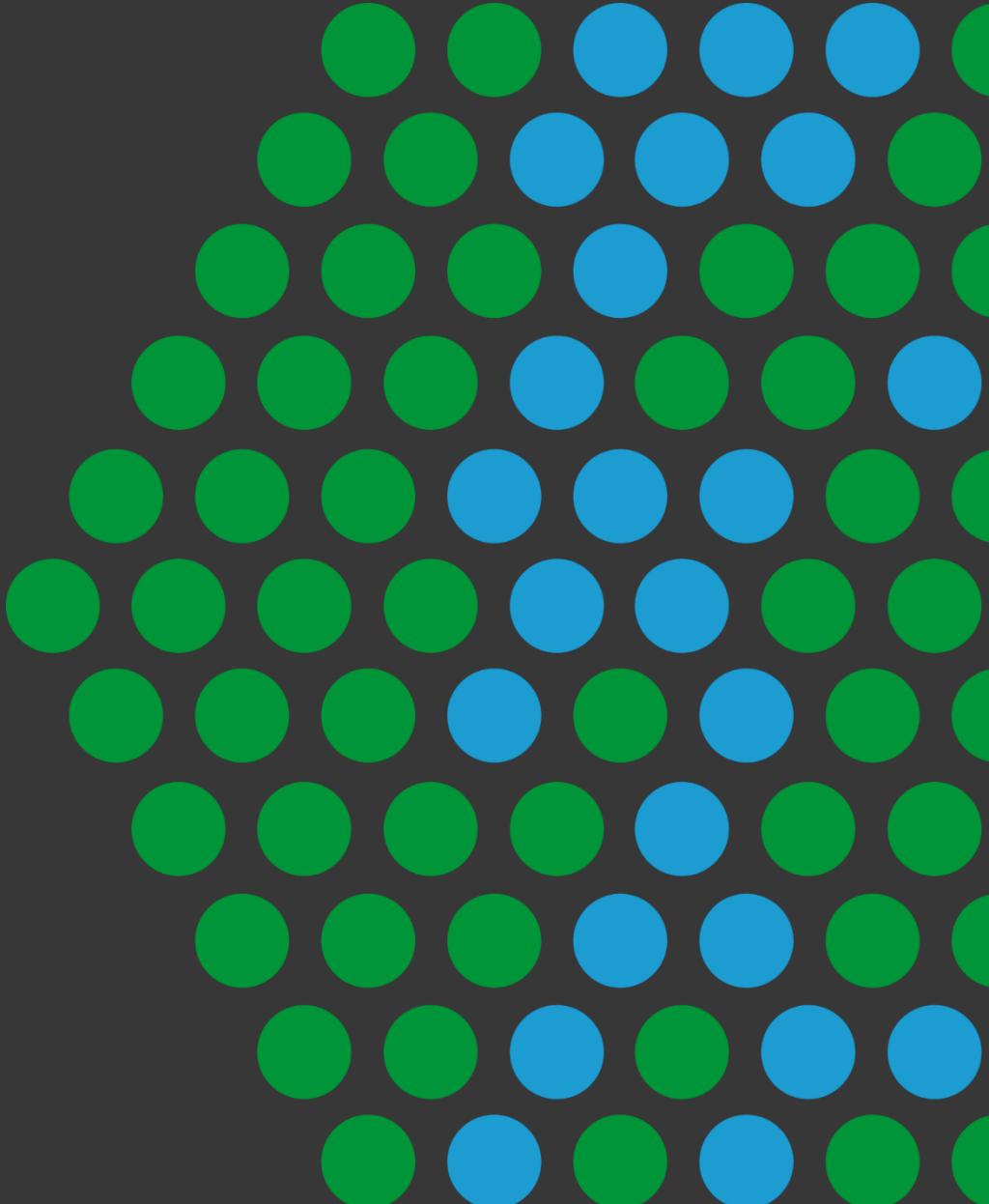




NGINX Application Platform

Jesse Goodier - NGINX Solutions Architect

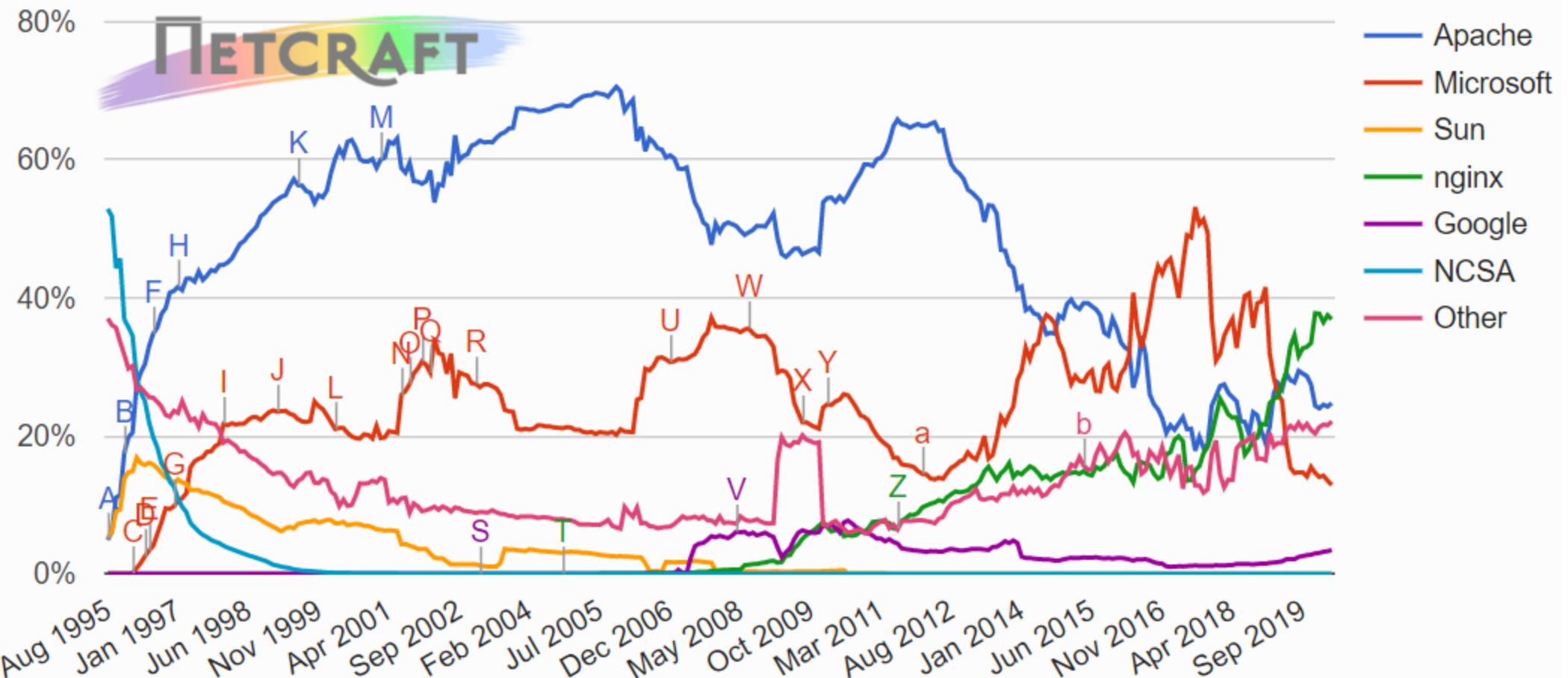
January 20, 2021



NGINX Background

NGINX

Web server developers: Market share of all sites



NGINX

2004: NGINX Open-Source released (C10K)

2011: NGINX Inc founded

2013: NGINX Plus released

2017: NGINX Controller released

2019: F5 acquires NGINX

2020: NGINX App Protect released (May 19)

2020: NGINX R22 (June 20)

2020: NGINX Controller 3.9 (Sept 26)

2020: NGINX Service Mesh .6 (October)

NGINX Application Platform

A suite of technologies to develop and deliver digital experiences that span from legacy, monolithic apps to modern, microservices apps.



NGINX Controller

Centralized monitoring and management



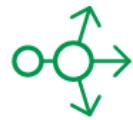
Analytics



Control



Policy



Load Balancer



Content Cache



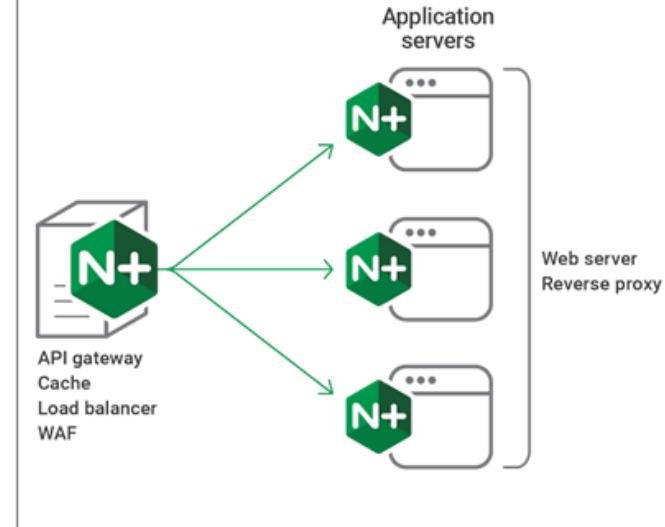
API Gateway



WAF



NGINX Plus
Application delivery



Cloud



Virtual Machine



Container



Bare Metal

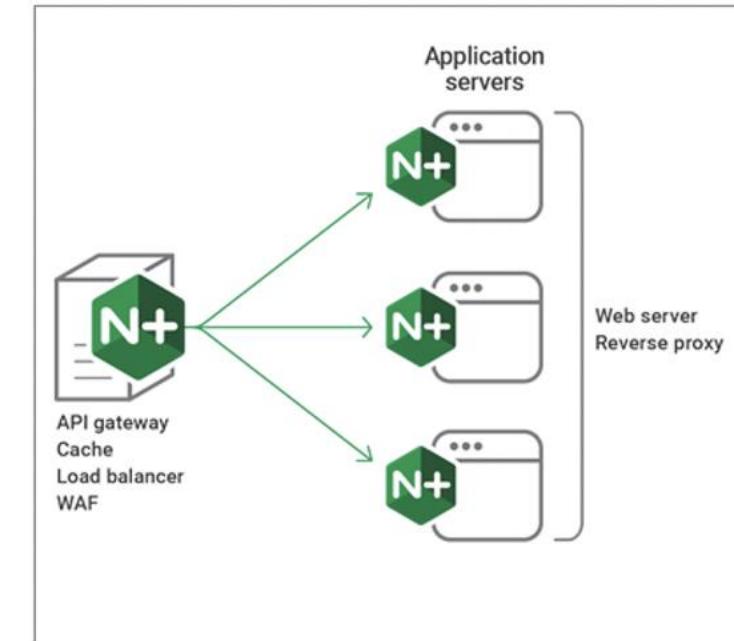
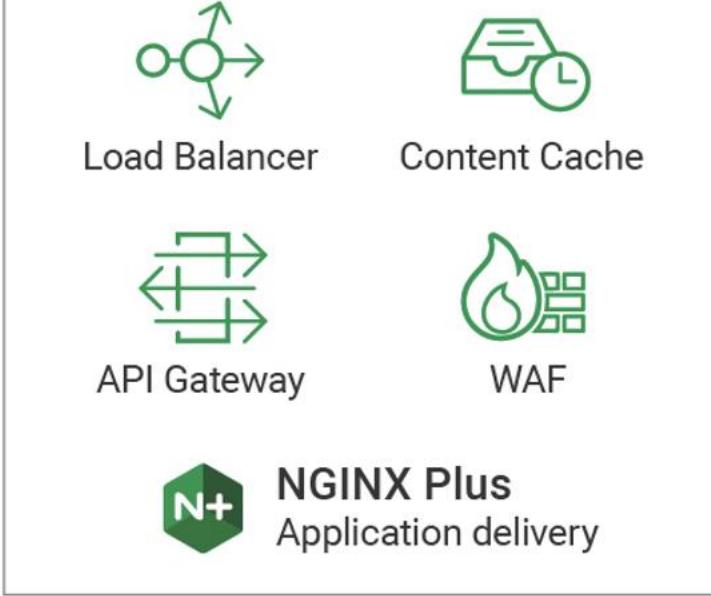
Infrastructure – Multi-cloud versatility



What is NGINX Plus?

ENTERPRISE SOLUTIONS WITH DYNAMIC MODULES

- Enterprise class visibility with 90+ additional metrics and live dashboard built-in
- WAF (OWASP top 10 and/or advanced protection)
- JWT Authentication (simple integration with okta/ping/etc)
- Native OpenID Connect support
- Active health checks on status code and response body
- Key value store (dynamic IP black-listing, blue/green deployments)
- High Availability / Zone Sync across cluster
- Dynamic reconfiguration—zero downtime
- Service discovery using DNS
- Sticky Session persistence based on cookies





Kubernetes Integration

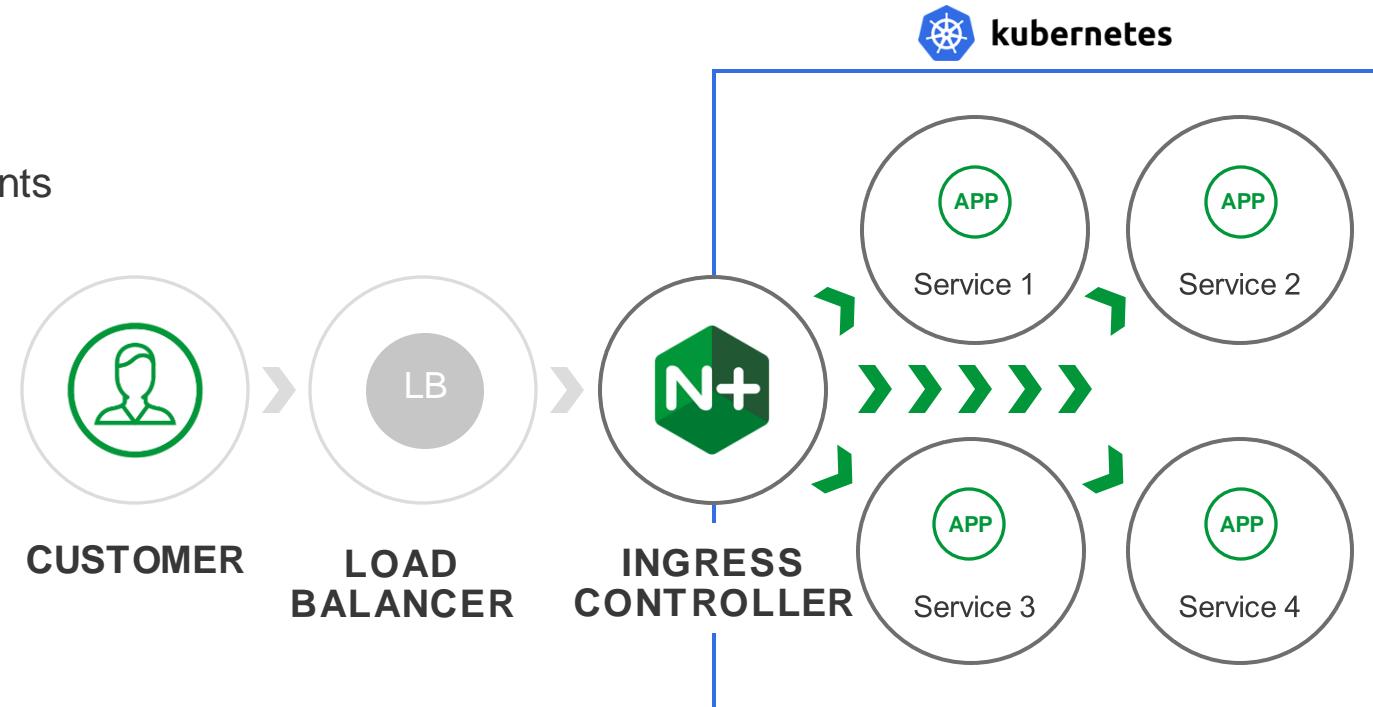
HIGH PERFORMANCE TRAFFIC MANAGEMENT FOR KUBERNETES

BENEFITS:

- Flexible app services for Kubernetes and OpenShift Router environments
- Consistency across pod deployments for rapid code to customer
- Scale apps with high performance

PLUS CAPABILITIES:

- Enterprise visibility with 90 additional metrics
- Live dashboard
- Session persistence
- JSON Web Token (JWT) authentication
- WAF with NGINX App Protect

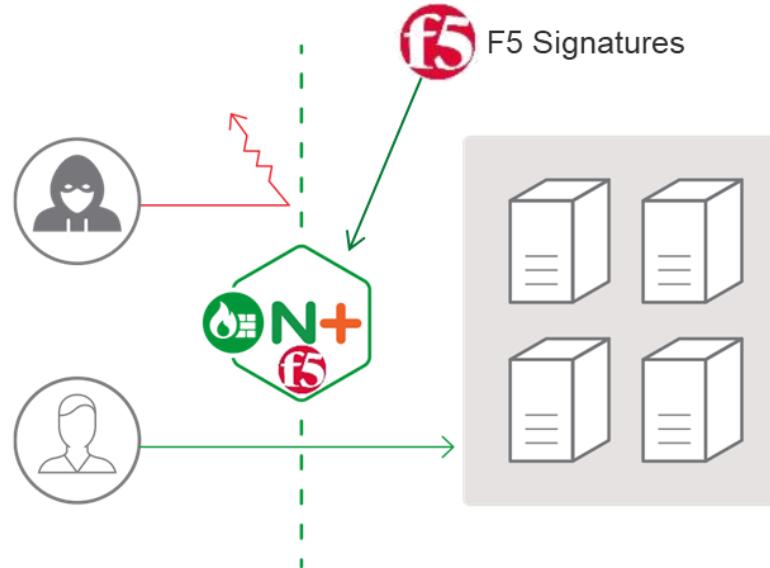


Available in Kubernetes and OpenShift



NGINX Technical Updates

NGINX App Protect



- ✓ High performing
- ✓ Security protection beyond signatures
- ✓ Trusted Signatures from F5



- ✓ Simple CI/CD integration
- ✓ Designed for modern infrastructures
- ✓ Rapid feedback loop for security remediations

