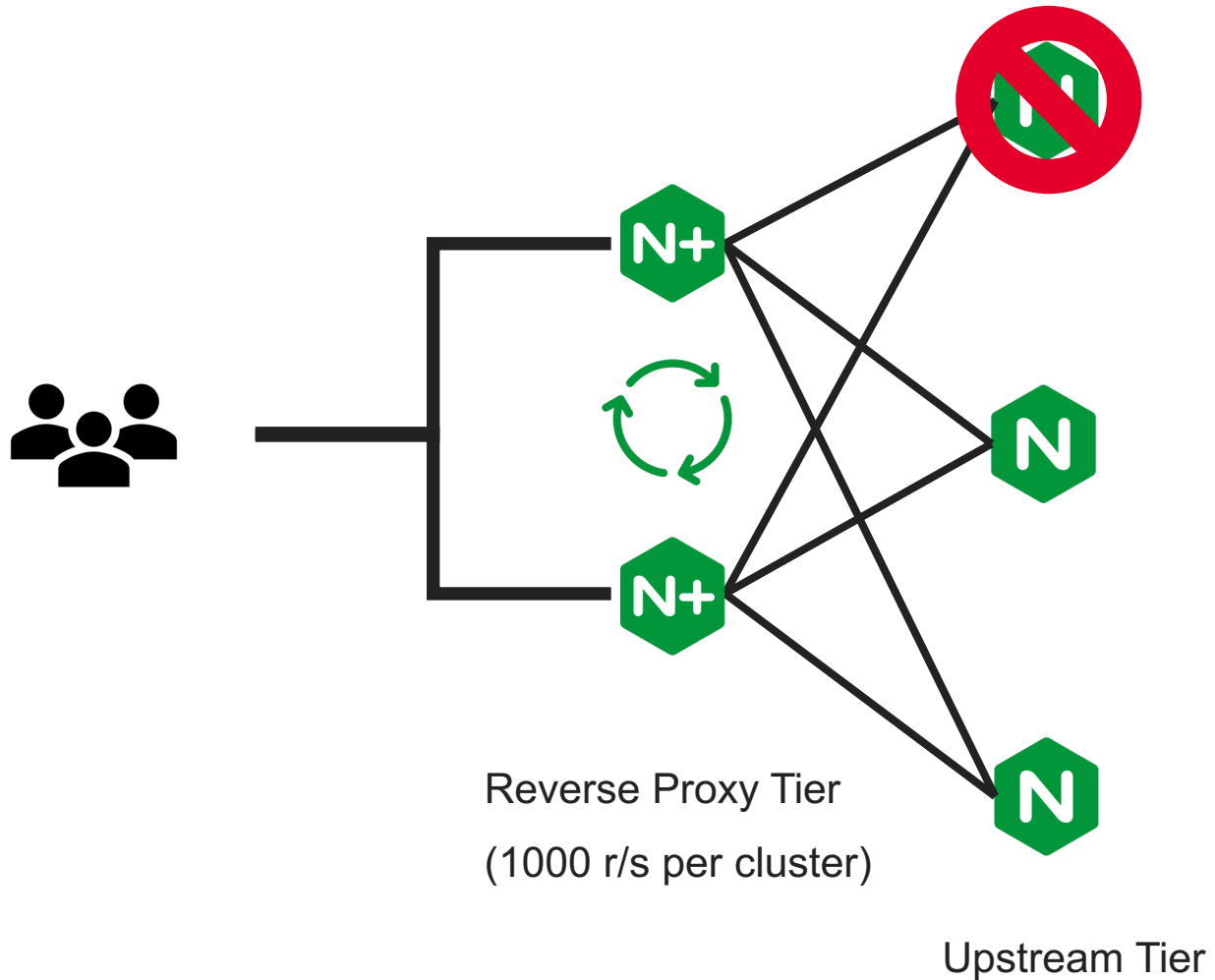
Goals

Redundancy

1000 r/s

HA Reverse Proxy