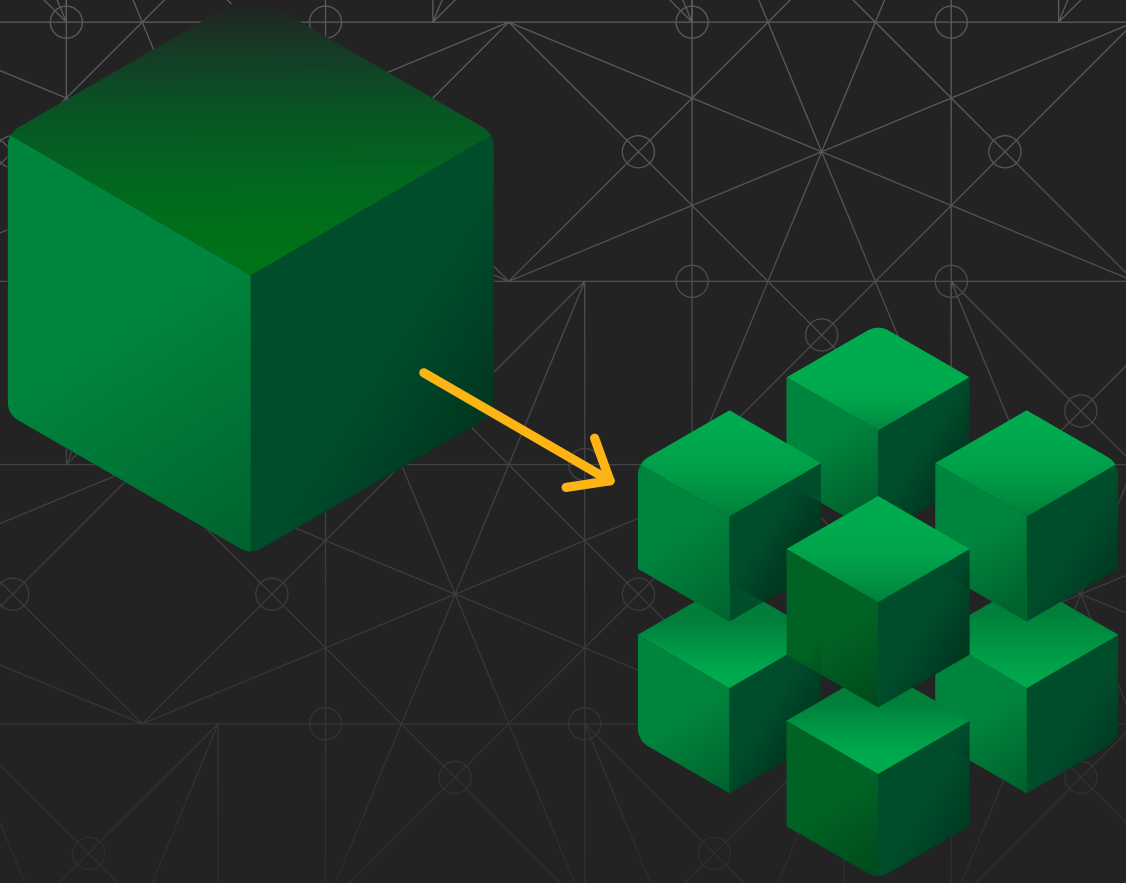
NGINX Application Platform

High Performance Application Delivery for Microservices



NGINX

To Survive, Every Company Must Become a Tech Company

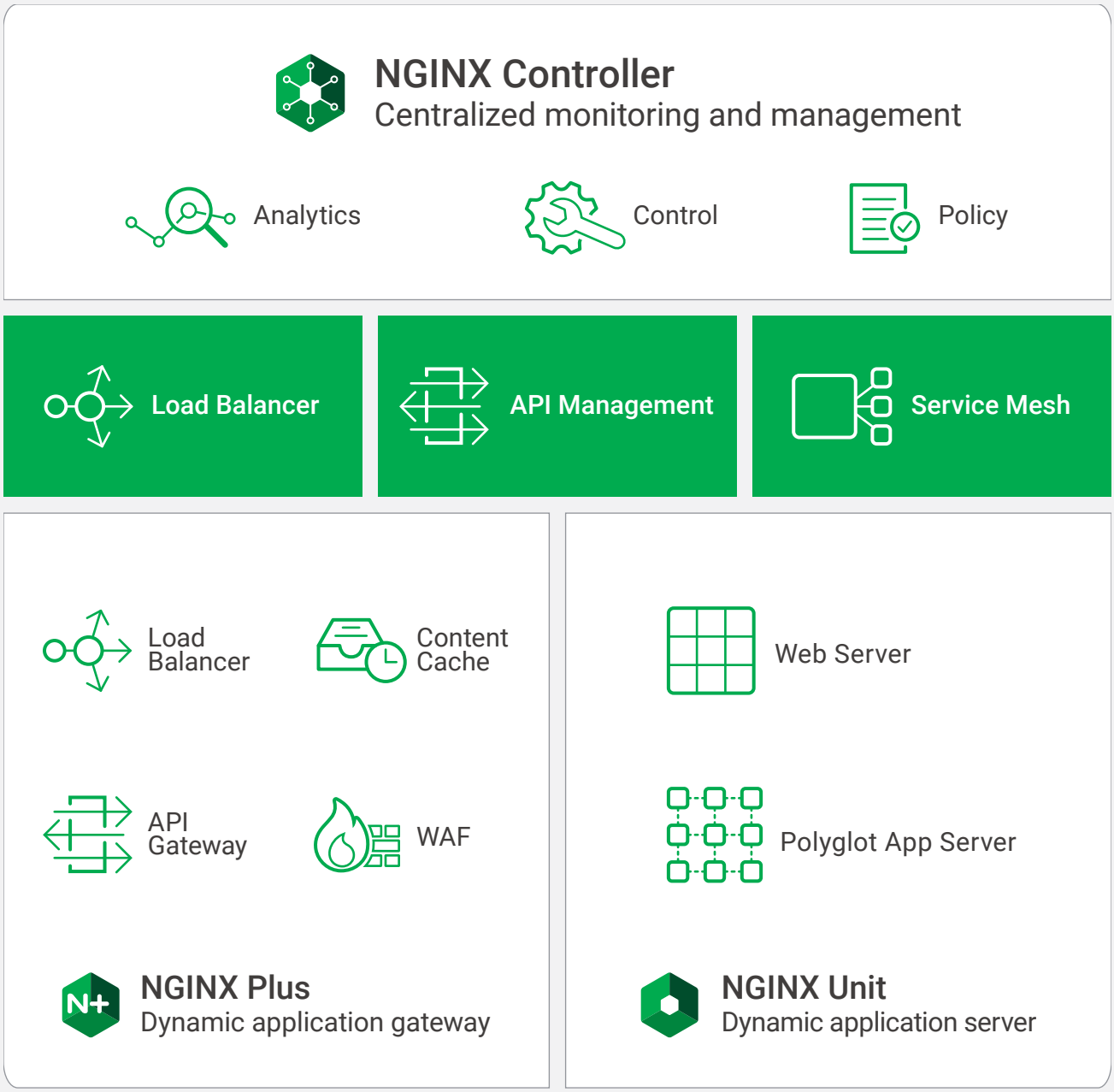
Whether or not we like it, we live in a digital world. Technology is front and center in our day-to-day lives. From entertainment to banking to how we communicate with our friends, technology has changed how we interact with the world. Businesses of all sizes and in all industries are rolling out compelling digital capabilities to attract, retain, and enrich customers.

But being digital isn't enough, you need to be the best. A recent study showed that a 4-second video buffering delay caused the stress equivalent to watching a horror movie alone.[1] People demand the best digital experiences; unhappy users will quickly move to competitors that can give them a better experience.

So how do you master the digital world? The Internet giants, such as Netflix and Amazon, have pioneered an approach to building applications powered by a new suite of technologies. These tools are overwhelmingly open source, cloud-native, and place a premium on adaptability, performance, and scale. Of all these new tools, none is more ubiquitous than NGINX.

More than 300 million websites[2] rely on NGINX Open Source to deliver their applications. Our technology helps companies remove friction from digital delivery, optimize digital supply chains, and roll out digital services faster. The NGINX Application Platform is a consolidated set of tools that improves application performance, automates application delivery, and decreases capital and operational costs.

1. https://www.ericsson.com/en/press-releases/2016/2/streaming-delays-mentally-taxing-for-smartphone-users-ericsson-mobility-report
2. https://news.netcraft.com/archives/2018/10/29/october-2018-web-server-survey.html



The NGINX Application Platform is a set of four tools are at the core of what organizations need to create applications with performance, reliability, security, and scale. The NGINX Application Platform includes:

NGINX Controller – Mission control for your applications, NGINX Controller provides centralized monitoring and management.