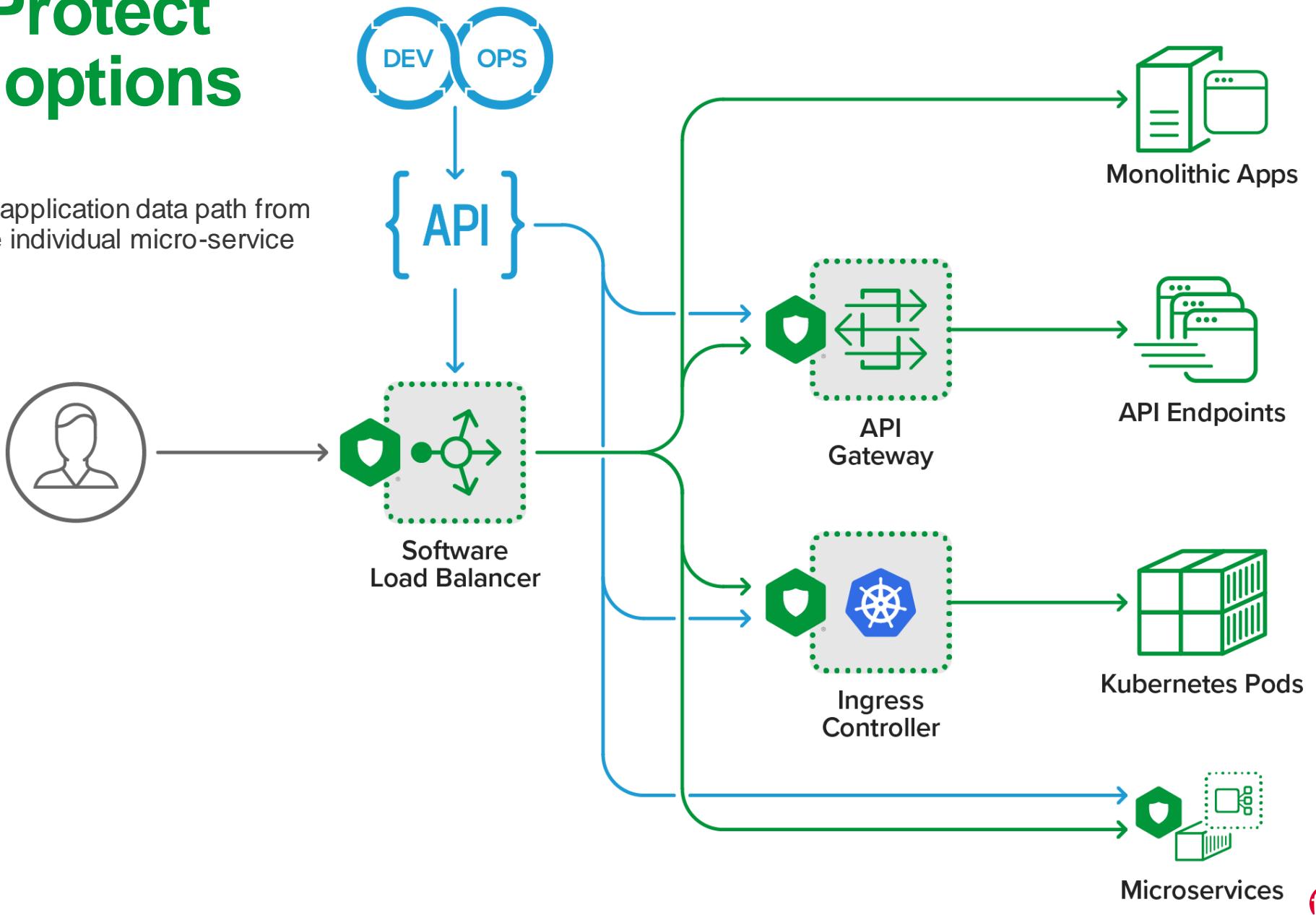


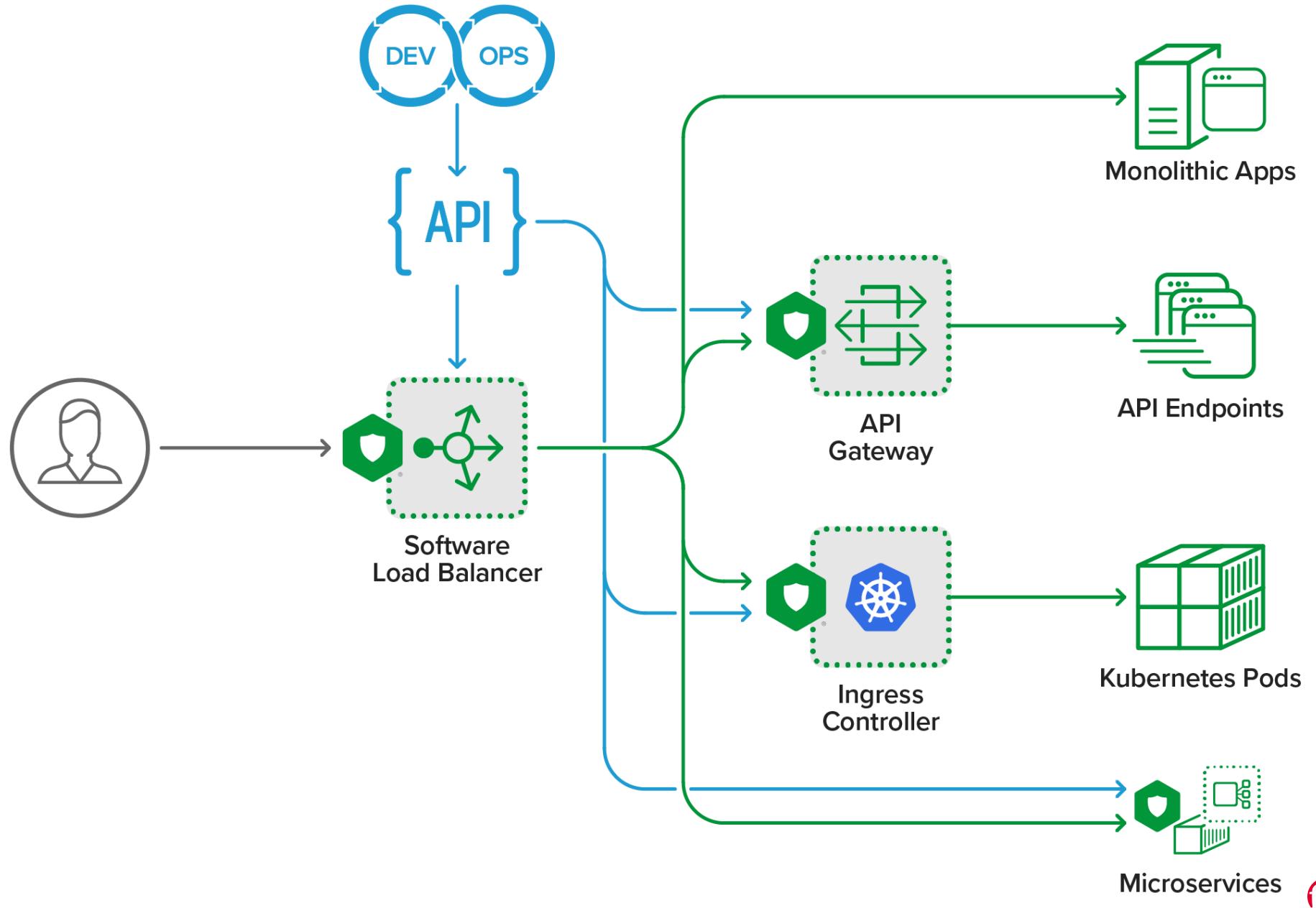
- ✓ Security as Code - Unified F5 declarative interface
- ✓ Security statistics via syslog
- ✓ Backed by F5 Support

NGINX App Protect Deployment options

Ultimate Flexibility

- Insert WAF at any point in the application data path from perimeter of the network to the individual micro-service

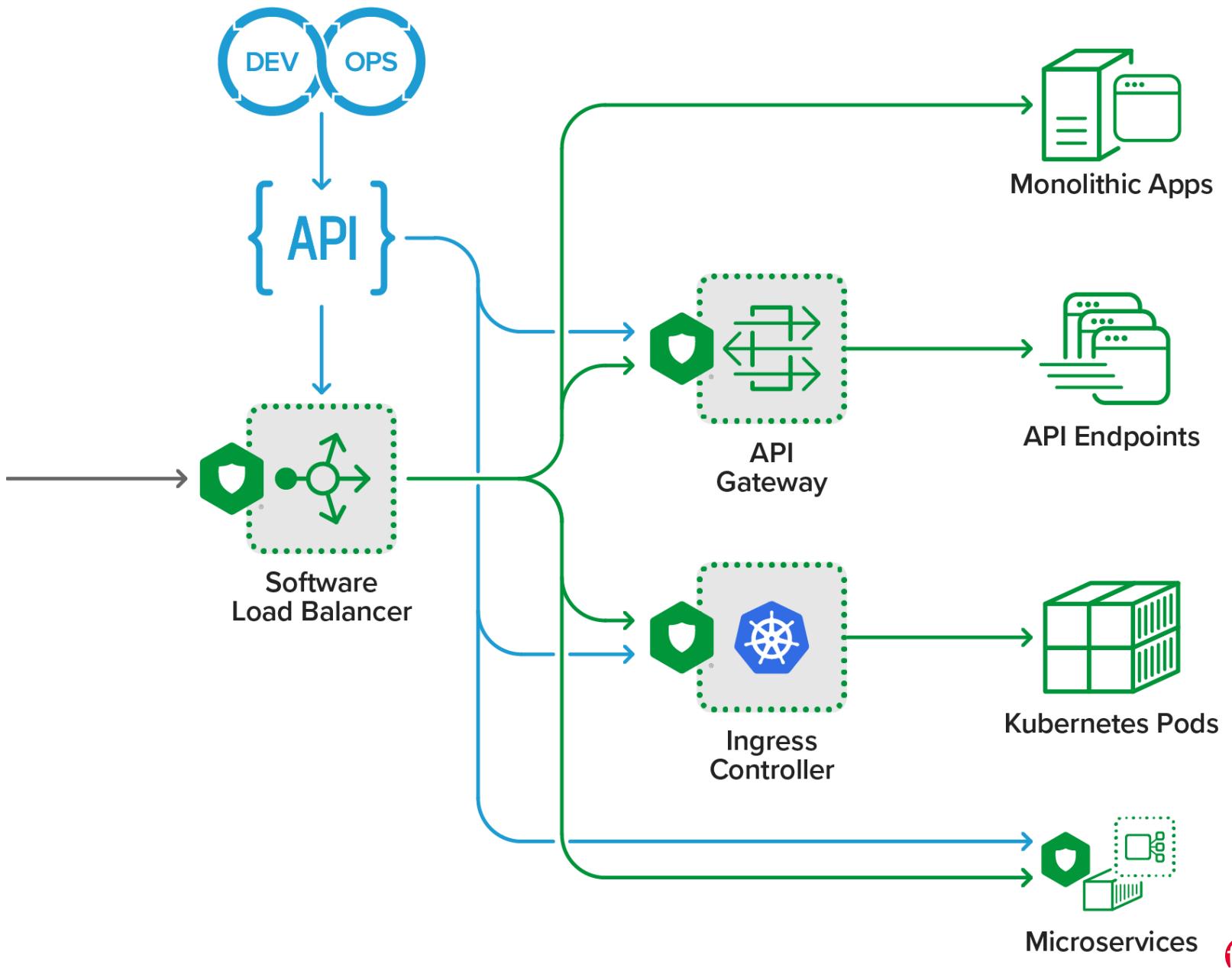


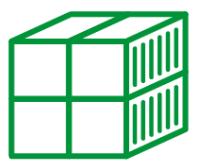




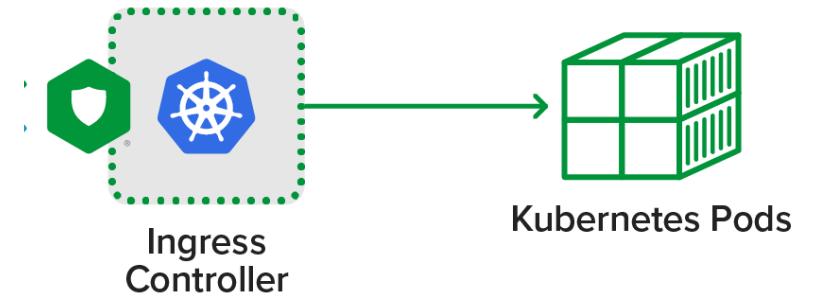
F5 BIG IP

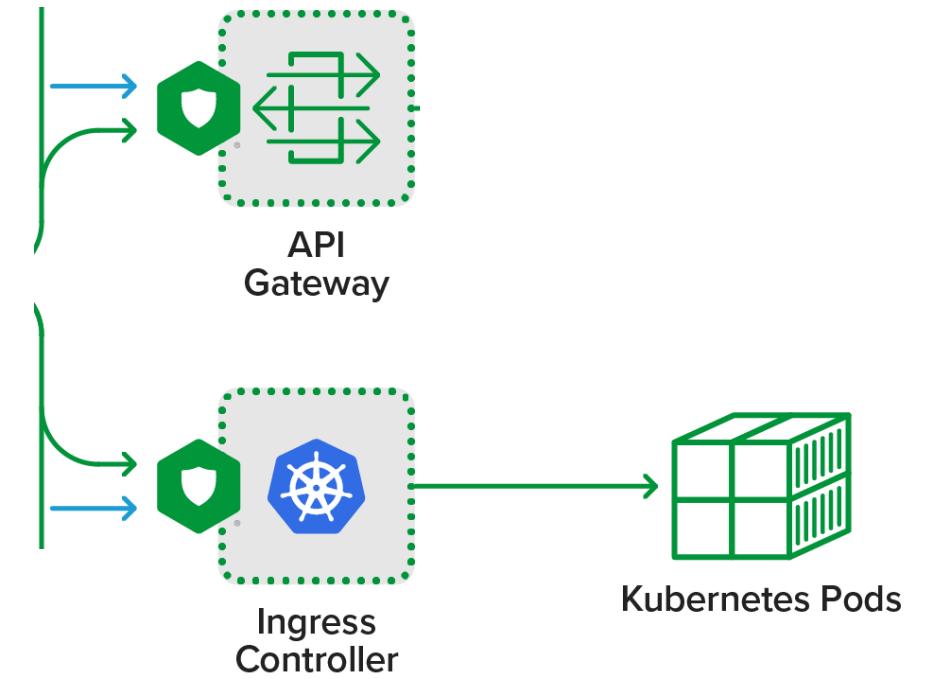
- Local L4-7 LB
- Global L4-7 LB
- SSL offload
- Advanced WAF
- Access mgmt.
- L4 firewall
- SSL orchestration
- Anti-DDoS
- Bot detection
- CGNAT
- Kubernetes CIS