with NGINX Plus

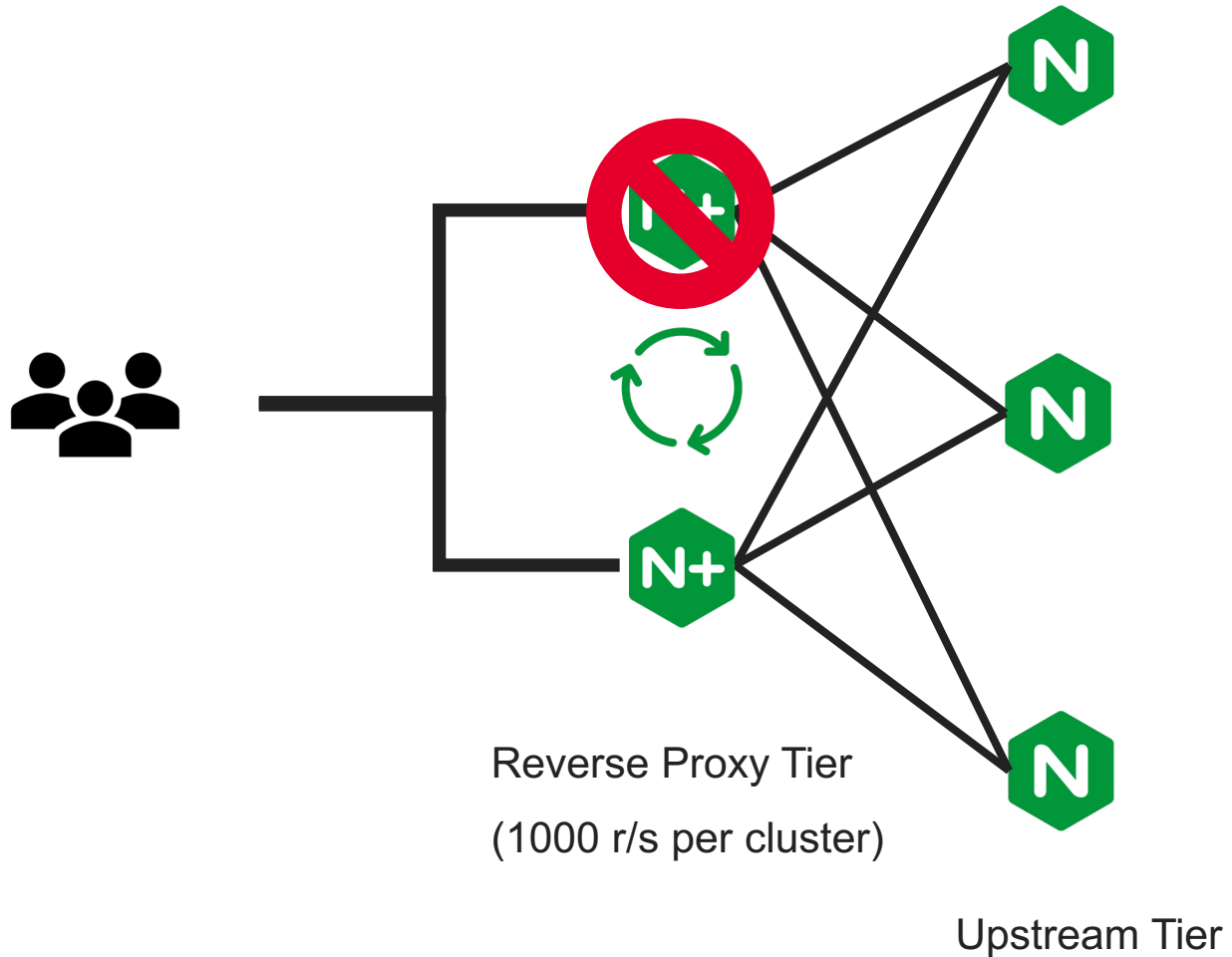Active Health Check, proxy stops