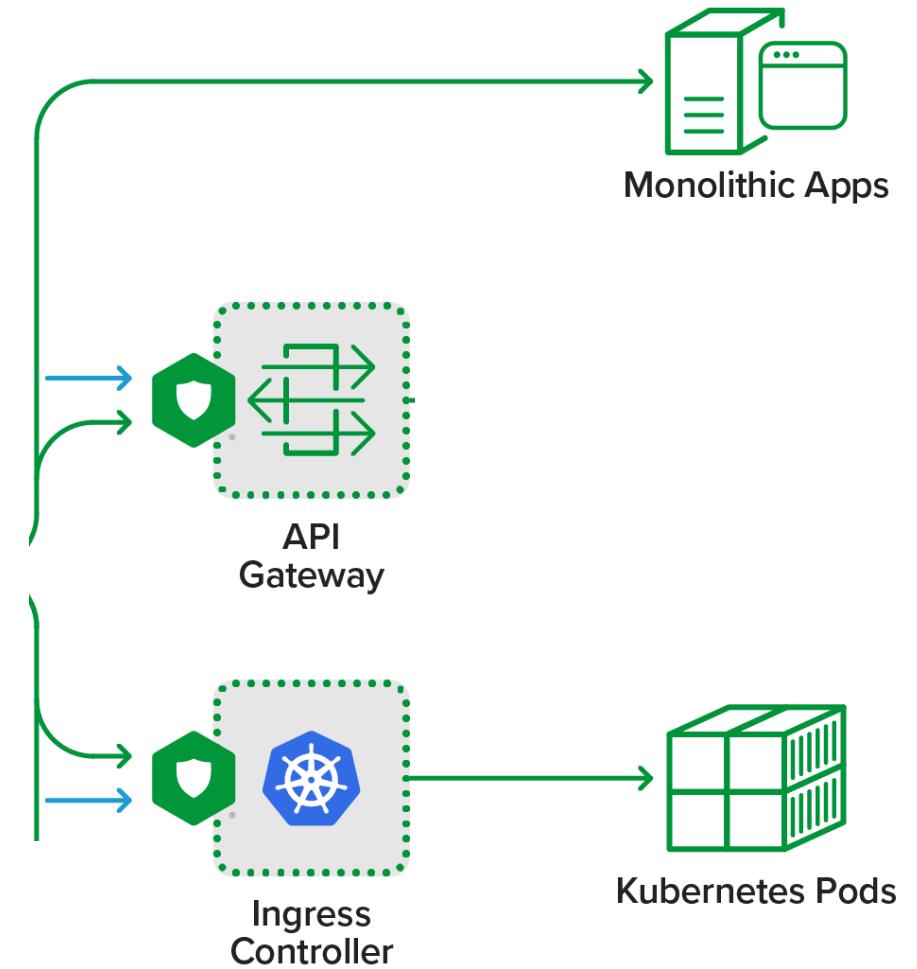


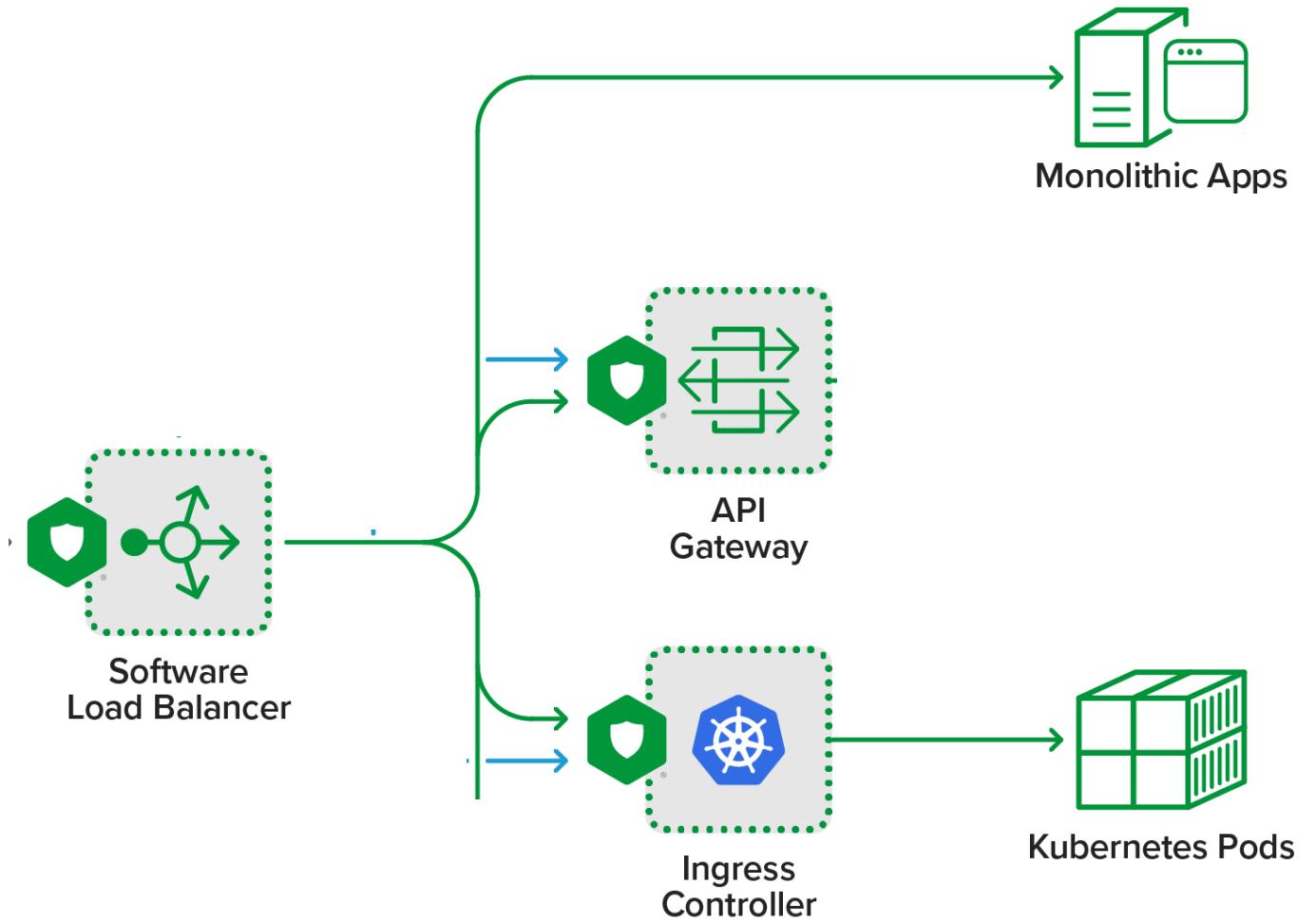


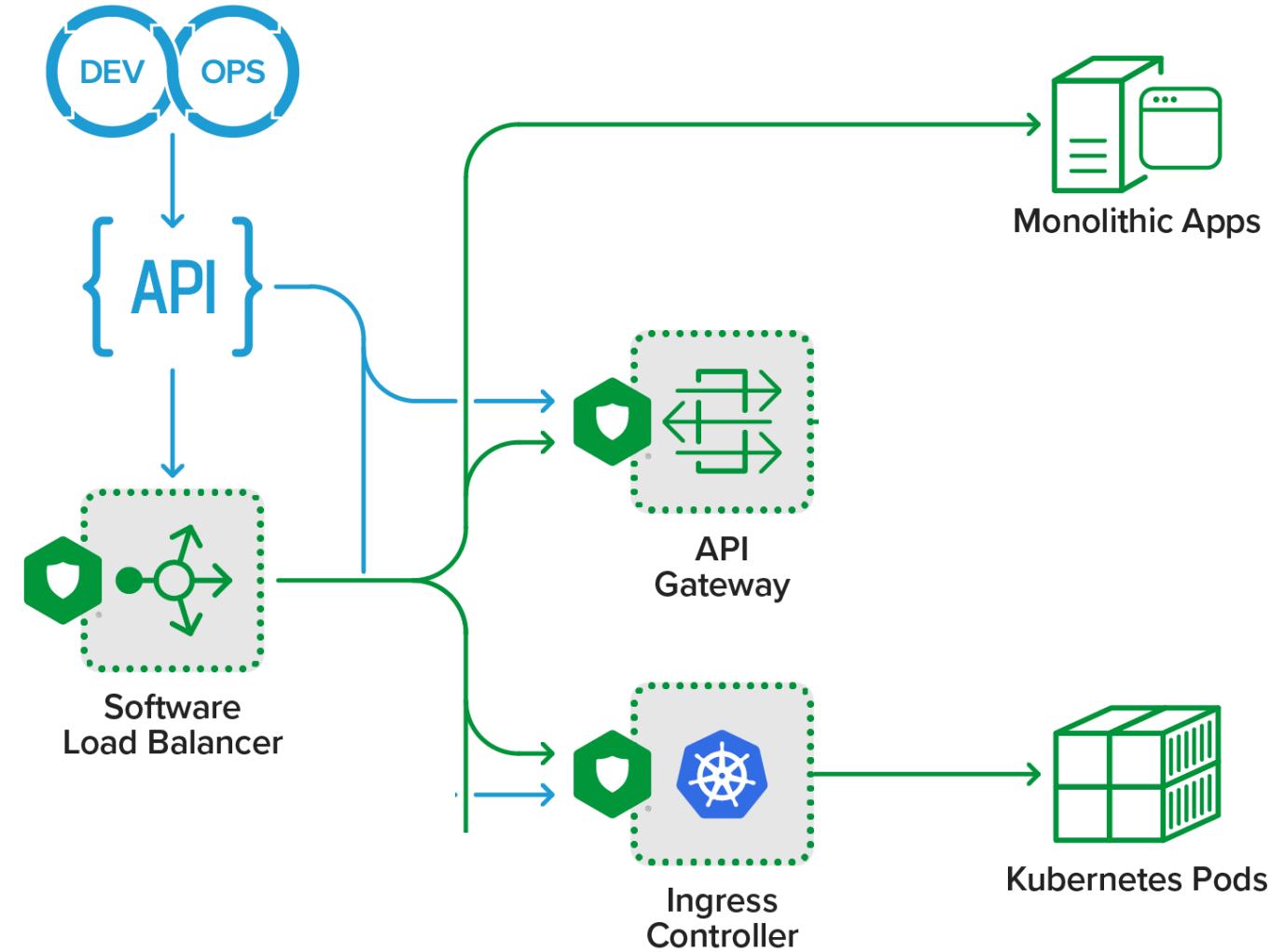
Kubernetes Pods









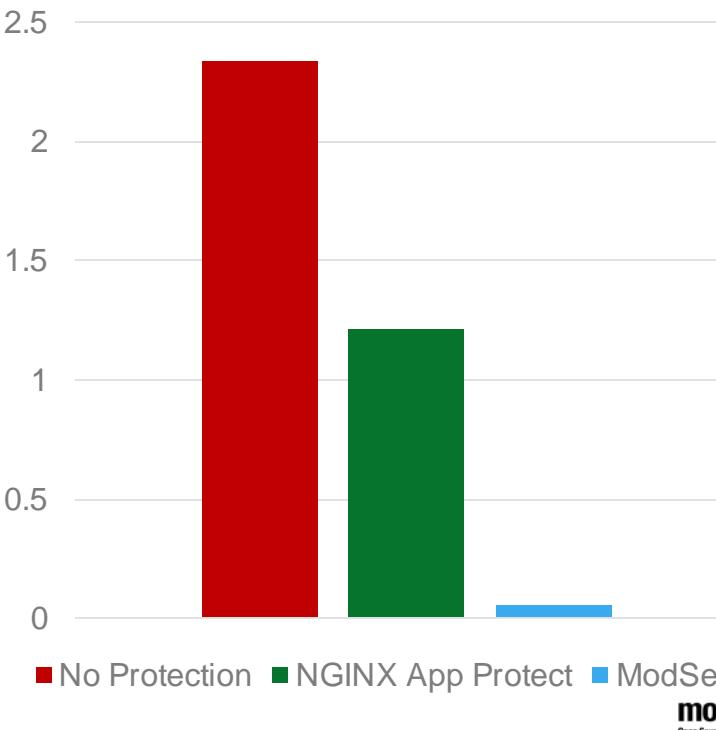


NGINX App Protect Performance

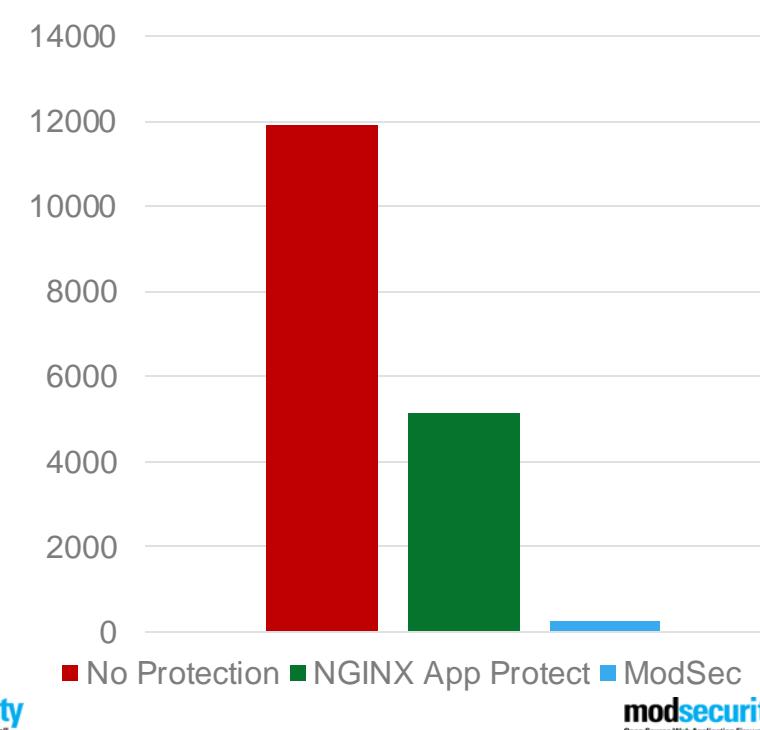
Comprehensive security policy has no impact on latency, and offers better throughput and requests/second when compared to ModSec

- ModSec Configuration: OWASP Top 10 (enable all CRS 3v rules)
- NGINX App Protect Configuration: OWASP Top 10 (Enable signatures), Evasion technique, Data Guard, Disallowed file types, HTTP protocol compliance

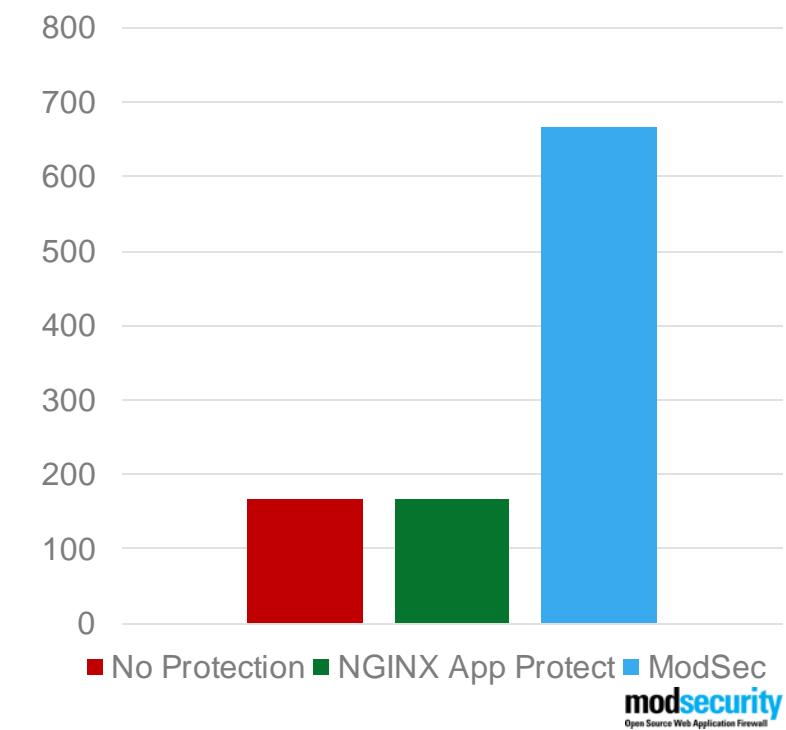
Throughput (MB/sec)



Requests/sec



Latency (ms)



Application Layer Threats Sales Positioning

	SELF-MANAGED	PARTIALLY-MANAGED (SaaS)	FULLY MANAGED W/SOC
Specialized Controls <small>FRAUD & ABUSE PREVENTION (HIGH VALUE B2C APPS)</small>			Shape Enterprise Defense
Tailored Controls <small>TARGETED ATTACKS AND ADVANCED THREAT ACTORS (INCLUDES FUNDAMENTALS)</small>	Advanced WAF (Hardware, VE, Cloud)	Future Shape Offer (Shape + Behavioral App Protect + Bot Protect)	Silverline WAF Shape Defense (available through Silverline)
Fundamental Controls <small>SOFTWARE VULNERABILITIES AND COMMON ATTACK VECTORS (E.G. "LONG TAIL APPS")</small>	NGINX App Protect	Essential App Protect	



NGINX Controller



Simplify Code to Customer Delivery

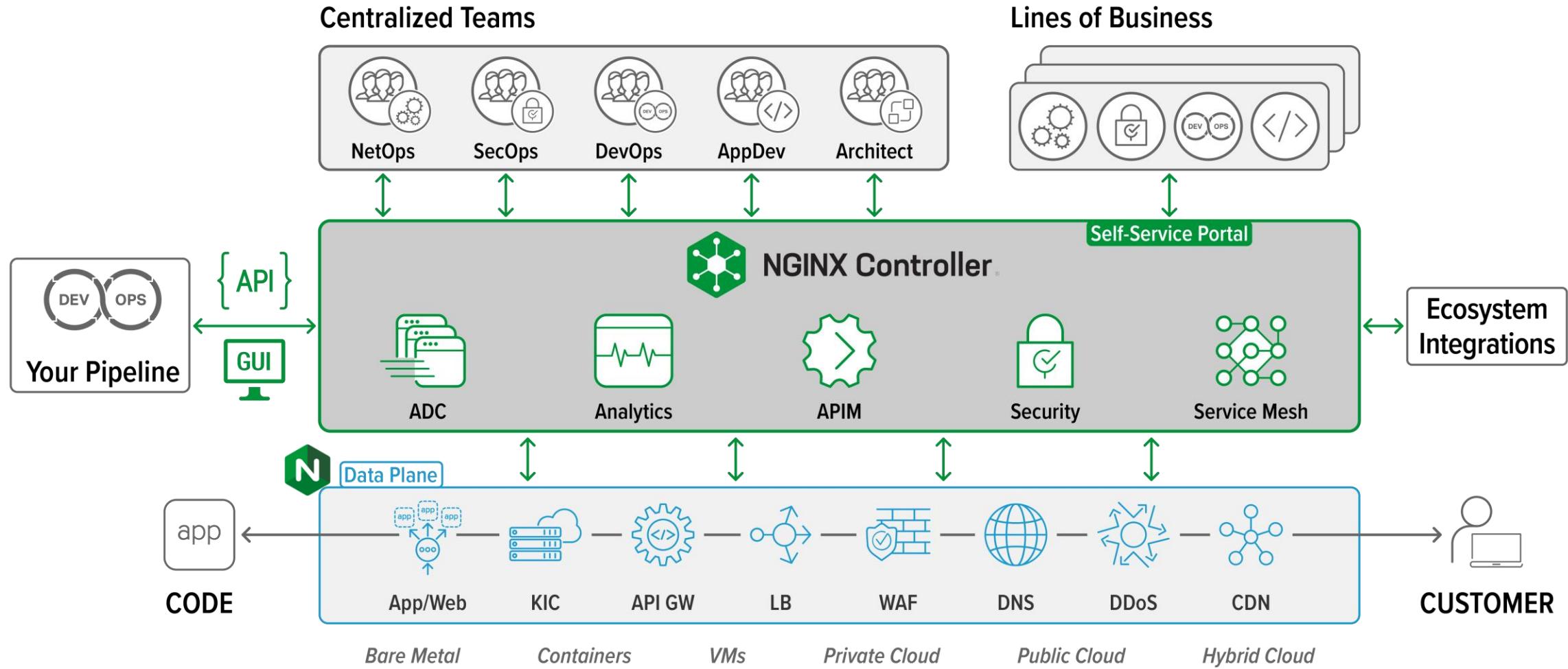


Respond with Intelligent Insights



Empower with Self-Service

NGINX Controller 3.0+ Built for Modern App Teams



NGINX Controller

Services

All Apps

Trading.ACMEFinancial.net

Overview

Components

ANALYTICS

Critical Analytics

Security Analytics

Security Events

RECENT APP COMPONENTS

Trading Referrals

Trading Transfers Component

Trading App - File Uploads

Trading API Secure

Trading API

SUMMARY METRICS LATENCY METRICS

HTTP 4XX Errors
624,168 ↑1.82% Compare: 613,011

HTTP 5XX Errors
69 ↓-41.53% Compare: 118

Total Requests
13,948,413 ↑0.93% Compare: 13,819,450

Bytes In
1.12 Gbps ↑0.95% Compare: 1.11

Bytes Out
692.31 Gbps ↑0.36% Compare: 689.8

Status: Configured

Last 6 hours

Compare to: Prev day

Go to Analytics

environments

gateways

apps

apis

certs

identity providers

all-trading.acmefinancial.net 404 400 499 all-trading.acmefinancial.net-cmp 400-cmp 404-cmp 499-cmp

The dashboard provides a comprehensive overview of the application's health and performance. Key metrics include:

- HTTP 4XX Errors:** 624,168 (↑1.82% Compare: 613,011)
- HTTP 5XX Errors:** 69 (↓-41.53% Compare: 118)
- Total Requests:** 13,948,413 (↑0.93% Compare: 13,819,450)
- Bytes In:** 1.12 Gbps (↑0.95% Compare: 1.11)
- Bytes Out:** 692.31 Gbps (↑0.36% Compare: 689.8)

The latency metrics section includes a line chart showing the volume of bytes transferred over time (Last 6 hours) for various components:

- all-trading.acmefinancial.net
- 404
- 400
- 499
- all-trading.acmefinancial.net-cmp
- 400-cmp
- 404-cmp
- 499-cmp

NGINX Controller

Services

All Apps

Trading.ACMEFinancial.net

Overview

Components

ANALYTICS

Critical Analytics

Security Analytics

Security Events

RECENT APP COMPONENTS

Trading Referrals

Trading Transfers Component

Trading App - File Uploads

Trading API Secure

Trading API

Environments

Gateways

Apps

APIs

Certs

Identity Providers

Security Summary

Last 6 hours Compare to: Prev day

WAF Rejected HTTP Requests
238,989 ↑ 2.25% Compare: 233,725 undefined

WAF Flagged HTTP Requests
34,350 ↑ 2.55% Compare: 33,497 undefined

WAF Rejected Bytes In
32.32 MBps ↑ 2.04% Compare: 31.67 MBps

WAF METRICS

WAF Requests Rejected ↑ 2.15%

10 AM 11 AM 12 PM 01 PM 02 PM 03 PM

SECURITY_WAF_VIOLATION SECURITY_WAF_VIOLATION-cmp

NGINX Controller

Services

All Apps

Trading.ACMEFinancial.net

Overview

Components

Environments

ANALYTICS

Critical Analytics

Security Analytics

Security Events

Gateways

RECENT APP COMPONENTS

Trading Referrals

Trading Transfers Component

Trading App - File Uploads

Trading API Secure

Trading API

APIs

Certs

Identity Providers

Security Events

Show Filters

Last 24 hours

Refresh

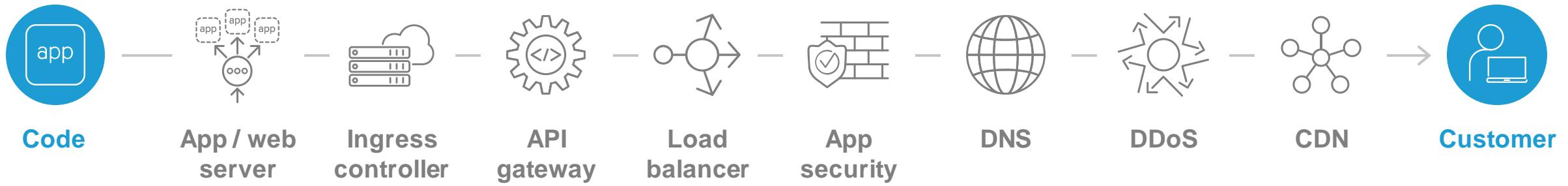
Filter

Time	Component	Remote Address	Outcome	Reason	Attack Type
2020-08-27T15:20:10Z	trading-main-cas	10.1.20.23	! REJECTED	SECURITY_WAF_VIOLATION	Non-browser Client, Cross Site Scripting (XSS)
2020-08-27T15:20:10Z	trading-main-cas	10.1.20.23	! REJECTED	SECURITY_WAF_VIOLATION	Non-browser Client, Abuse of Functionality, Cross Site Scripting (XSS)
2020-08-27T15:20:10Z	trading-main-cas	10.1.20.23	! REJECTED	SECURITY_WAF_VIOLATION	Non-browser Client, Cross Site Scripting (XSS)
2020-08-27T15:20:10Z	trading-main-cas	10.1.20.23	✓ PASSED	SECURITY_WAF_FLAGGED	SQL-Injection
2020-08-27T15:20:10Z	trading-main-cas	10.1.20.23	! REJECTED	SECURITY_WAF_VIOLATION	Non-browser Client, Cross Site Scripting (XSS)
2020-08-27T15:20:10Z	trading-main-cas	10.1.20.23	! REJECTED	SECURITY_WAF_VIOLATION	Non-browser Client, Cross Site Scripting (XSS)
2020-08-27T15:20:10Z	trading-main-cas	10.1.20.23	! REJECTED	SECURITY_WAF_VIOLATION	Non-browser Client, Abuse of Functionality, Cross Site Scripting (XSS)
2020-08-27T15:20:10Z	trading-main-cas	10.1.20.23	! REJECTED	SECURITY_WAF_VIOLATION	Non-browser Client, Cross Site Scripting (XSS)

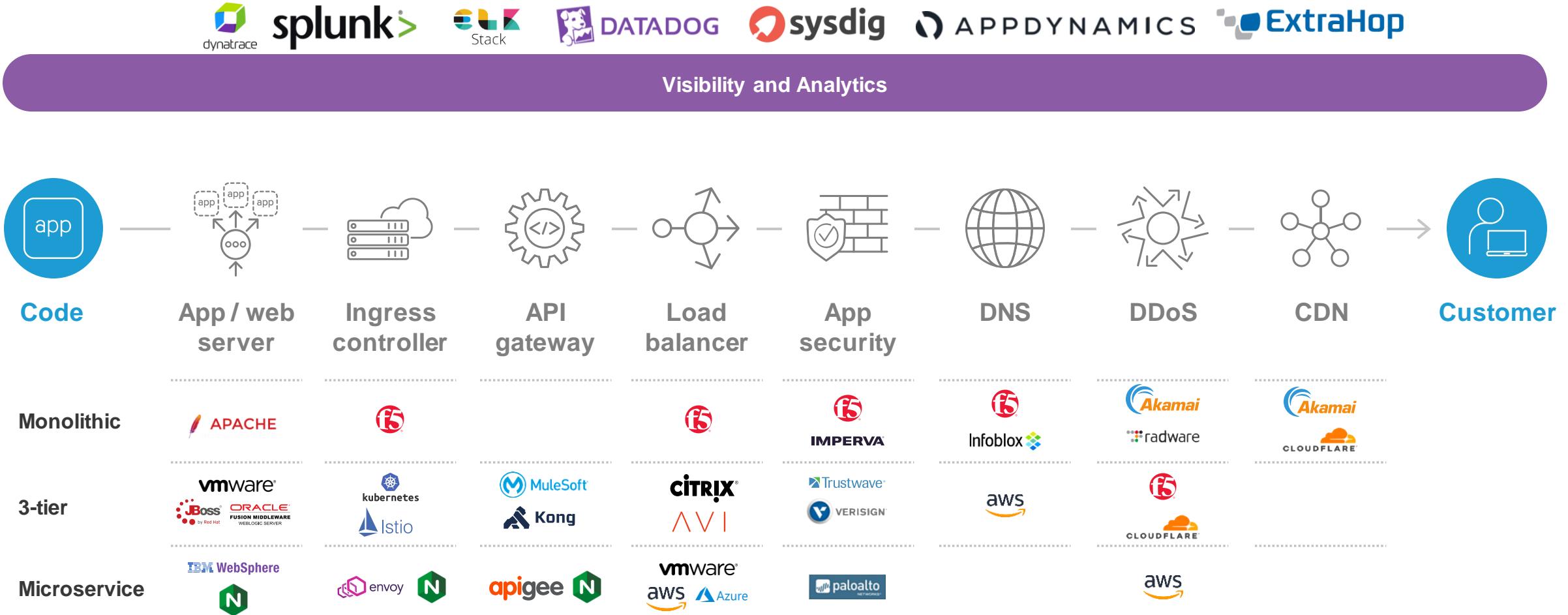
Code To Customer

F5 and NGINX

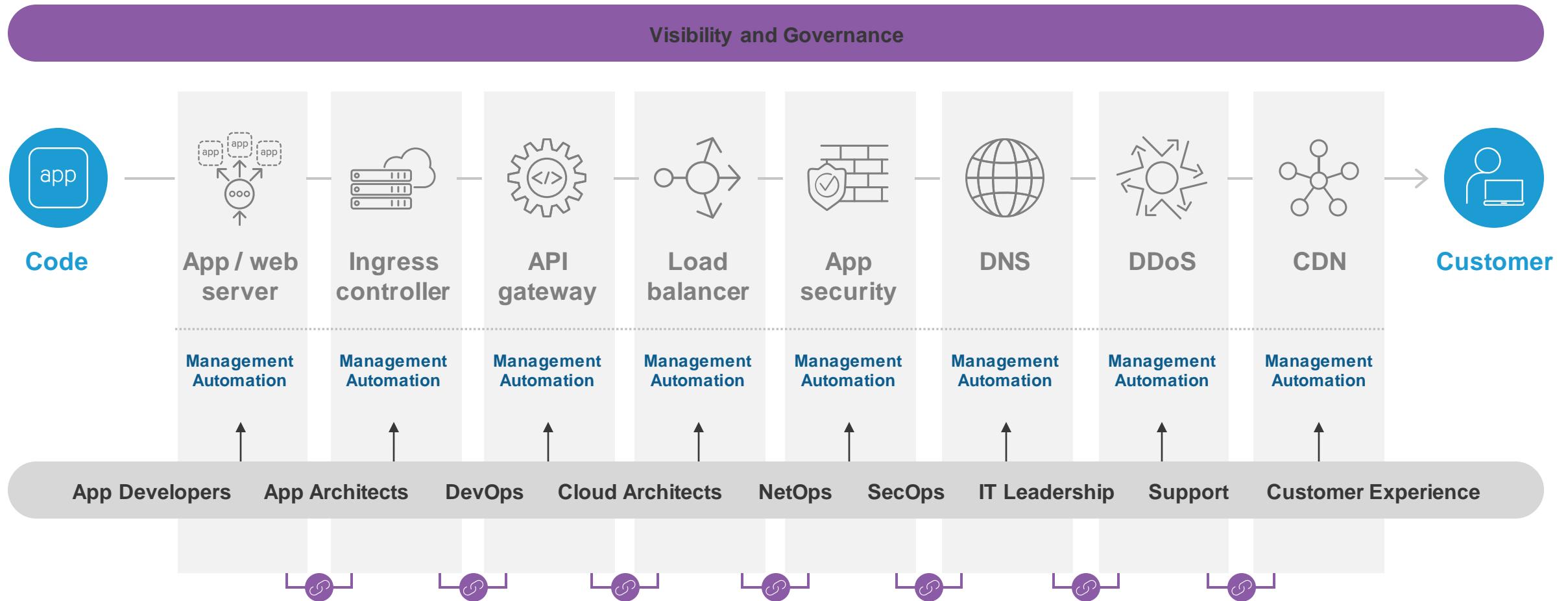
Customers have several services along the application data path



Different vendors for each application architecture

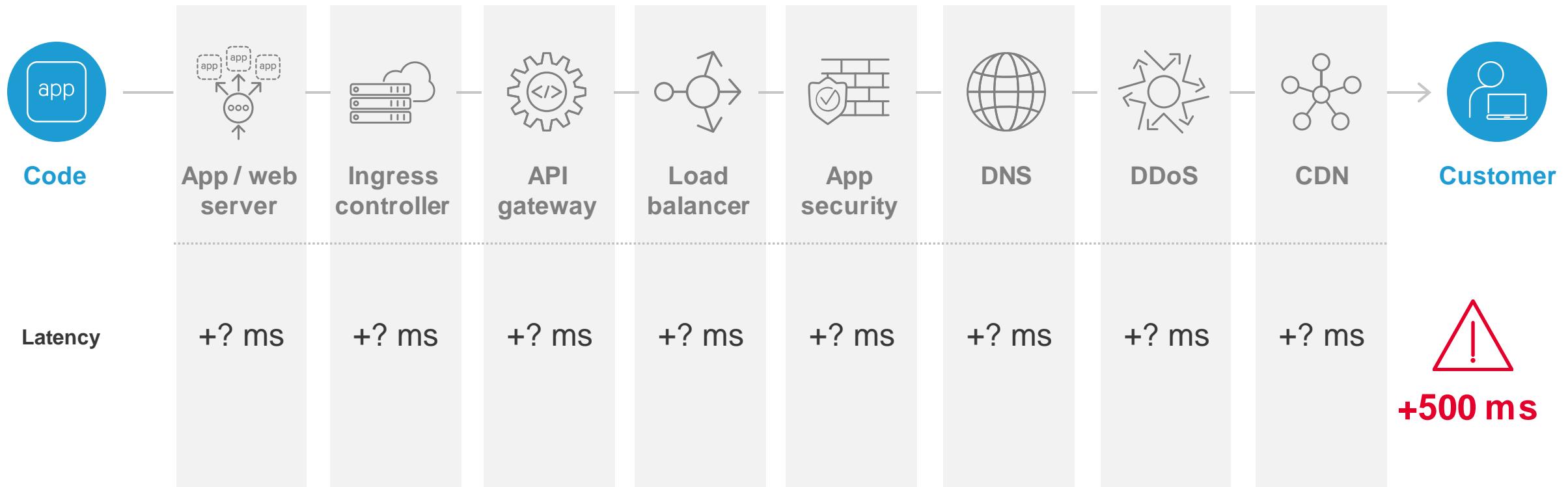


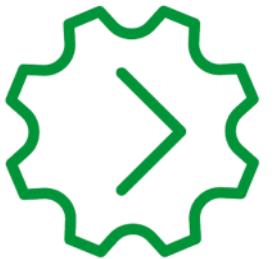
Limited orchestration across the data path



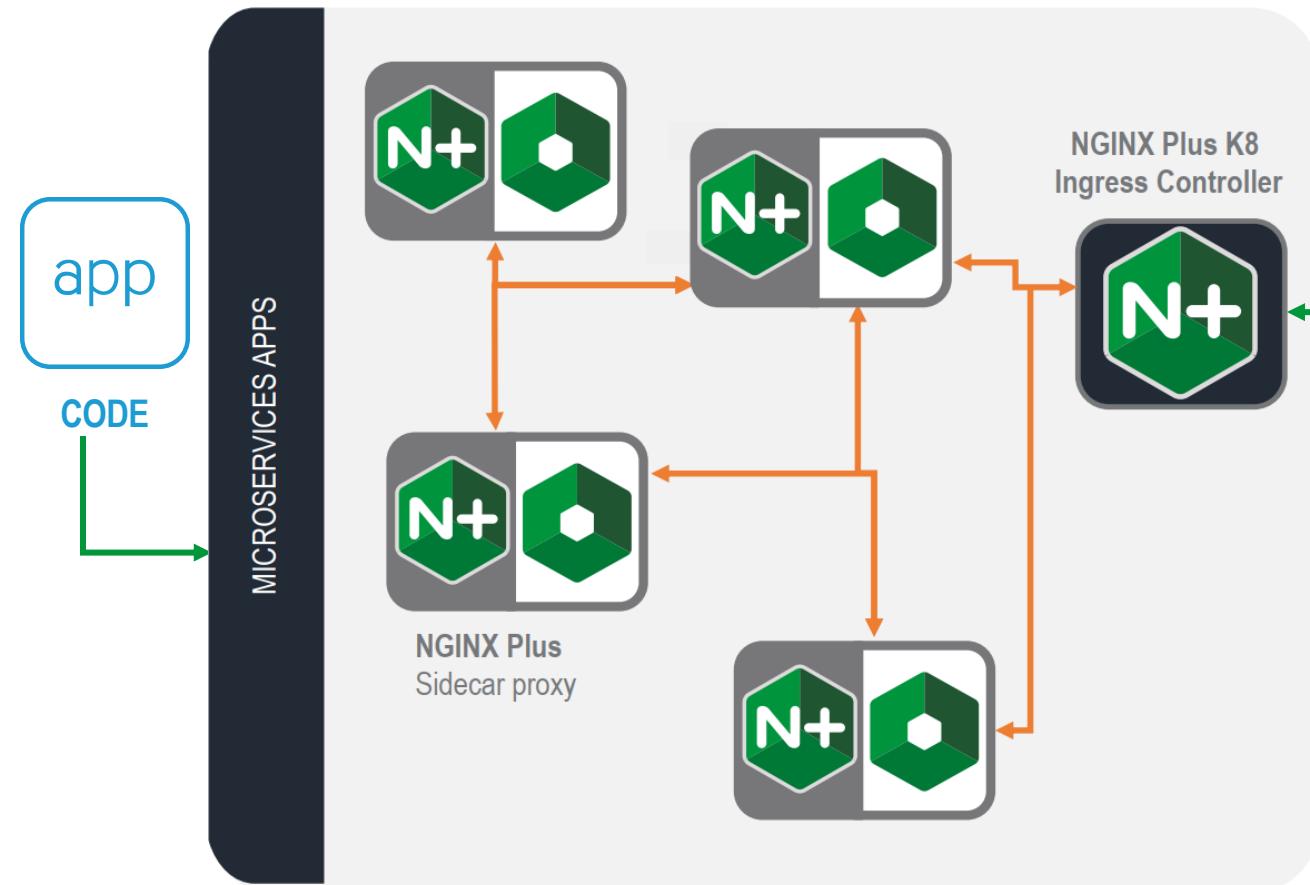
And unable to easily pinpoint issues

AND UNABLE TO EASILY PINPOINT ISSUES





Modern Application Architecture



30

Scale ADCs Across Multi-Cloud

Consistent, Secure and Portable Apps

Perimeter



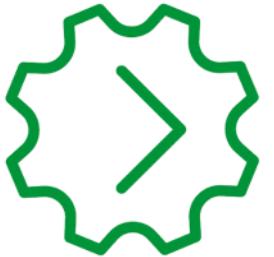
NGINX Plus

- Reverse proxy
- L4-7 LB
- API gateway
- Per-app WAF

F5 BIG IP

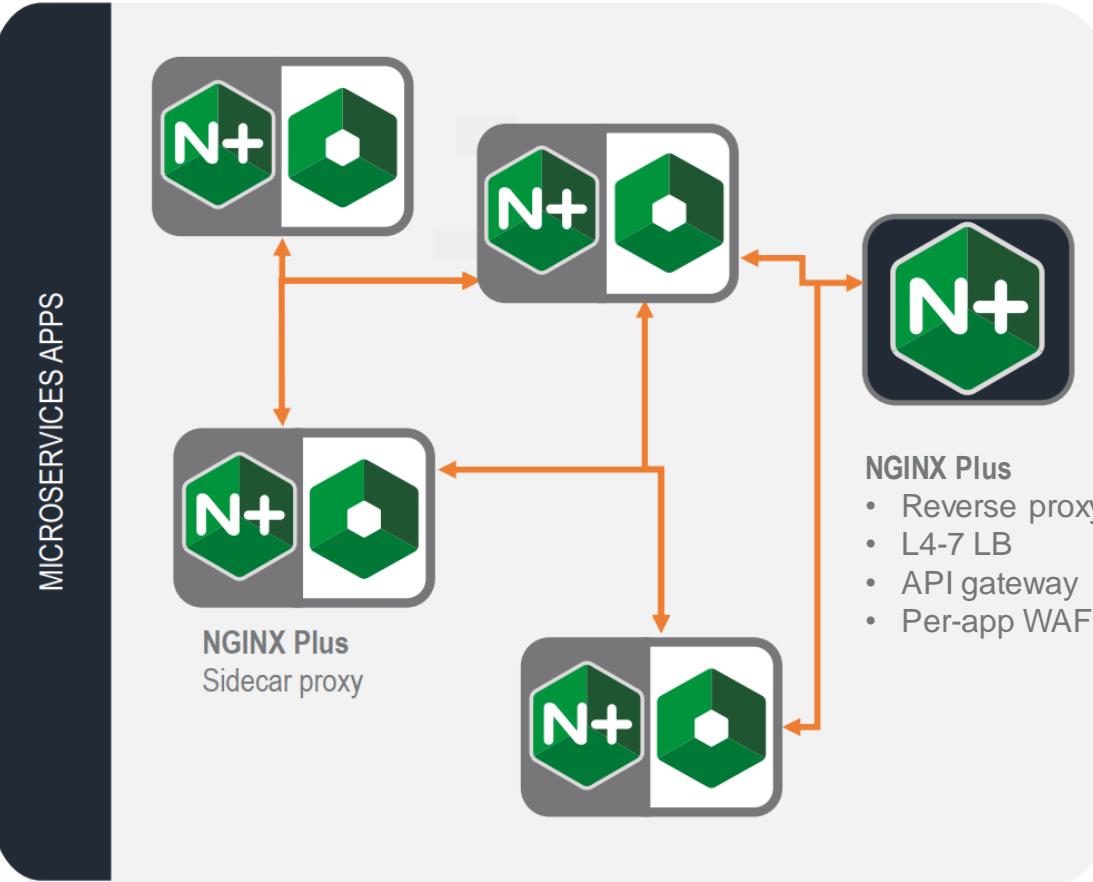
- Local L4-7 LB
- Global L4-7 LB
- SSL offload
- Advanced WAF
- Access mgmt.
- L4 firewall
- SSL orchestration
- Anti-DDoS
- Bot detection
- CGNAT