sending requests to unhealthy upstreams
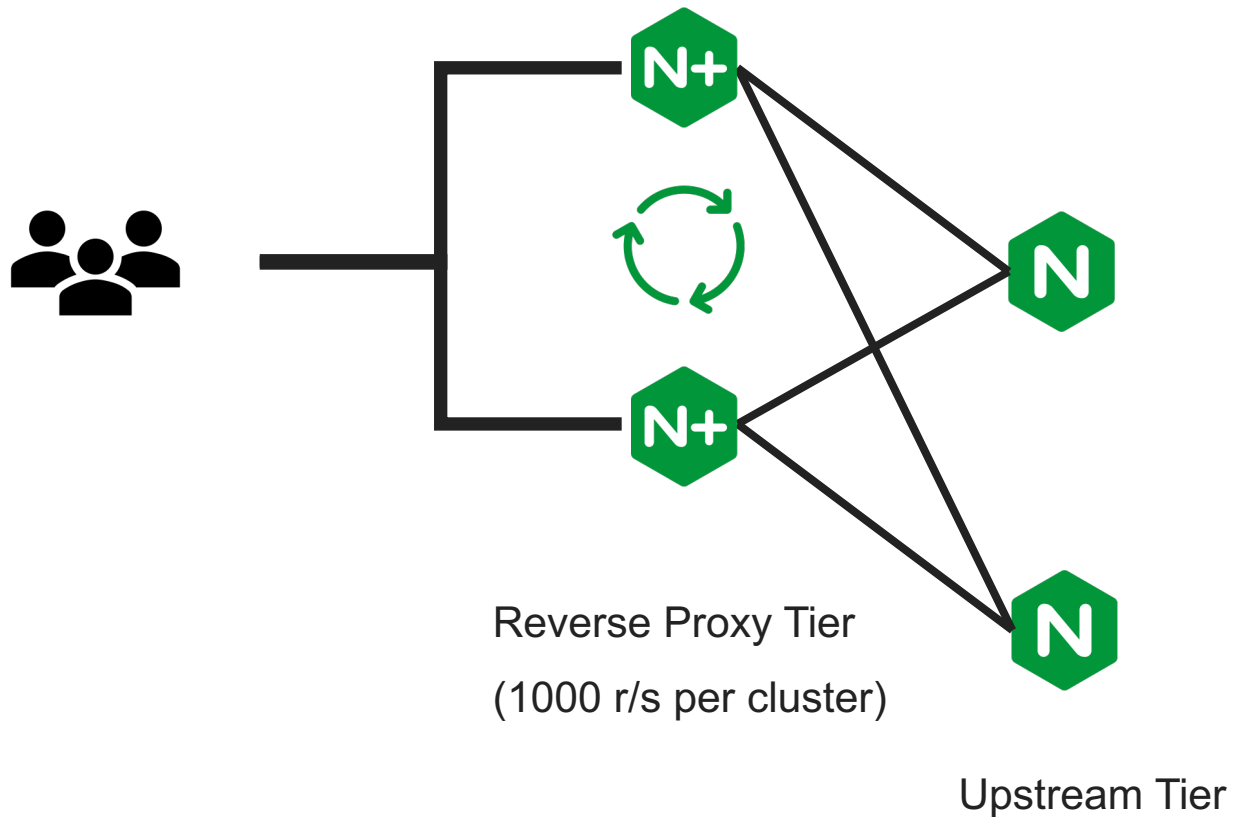
Goals