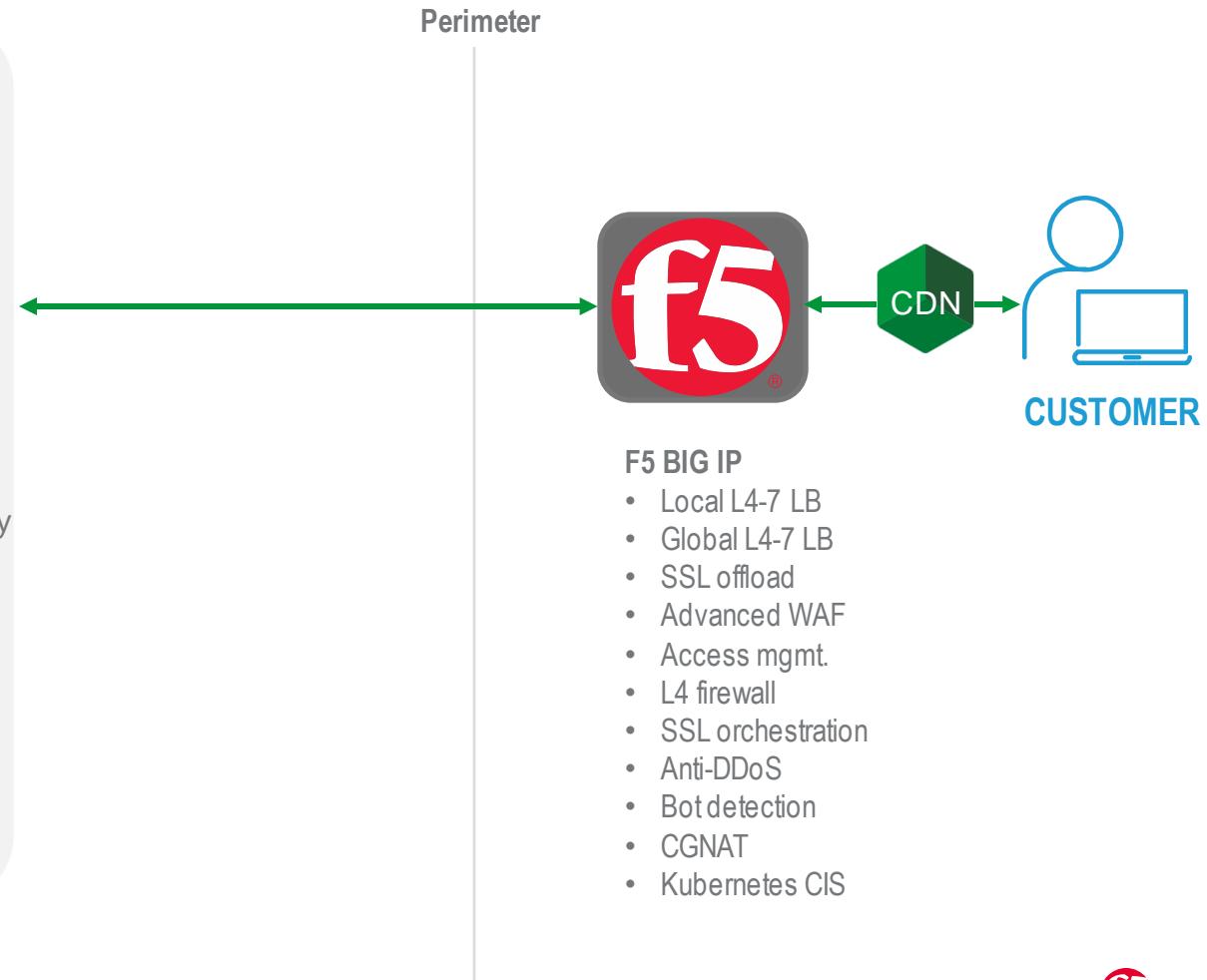
Modern Application Architecture

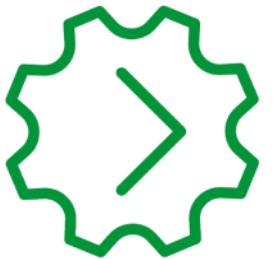
app
CODE



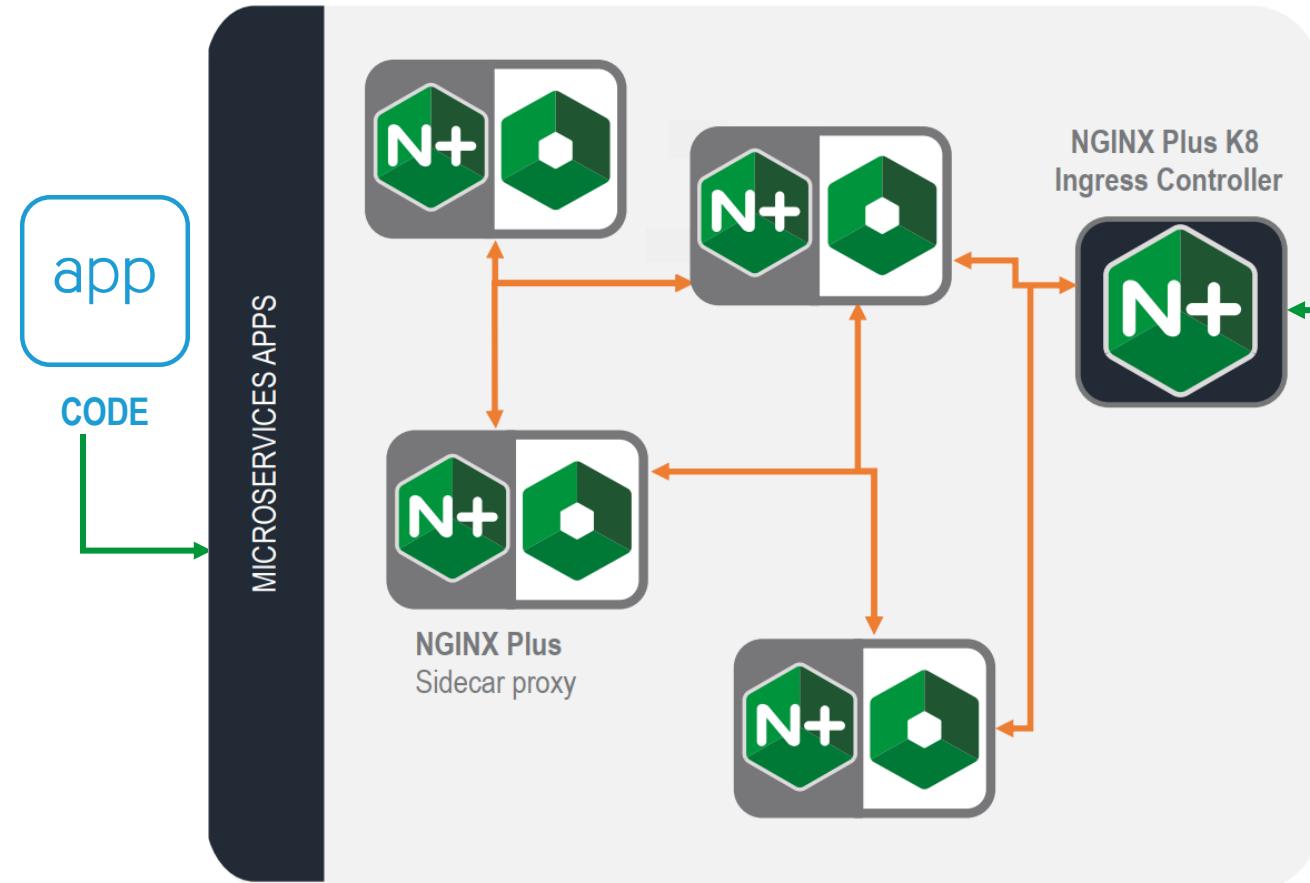
Scale ADCs Across Multi-Cloud

Consistent, Secure and Portable Apps





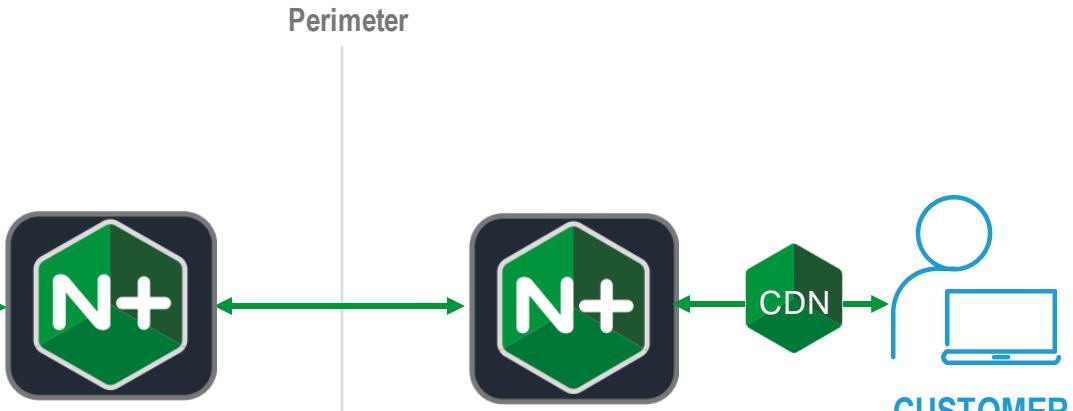
Modern Application Architecture



Scale ADCs Across Multi-Cloud

Consistent, Secure and Portable Apps

Perimeter

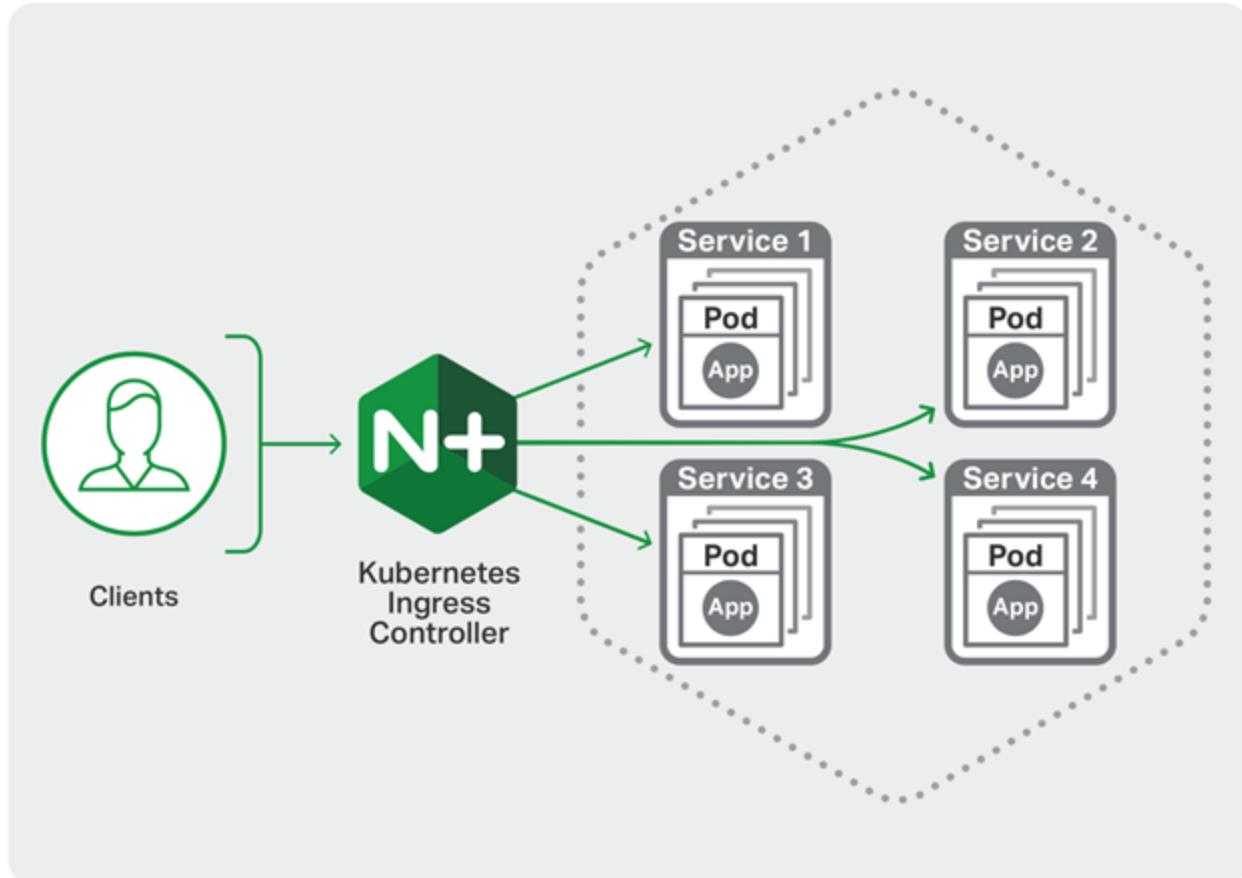


NGINX Plus

- Reverse proxy
- L4-7 LB
- API gateway
- Per-app WAF

Kubernetes Ingress Controller

An advanced Layer 7 load-balancing solution for exposing Kubernetes services to the Internet



- NGINX commonly used as Ingress Controller
- Dynamic reconfiguration of endpoints (no configuration reloading)
- Additional metrics, provided by a streamlined Prometheus exporter
- Dedicated Helm chart repository
- Support for Custom resources to expose more (all) NGINX Plus features as an Ingress

Kubernetes Ingress Controllers

The Key Differences

The table below summarizes the key difference between nginxinc/kubernetes-ingress and kubernetes/ingress-nginx Ingress controllers. Note that the table has two columns for the nginxinc/kubernetes-ingress Ingress controller, as it can be used both with NGINX and NGINX Plus. For more information about nginxinc/kubernetes-ingress with NGINX Plus, read [here](#).

Aspect or Feature	kubernetes/ingress-nginx	nginxinc/kubernetes-ingress with NGINX	nginxinc/kubernetes-ingress with NGINX Plus
Fundamental			
Authors	Kubernetes community	NGINX Inc and community	NGINX Inc and community
NGINX version	Custom NGINX build that includes several third-party modules	NGINX official mainline build	NGINX Plus

<https://github.com/nginxinc/kubernetes-ingress/blob/master/docs/nginx-ingress-controllers.md>

What Is A Service Mesh?

WHAT'S MISSING IN K8S AND WHAT DO YOU REALLY WANT AND NEED FROM A MESH?

Service mesh aims to improve application traffic control, observability and security for distributed systems.

- *The New Stack*

What Is A Service Mesh?

WHAT'S MISSING IN K8S AND WHAT DO YOU REALLY WANT AND NEED FROM A MESH?

- A service mesh adds **L7 traffic management & security**:
 - sidecar deployment
 - policy management
 - application availability/health,
- Service mesh isn't just one "thing", it's a lot of managed and dependent components
- Takes over where K8s networking stops (service/pod IP endpoints)
- ***Traffic management for containers***

What Does A Service Mesh Do?

Service Mesh controls communications between pods and external apps



Secure Traffic

End-to-end encryption (Mutual TLS / mTLS), ACLs



Orchestration

Injection and sidecar management, K8s API integration



Manage All Service Traffic

Load Balance, Circuit breaker, B|G, Rate Limiting...



Measure Traffic

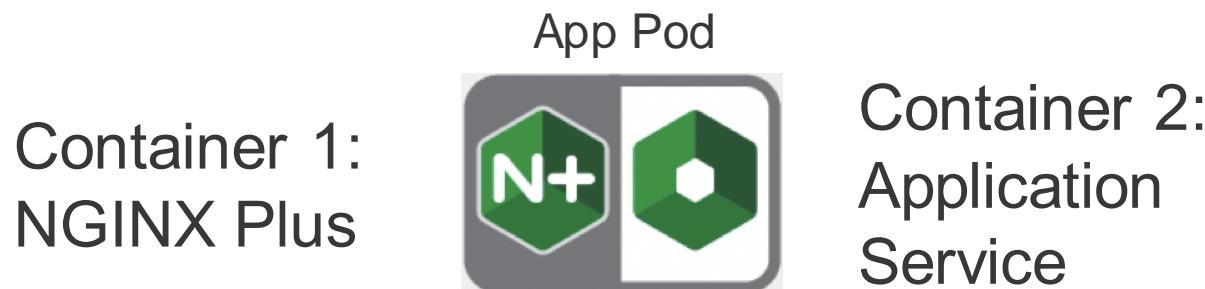
Generate transaction traces and real-time monitoring

What Is A Sidecar?

A Sidecar is a containerized service that another containerized service depends on for some function: “Helper Containers”

- Not just networking, can be used for any separation of process: API GW, logging, data mining, etc.

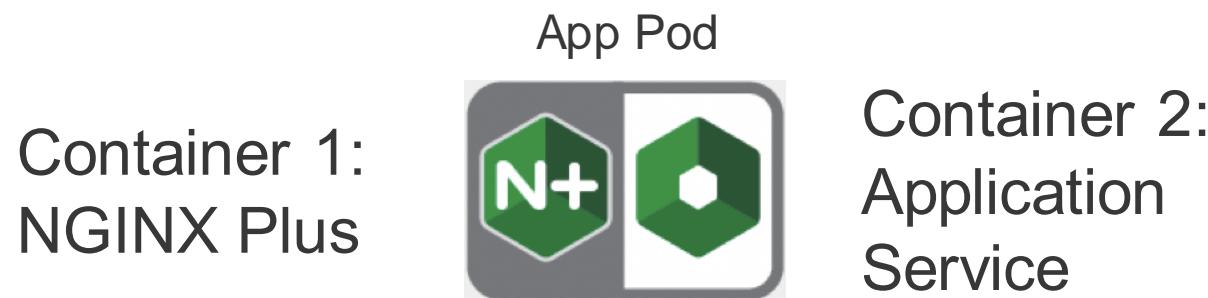
In our world, a Sidecar would be a reverse proxy that sits beside an application service container (in the same pod) and provides all inbound and outbound network routing to that application container



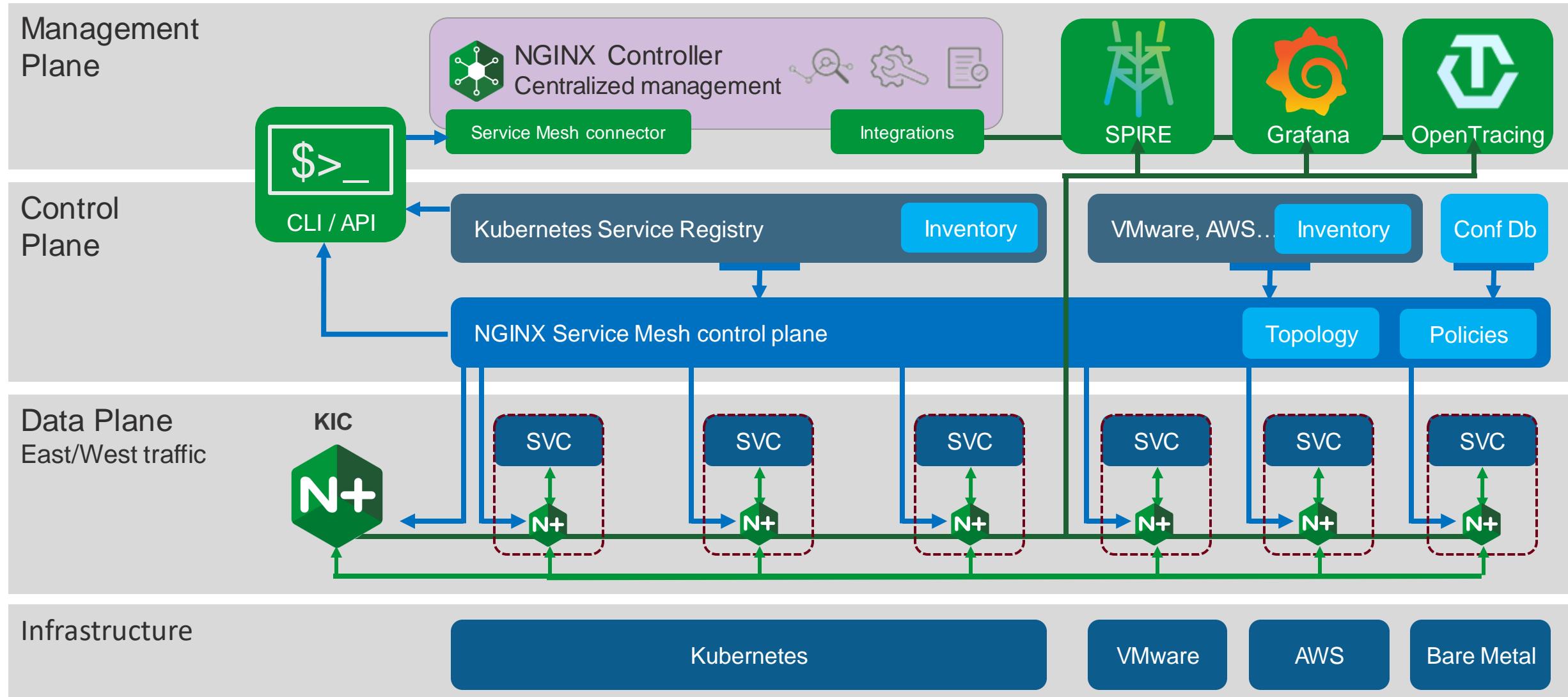
How Are Sidecars Deployed?

Separate Container In The App Pod

- The separate container is attached to the app service container in a pod
- Networking in the app container is altered via a policy from the mesh that tells the app “You can only talk to your sidecar for network access.”
- Policy and architecture are defined and orchestrated via the control plane, managed with a combo of ConfigMap and control plane.
- A Service Mesh takes care of auto-associating the sidecar with the app container in the same pod via ***Sidecar Injection***



What is the NGINX Service Mesh?





Customer Workshops

+

SE Technical Training



Security

API Traffic Visibility

Expose web & API traffic with Big-IP & NGINX Plus and act upon it to provide better business visibility, threat protection, and improve the customer experience.

Per App WAF

Learn the basics of configuring WAF for Big-IP and configuring the software-based WAF from NGINX Plus.

App Security Workshop

Learn JWT authentication (AuthZ & AuthA) for Big-IP & NGINX Plus.

API Security Workshop

Learn how to mitigate common API attacks using BIG-IP AWF & NGINX Plus/Controller

Kubernetes Security 101

Learn how to protect your k8s environment using encryption, token authentication, per application WAF and mitigate DOS attacks.

Building Your Own CDN

Learn how a content delivery network works, and when it makes sense to build your own CDN. Get a technical deep dive on building a CDN with NGINX Plus.

Networking

Cisco ACI Workshop

Gain hands on experience on managing the BIG-IP & NGINX Plus instances in your ACI environment for standard & Kubernetes workloads

Automation & Orchestration

Learn how to use F5 and industry standard infrastructure automation tools to streamline your workflows

Kubernetes Networking 101

Learn basics and best practices with Big-IP CIS and N+ KIC.

Monitoring & Analytics

Learn how to integrate/export Big-IP & NGINX Plus metrics to your APM tool of choice (Splunk, DataDog, AppD, DynaTrace)

Ingress Services Workshop with OpenShift

You will learn how to deploy a K8S and OpenShift hosted application / container on BIG-IP using F5 CIS (container ingress service) and N+ KIC

Modern Apps

NGINX 101 – Automating, Configuring and Monitoring API Gateways

Hands-on installation of NGINX Plus and Controller. Configure API Gateways and API Management.

Migrating to the NGINX ADC

Attend this workshop to gain hands on experience deploying NGINX Plus as a Software Load Balancer and leveraging your F5 skills to deploy on this container-friendly architecture.

NGINX Kubernetes Ingress Controller

Learn how the latest developments introduced in NGINX KIC and get started using the N+ KIC to load balance, route, and secure Kubernetes applications make deploying Kubernetes easier than ever before.

NGINX ADC: Basics and Best Practices

We cover best practices for installing, configuring, and troubleshooting NGINX Plus. We provide insights about using NGINX Controller to manage your NGINX Plus instances.

DevSecOps

Tools to help standardize CI/CD Processes across IT silos

Hands on workshop with NGINX Controller 3.x for teams of Dev Ops, Sec Ops, and Net Ops professionals to gain experience solving the challenges that modern application processes put on Enterprises.

TBD w/ Sysdig

Customize per the Partner

3rd Party

Global Server Load Balancing with NS1 and NGINX

Learn about how NGINX integrates with NS1, which uses a modern API-first architecture to radically transform DNS, DHCP, and IP address management into an intelligent, efficient and automated system to improve performance and reliability across your PoPs.

AI-Powered API Cybersecurity with NGINX and Ping Identity

Learn how to secure every API touchpoint using Ping Identity and the NGINX Controller API Management Module.

Analyzing Logs with N+ & Datadog

Learn how to leverage Datadog's Log Management solution to get the most out of your NGINX logs, which contain a wealth of information that you can use to improve usage, health, and performance.

API Security with NGINX & Okta

Learn how to secure every API touchpoint using Okta and the NGINX Controller API Management Module. Solve "sensitive data exposure" by APIs, #3 on the 'OWASP Top 10 Application Security Risks' list.

Modern Apps

Automate CI/CD w/ Terraform Workshop

Customize per the Partner

TBD w/ Rancher

Customize per the Partner

TBD w/ Hashi (Vault / Consul)

Customize per the Partner

TBD w/ OpenShift

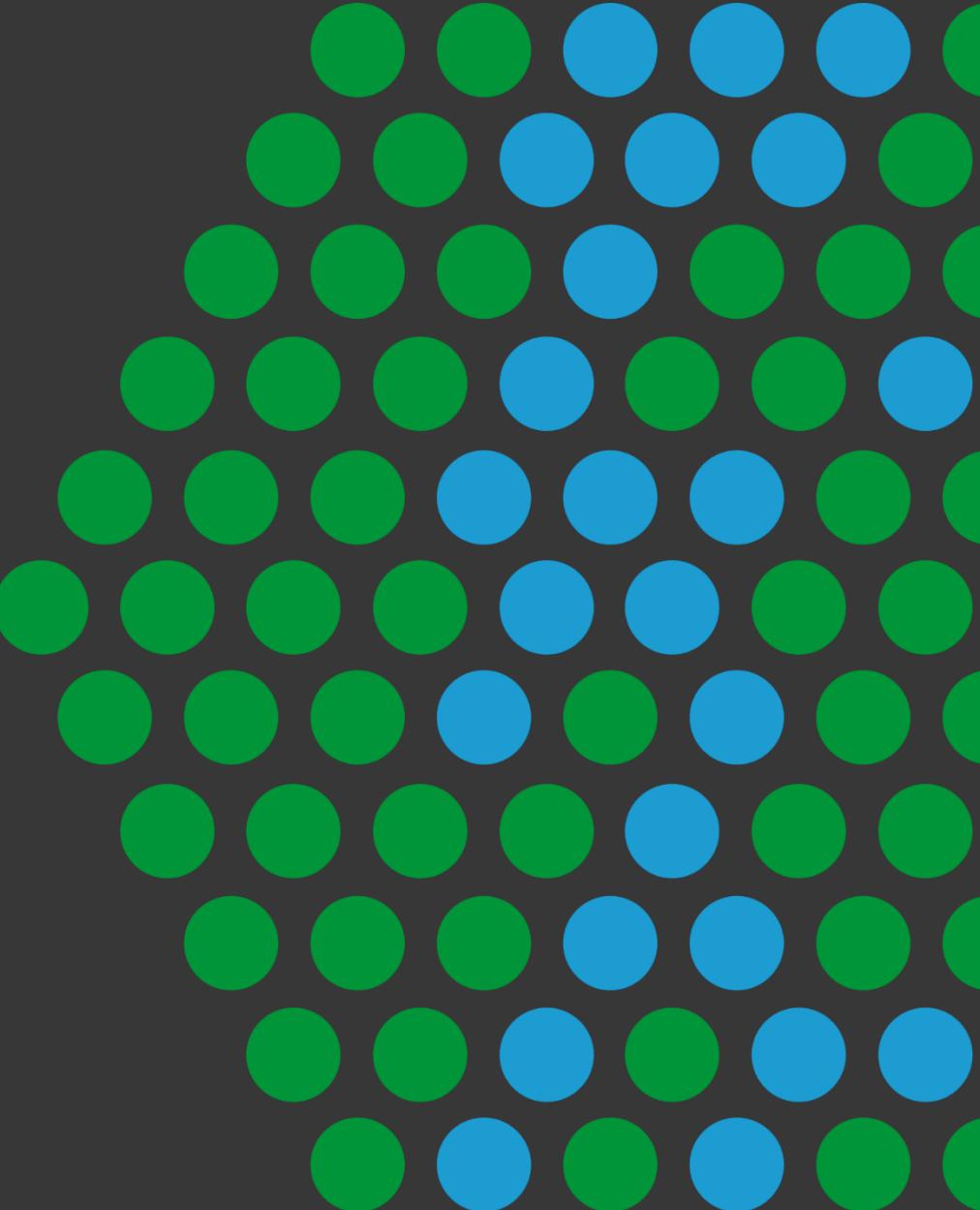
Customize per the Partner

TBD w/ PCF

Customize per the Partner



Thank You



ASM and NGINX App Protect Feature Matrix (GA)

Features	ASM/AWAF	NGINX App Protect
OWASP Top 10 prevention	Yes	Yes
Advanced protection against threats including Web injections, data leakage, session hijacking, HPP attacks, buffer overflows, shellshock	Yes	Yes
SSL termination with re-encryption	Yes	Yes
Client-side certification support	Yes	Yes
Response checking	Yes	Yes
Violation risk scoring	Yes	Yes
Live signature updates	Yes	Yes
Application layer encryption	Yes	No
Brute Force mitigation	Yes	No
Credential Stuffing protection	Yes	No
Behavioral denial-of-service (DoS) protection	Yes - protection for all applications	No
L7 DoS and DDoS detection including HASH DoS, Slowloris, floods, Keep-Dead, XML bomb	Yes	No
Web scraping prevention	Yes	No
Automated attack defense and bot detection	Yes	No
Mobile bot protection	Yes - with the F5 Anti-Bot Mobile SDK	No
Geolocation blocking	Yes	No
IP intelligence reputation services	Yes - with F5 IP Intelligence Services	No
Security incident and violation correlation	Yes	No
Client authentication	LDAP, RADIUS; more methods available with F5 Access Manager	No
Database security	Yes - with Oracle Database firewall	No
Web service encryption and decryption	Yes - and with signature validation	No
Device-ID detection and fingerprinting	Yes	No
WebSocket traffic filtering	Yes	No
IP shunning (layer 3 blacklisting in HW)	Requires F5 BIG-IP AFM license	No

What is an API?

CONNECTIVE TISSUE OF THE DIGITAL WORLD

API: – Application Programming Interface

API clients or consumers such as a mobile app invoke API calls to deliver functionality.

Examples:

- Uber uses Google Maps APIs

<https://maps.googleapis.com/maps/api/directions/json?origin=Disneyland&destination=Universal+Studios+Hollywood>

Uber spent \$58 million on it over three years

- Dropbox:

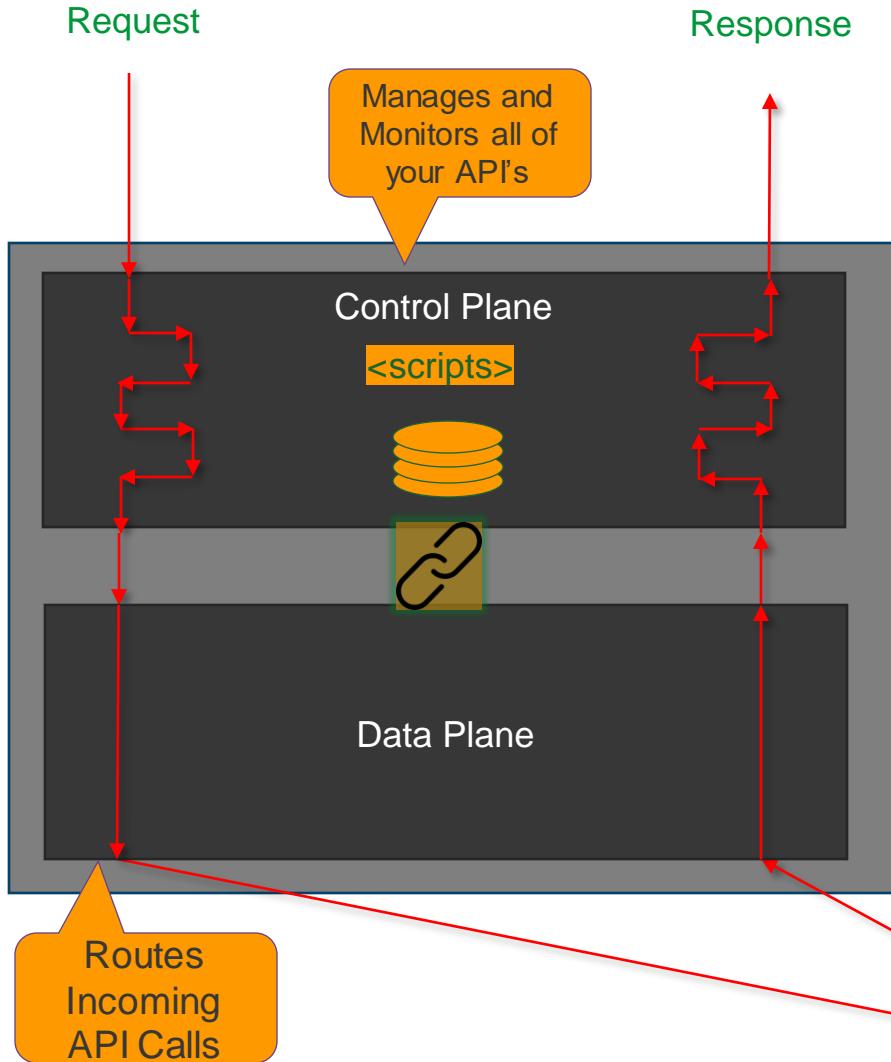
https://api.dropboxapi.com/2/file_requests/count:

Returns the total number of file requests owned by this user. Includes both open and closed file requests.

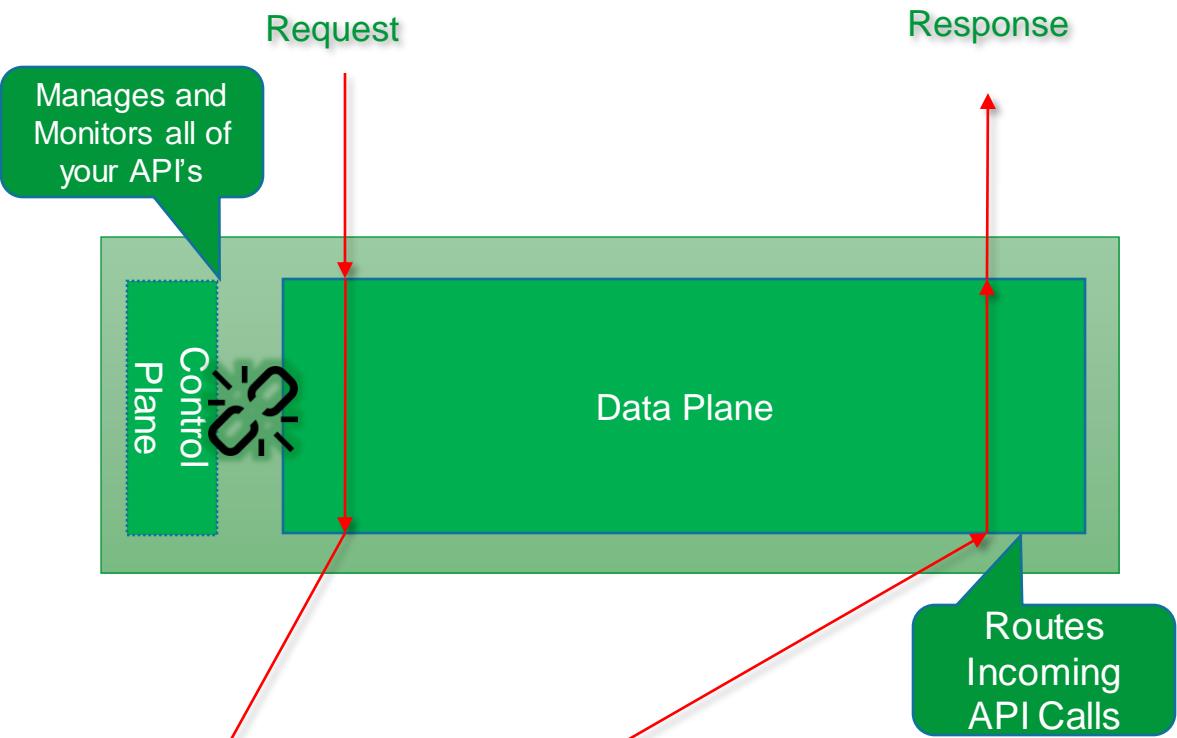
```
{ "file_request_count": 15 }
```

NGINX API Management under the covers

Traditional API Management



NGINX API Management



Kubernetes Ingress Controller

An advanced Layer 7 load-balancing solution for exposing Kubernetes services to the Internet

Where are
you with
Kubernetes?

- NGINX commonly used as Ingress Controller
- Dynamic reconfiguration of endpoints (no configuration reloading)
- Additional metrics, provided by a streamlined Prometheus exporter
- Dedicated Helm chart repository
- Support for Custom resources to expose more (all) NGINX Plus features as an Ingress

Kubernetes Ingress Controllers

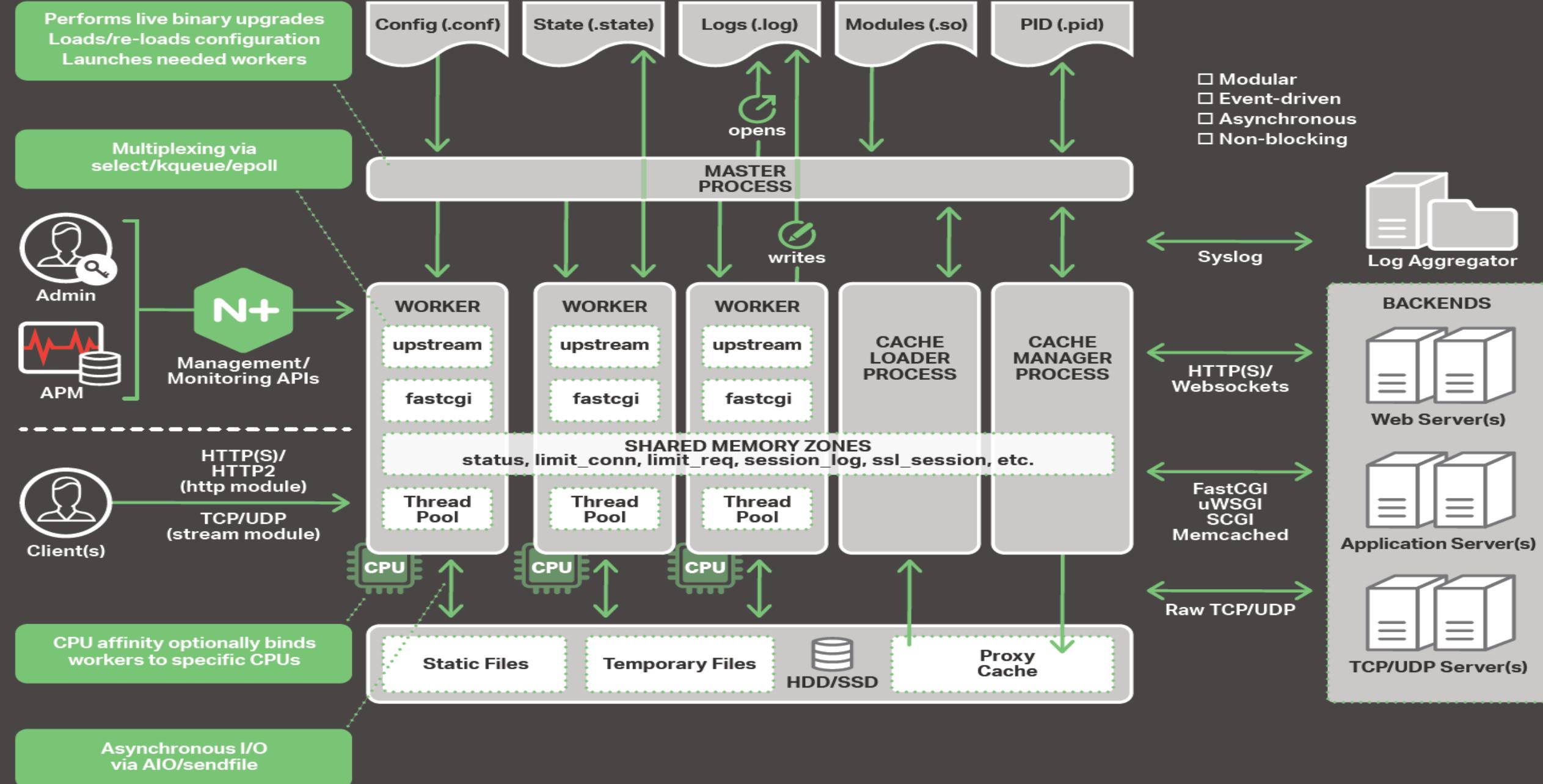
The Key Differences

The table below summarizes the key difference between nginxinc/kubernetes-ingress and kubernetes/ingress-nginx Ingress controllers. Note that the table has two columns for the nginxinc/kubernetes-ingress Ingress controller, as it can be used both with NGINX and NGINX Plus. For more information about nginxinc/kubernetes-ingress with NGINX Plus, read [here](#).

Aspect or Feature	kubernetes/ingress-nginx	nginxinc/kubernetes-ingress with NGINX	nginxinc/kubernetes-ingress with NGINX Plus
Fundamental			
Authors	Kubernetes community	NGINX Inc and community	NGINX Inc and community
NGINX version	Custom NGINX build that includes several third-party modules	NGINX official mainline build	NGINX Plus

<https://github.com/nginxinc/kubernetes-ingress/blob/master/docs/nginx-ingress-controllers.md>

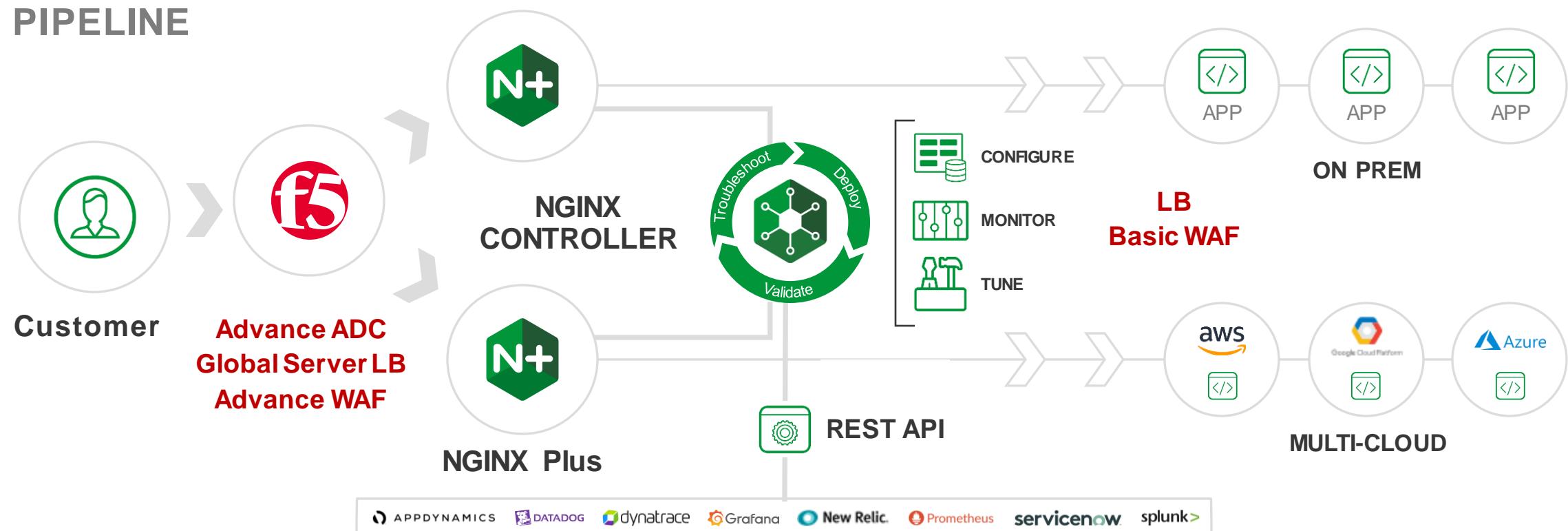
The Powerful and Efficient Architecture of NGINX



ADC for Multi-Cloud



DELIVER NEW APPS FREELY IN DEVOPS CI/CD PIPELINE



CHALLENGES

- Managing disparate application services and policies across multi-cloud
- Growing focus on container app dev and production
- Controlling 100s and 1000s of open-source instances
- Applications under attack cause disruption and outages

SOLUTION

NGINX Controller for management services

NGINX Plus with WAF module for Cloud-Native Apps

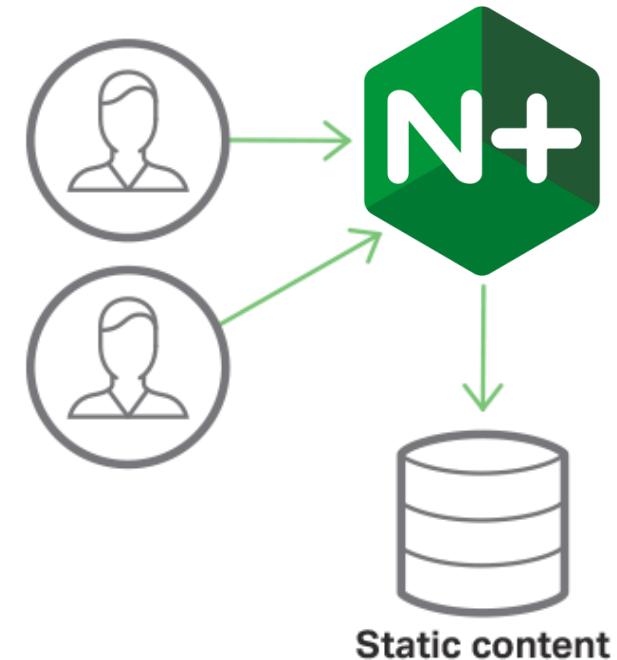
Where Do We Focus?