Redundancy

1000 r/s

Reverse Proxy Tier
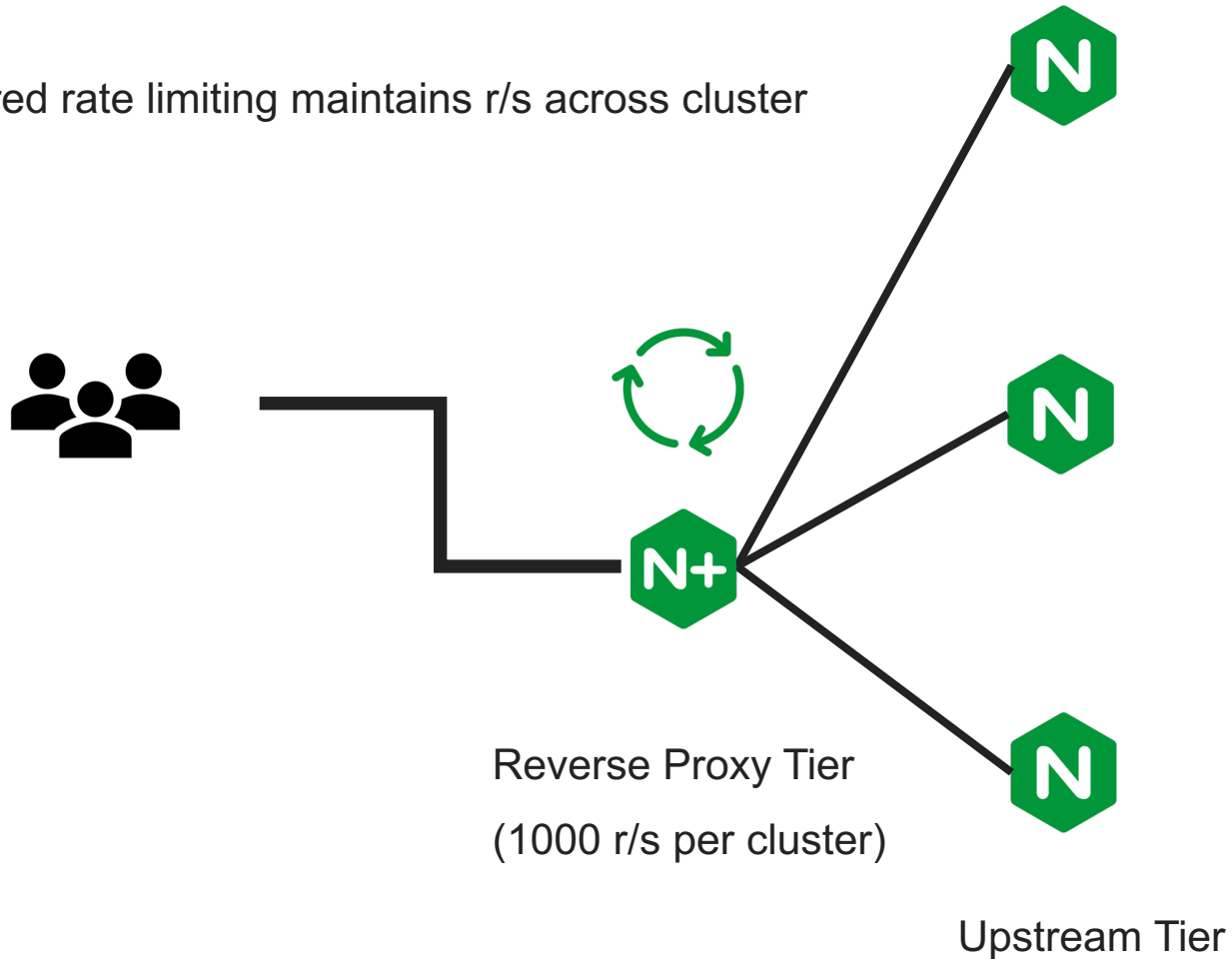
(1000 r/s per cluster)

Upstream Tier

©2022 F5

HA Reverse Proxy with NGINX Plus

Reverse Proxy Tier

(1000 r/s per cluster)

Upstream Tier

Goals

Redundancy

1000 r/s

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)

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

Example Material

https://github.com/timquinlan/nginxplus_api_gw

Links

NGINX Material: https://github.com/timquinlan/nginx102

NGINX Plus Material: https://github.com/timquinlan/nginxplus_api_gw

NGINX Plus Trial: https://www.nginx.com/free-trial-request

NGINX Docs: https://nginx.org/en/docs

NGINX Plus Docs: https://docs.nginx.com/

NGINX Plus Features: https://www.nginx.com/products/nginx/compare-models

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

NGINX Modules: https://www.nginx.com/resources/wiki/modules/

NJS Examples: https://github.com/nginx/njs-examples

Contact:

Tim Quinlan

t.quinlan@f5.com

@trquacker