- Web Server
- Load Balancer
- Dynamic Modules
- Security Controls
- Monitoring
- High Availability
- Kubernetes Integration
- Content Cache
- Streaming Media
- Microservices

Web Server

Efficiently deliver static assets

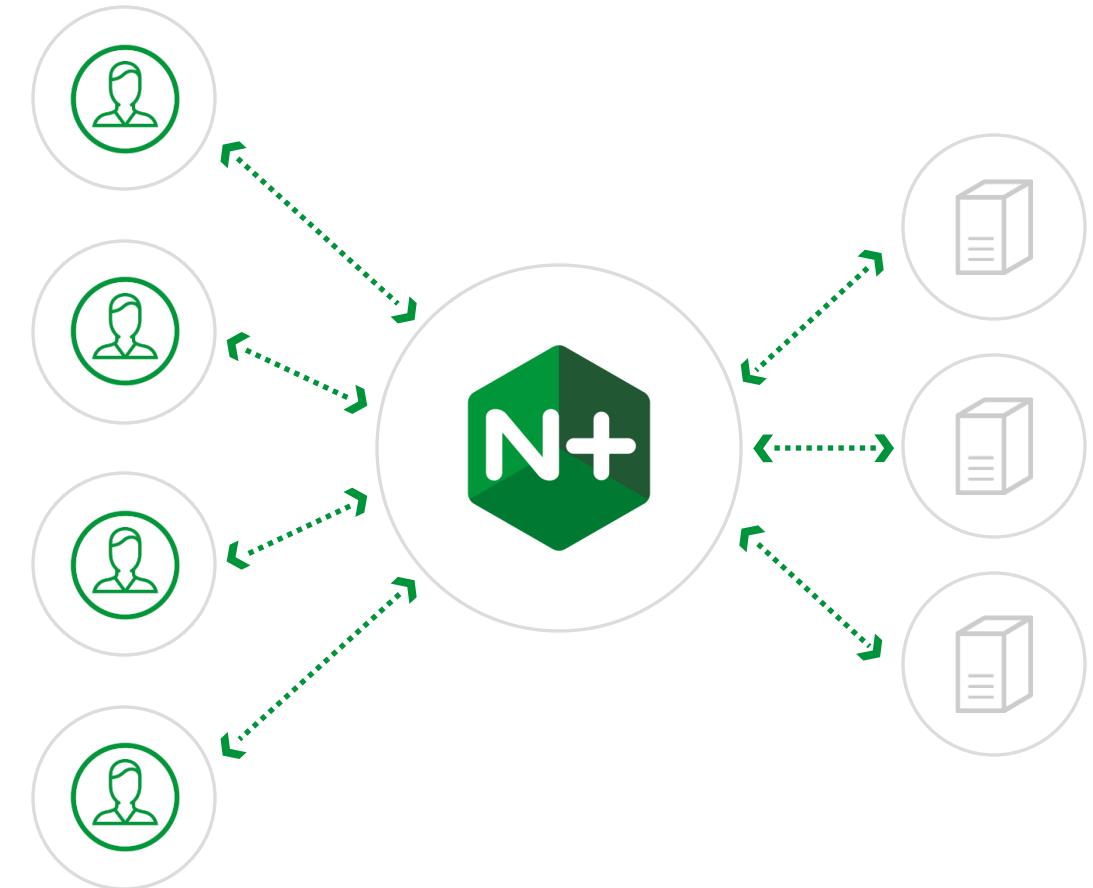
- Handle hundreds of thousands of clients simultaneously
- Use up to 90% less memory than other web servers
- Reverse proxy multiple protocols: HTTP, Memcached, PHP-FPM, SCGI, uwsgi
- HTTP/2 gateway



Load Balancer

Complete software load balancing

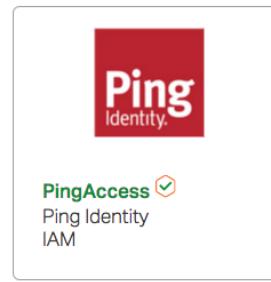
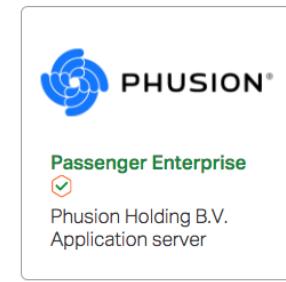
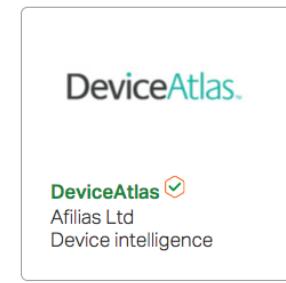
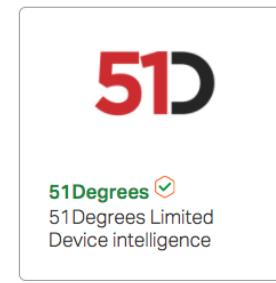
- HTTP, TCP, and UDP load balancing
- Layer 7 request routing using URI, cookie, args, and more
- Session persistence based on cookie
- Active health checks on status code and response body
- Service discovery using DNS



Dynamic Modules

Dynamically plug in additional features

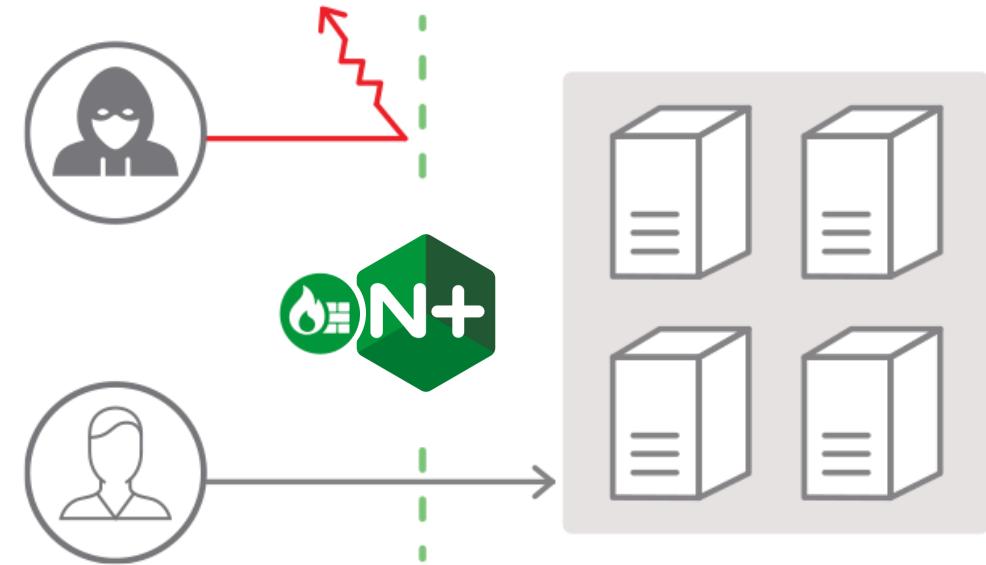
- **Single sign-on** : ForgeRock, IDF Connect, Ping Identity
- **Device detection**: WURFL, DeviceAtlas, 51Degrees
- **Security**: Stealth, Wallarm, NGINX WAF
- **Scripting**: NGINX JavaScript module, Lua
- **GeoIP**: Locate users by IP address (requires MaxMind GeoIP db)
- Dynamic modules repository



Security Controls

Protect your applications:

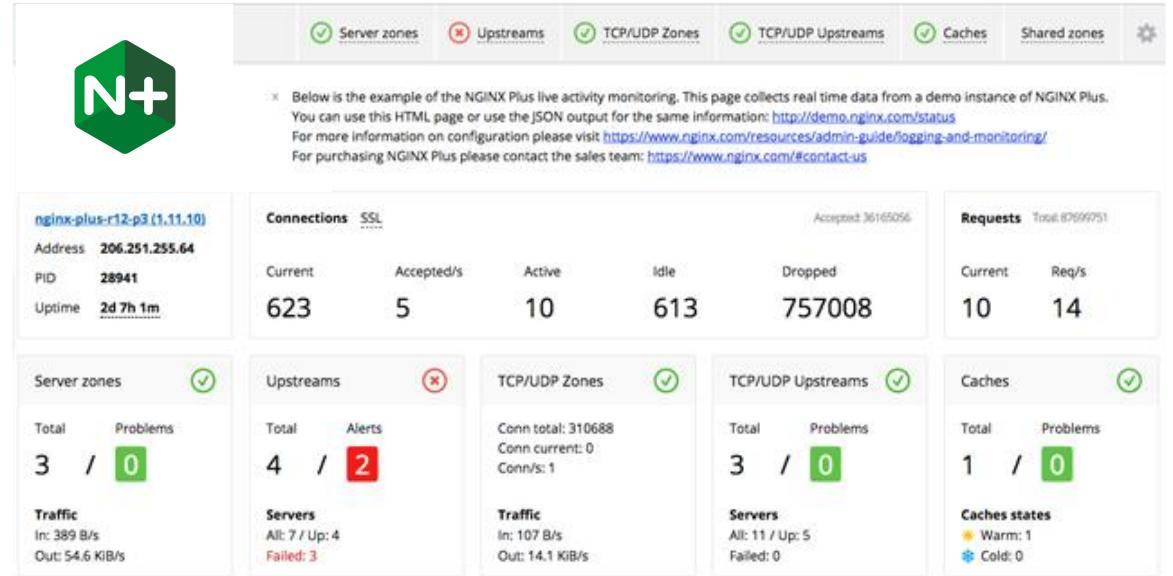
- Request/connection limiting
- Dual stack RSA/ECC SSL offloading
- IP access control list (ACL)
- JWT authentication for APIs and OpenID Connect single sign-on (SSO)
- NGINX WAF dynamic module



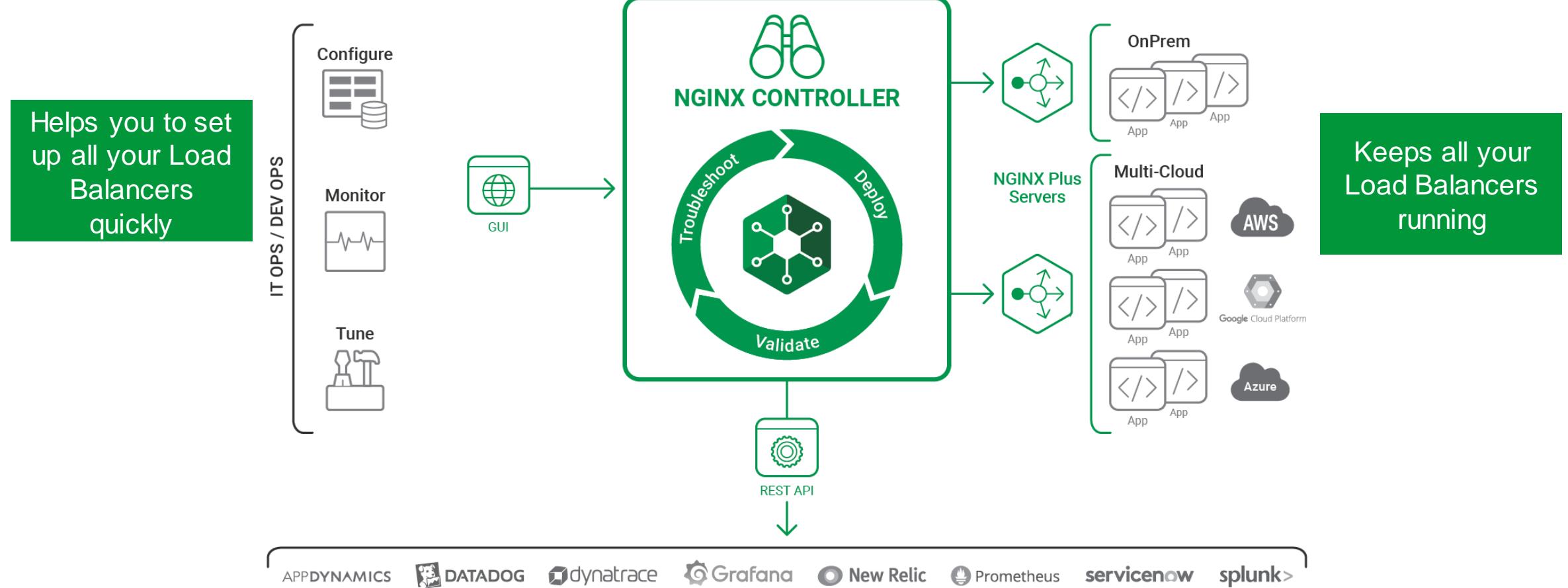
Monitoring

Debug complex application architectures

- Real-time performance metrics
- Plug-ins for AppDynamics, Datadog, Dynatrace, New Relic, & NGINX Controller
- Extended status with more than 90 unique metrics
- Built-in, real-time graphical dashboard
- JSON and HTML output for integration with custom monitoring tools



NGINX Controller?

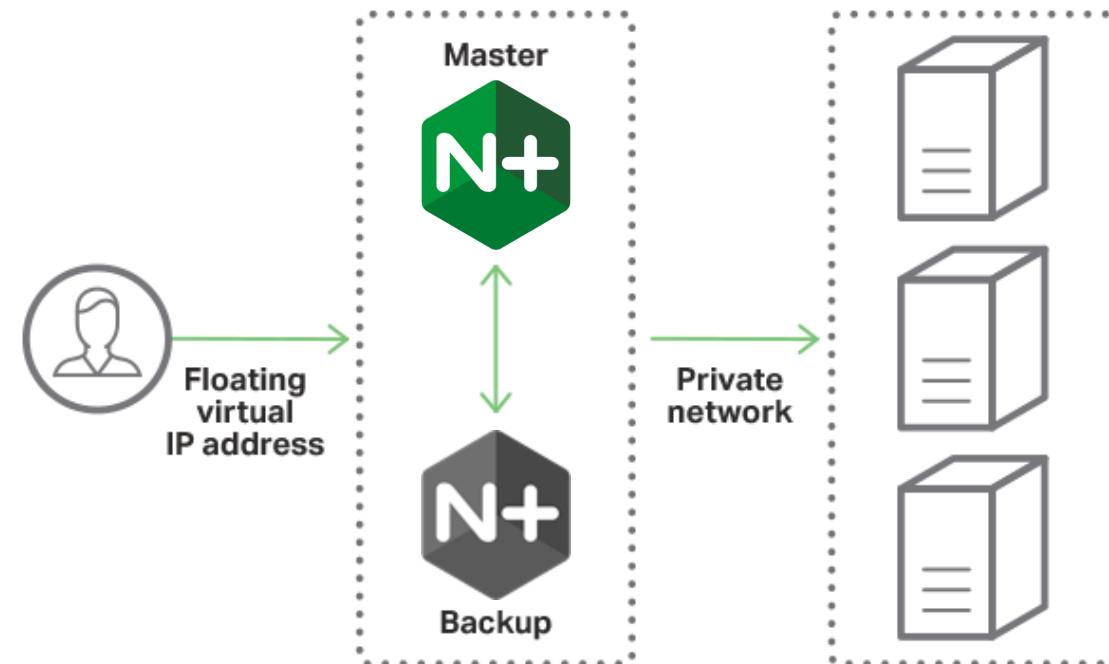


NGINX Controller manages NGINX Plus instances

High Availability (HA)

Scalable and reliable HA deployments:

- Active-active and active-passive HA modes
- Configuration synchronization between servers in a cluster
- State sharing for features including Sticky-Learn session persistence, rate limiting, and key-value stores
- Easy installation with built-in script





Kubernetes Integration

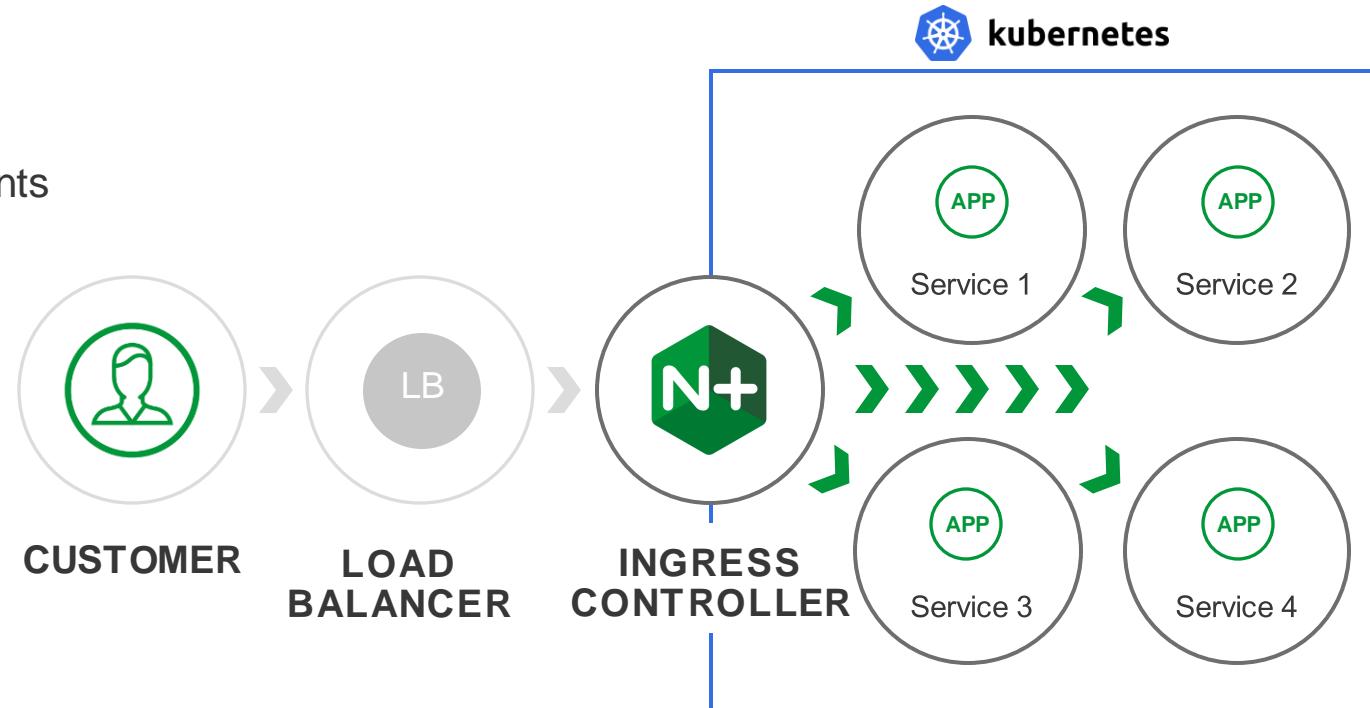
HIGH PERFORMANCE TRAFFIC MANAGEMENT FOR KUBERNETES

BENEFITS:

- Flexible app services for Kubernetes and OpenShift Router environments
- Consistency across pod deployments for rapid code to customer
- Scale apps with high performance

PLUS CAPABILITIES:

- Session persistence
- JSON Web Token (JWT) authentication
- 24/7 support, no additional cost
- Load balancing w/ SSL/TLS termination
- WebSocket and HTTP/2 support
- URI rewriting before request is forwarded to application



Available in Kubernetes and OpenShift



Programmability

Infrastructure as Code Dynamically create custom configs

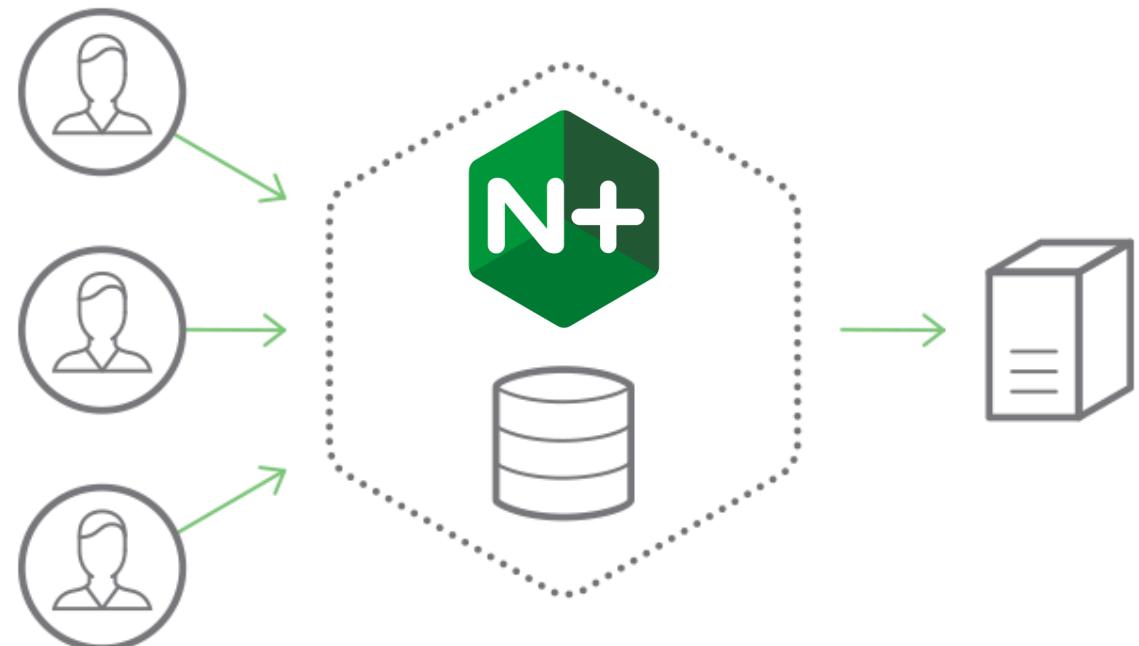
- NGINX JavaScript module for scripting
- Lua scripting language
- Ansible, Chef, and Puppet integration
- NGINX Plus API for managing upstream servers, key-value stores, and real-time metrics
- Dynamic reconfiguration without process reloads



Content Cache

The cache that powers the largest CDNs

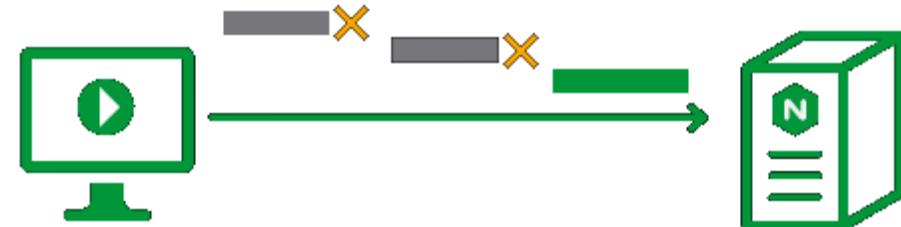
- Cache static and dynamic content
- Improve dynamic content performance with microcaching
- Serve "stale" content while revalidating in background for better performance
- Override or set Cache-Control headers
- Manage the cache easily with the cache-purging API



Streaming Media

Scalably deliver streaming media

- Live streaming: RTMP, Apple HTTP Live Streaming (HLS), Dynamic Adaptive Streaming over HTTP (DASH)
- VOD: Flash (flv), MP4
- Adaptive-bitrate VOD: HLS, Adobe HTTP Dynamic Streaming (HDS)
- Bandwidth controls for MP4 streaming





NGINX is a leader in microservices

**11
MILLION**

New domains run
NGINX each month,
or 4 per second*

**3
MILLION**

NGINX instances
run in production
micro-services**

**1
MILLION**

NGINX GitHub pulls
for Kubernetes
Ingress controller

**250
CUSTOMERS**

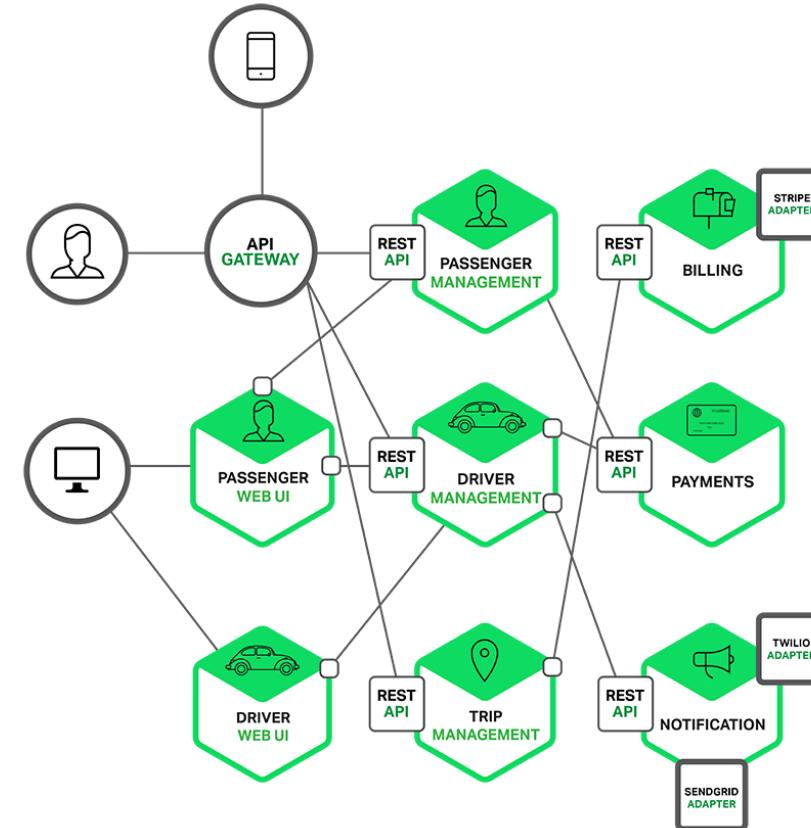
use NGINX
in production
micro-services**

*Source: Netcraft February Web Server Survey, ** Source: Internal customer data and surveys

NGINX Plus for Microservices

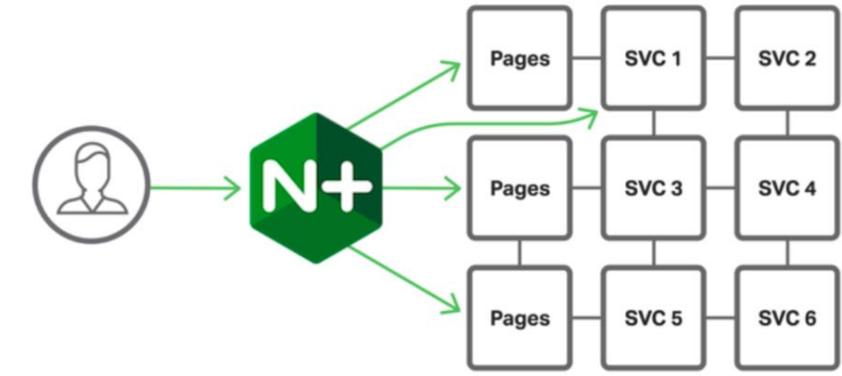
NGINX is in each microservice, ensuring they are:

- Connected
- Served
- Authenticated
- Secured
- Cached
- Load Balanced
- Scaled

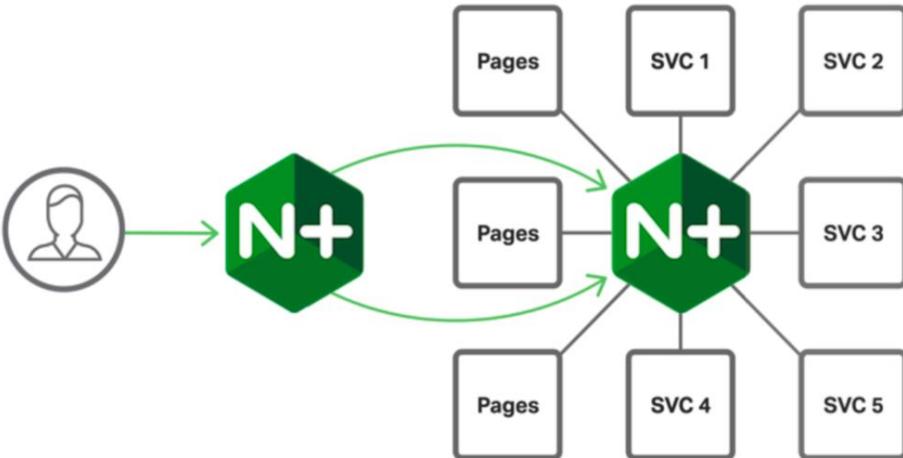


Microservices Dashboard

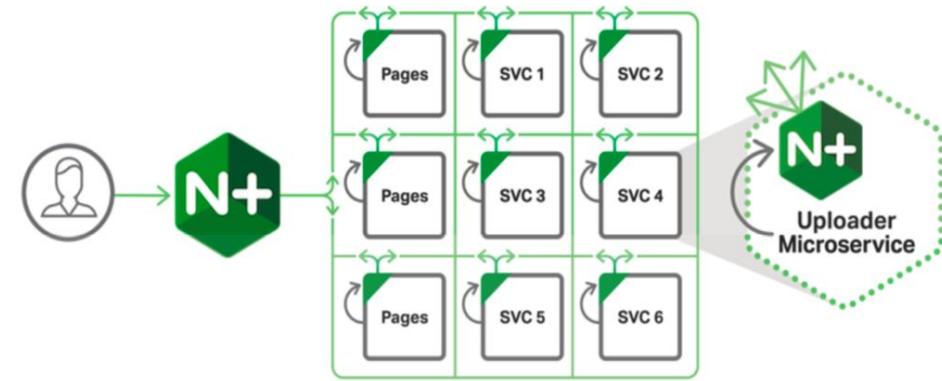
Proxy Model



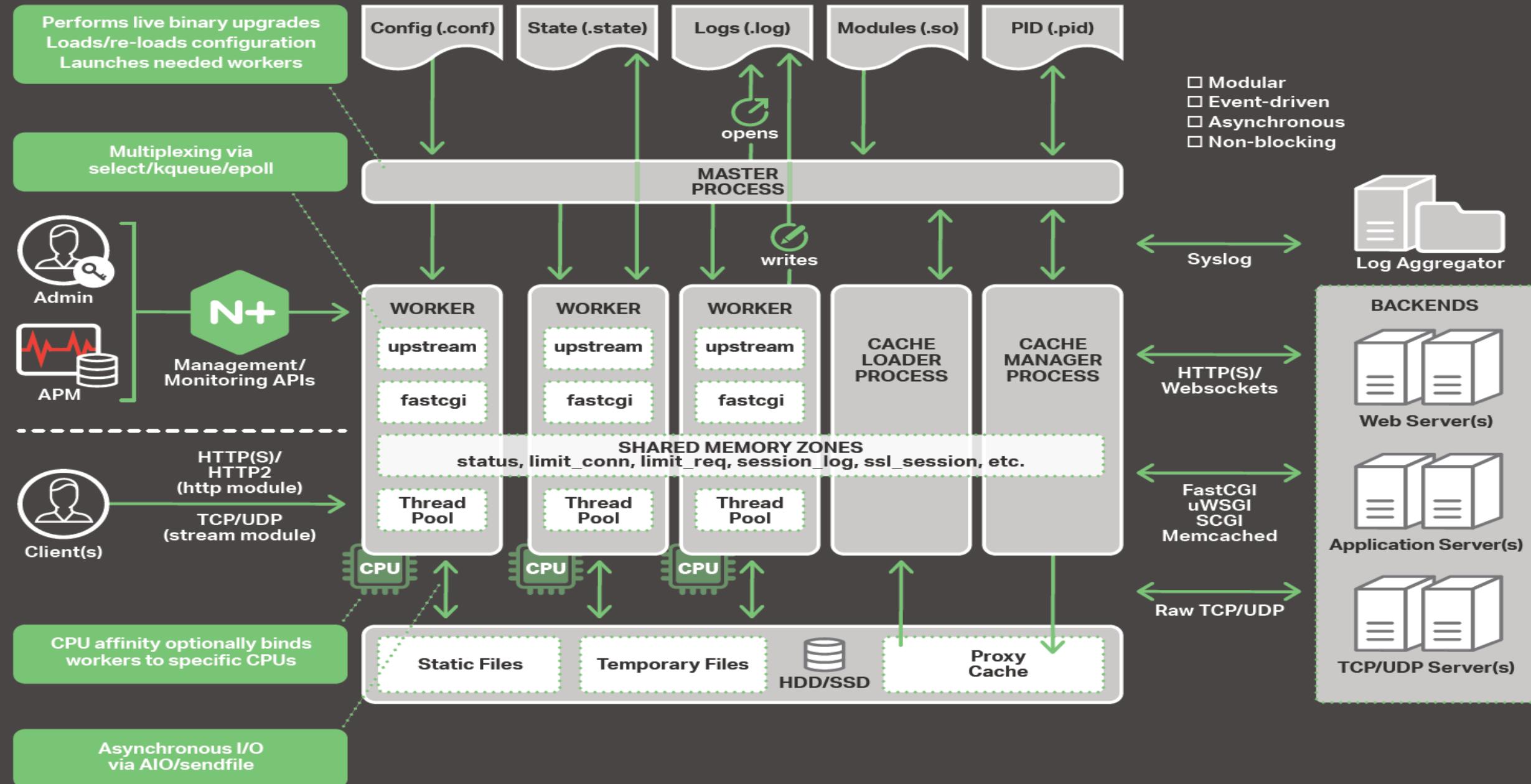
Router Mesh Model



Fabric Model



The Powerful and Efficient Architecture of NGINX



Thank You



Technical Slides



Key NGINX Commands

- `nginx -h` Display NGINX help menu
- `nginx -t` Check if NGINX configuration is ok
- `nginx -s reload` Check config is ok and gracefully reload NGINX processes
- `nginx -V` Similar to `-v`, but with more detailed information
- `nginx -T` Dump full NGINX configuration



Key NGINX Files and Directories

/etc/nginx/

nginx.conf

```
http {  
    include conf.d/*.conf;  
}
```

Global settings
(tunings, logs, etc)

HTTP block

/etc/nginx/conf.d/

virtualserver1.conf

```
server {  
    listen <parameters>;  
  
    location <url> {  
        ...  
    }  
  
    upstream { ... }  
}
```

Listen for requests

Rules to handle
each request

Optional: proxy to
upstreams

/var/log/nginx/

error.log

Important operational messages

access.log

Record of each request (configurable)



NGINX Dynamic Modules

- Handful of modules available in our repository

```
$ apt-get install nginx-plus-module-<name>
```
- Custom modules should be compiled against equivalent version of OSS

```
$ wget http://nginx.org/download/nginx-1.15.10.tar.gz
$ tar -xzvf nginx-1.15.10.tar.gz
$ cd nginx-1.15.10/
$ ./configure --with-compat --add-dynamic-module=../nginx-foo-modules
$ make modules
```
- `load_module` - allows loading modules...

```
$ load_module modules/module-name.so;
```



Simple Virtual Server

```
server {  
    listen      80 default_server;  
    server_name www.example.com;  
  
    return     200 "Hello World!";  
}
```

- `server` defines the context for a virtual server
- `listen` specifies IP/port NGINX should listen on. No IP means bind to all IPs on system
- `server_name` specifies hostname of virtual server
- `return` tells NGINX to respond directly to the request.



Basic Web Server Configuration

```
server {  
    listen      80 default_server;  
    server_name www.example.com;  
  
    location /web/ {  
        root   /usr/share/nginx/html;  
        index  index.html index.htm;  
    }  
}
```

- `root` specifies directory where files are stored
- `index` defines files that will be used as an index

- `index: www.example.com` -> `/usr/share/nginx/html/index.html`
- `root: www.example.com/i/file.txt` -> `/usr/share/nginx/html/i/file.txt`
- `alias: www.example.com/i/file.txt` -> `/usr/share/nginx/html/file.txt`



Basic Load Balancing Configuration

```
upstream my_upstream {  
    server server1.example.com;  
    server server2.example.com;  
    least_time;  
}  
  
server {  
    location / {  
        proxy_set_header Host $host;  
        proxy_pass http://my_upstream;  
    }  
}
```

- `upstream` defines the load balancing pool
- Default load balancing algorithm is round robin. Others available:
 - `least_conn` selects server with least amount of active connections
 - `least_time` factors in connection count and server response time. Available in NGINX Plus only.
- `proxy_pass` links virtual server to upstream
- By default NGINX rewrites Host header to name and port of proxied server. `proxy_set_header` overrides and passes through original client Host header.



Basic Reverse Proxy Configuration

```
server {  
    location ~ ^(.+\.php)(.*)$ {  
        fastcgi_split_path_info ^(.+\.php)(.*$);  
  
        fastcgi_pass 127.0.0.1:9000;  
  
        fastcgi_index index.php;  
        include fastcgi_params;  
    }  
}
```

- Requires PHP FPM:
`apt-get install -y php7.0-fpm`
- Can also use PHP 5
- Similar directives available for uWSGI and SCGI.
- Additional PHP FPM configuration may be required



Basic Caching Configuration

```
proxy_cache_path /path/to/cache levels=1:2  
    keys_zone=my_cache:10m max_size=10g  
    inactive=60m use_temp_path=off;  
  
server {  
    location / {  
        proxy_cache my_cache;  
        proxy_set_header Host $host;  
        proxy_pass http://my_upstream;  
    }  
}
```

- `proxy_cache_path` defines the parameters of the cache.
- `keys_zone` defines the size of memory to store cache keys in. A 1 MB zone can store data for about 8,000 keys.
- `max_size` sets upper limit of cache size. Optional.
- `inactive` defines how long an object can stay in cache without being accessed. Default is 10 m.
- `proxy_cache` enables caching for the context it is in



Basic SSL Configuration

```
server {  
    listen      80 default_server;  
    server_name www.example.com;  
    return 301 https://$server_name$request_uri;  
}  
  
server {  
    listen 443 ssl default_server;  
    server_name www.example.com;  
    ssl_certificate cert.crt;  
    ssl_certificate_key cert.key;  
  
    location / {  
        root   /usr/share/nginx/html;  
        index index.html index.htm;  
    }  
}
```

- Force all traffic to SSL is good for security and SEO
- Use Let's Encrypt to get free SSL certificates, see:
nginx.com/blog/using-free-ssltls-certificates-from-lets-encrypt-with-nginx



Multiplexing Multiple Sites on One IP

```
server {  
    listen      80 default_server;  
    server_name www.pizza.com;  
    # ...  
}  
  
server {  
    listen      80;  
    server_name www.tacos.com;  
    # ...  
}  
  
server {  
    listen      80;  
    server_name www.sushi.com;  
    # ...  
}
```

- NGINX can multiplex a single IP/port using the Host: header.
- `default_server` defines the virtual server to use if Host header is empty. It is best practice to have a `default_server`.