NGINX Plus – Based on NGINX Open Source, NGINX Plus consolidates multiple application delivery functions into a single piece of software

NGINX WAF – Protects against Layer 7 attacks such as SQL Injection and Remote Code Execution.

NGINX Unit – Open source, multi-language application server. Supports Go, Node.js, Perl, PHP, and Python.

Controller

Centralized Monitoring and Management

NGINX Controller is NGINX's control plane solution that manages the NGINX data plane. Whether you deploy NGINX as a load balancer/web server, API Gateway or a proxy in a service mesh environment, NGINX Controller enables you to manage the entire lifecycle of these solutions.

Built on a modular architecture, you can use NGINX Controller to configure, validate and troubleshoot your load balancers. NGINX Controller's API Management module allows you to define, publish, secure, monitor as well as collect analytics about APIs. An upcoming Service Mesh Module for NGINX Controller will simplify how organizations move from common Ingress patterns for containers to more complex service mesh architectures designed to optimize management of dozens, hundreds, or thousands of microservices.

"Our vision is for NGINX Controller to be the intelligent brain within the infrastructure, absorbing information and making decisions in real time without human intervention."

– Gus Robertson, CEO of NGINX, Inc.

Why NGINX Controller?

Simplified Management – Effortlessly deploy, validate and troubleshoot multiple NGINX Plus instances across a multi-cloud environment.

Strategic Command – Ensure application performance, reliability and availability by gaining deep visibility into OS, API, NGINX Plus metrics and adopting best practice performance and security recommendations.

Deployment Agility – Deploy new features and applications faster using a policy driven approach. Create policies for specific environments such as production or staging, monolith or microservices..

How NGINX Controller Works

NGINX Controller features are powered by a small agent that's installed on target NGINX Plus servers. Once the agent is installed and registered, NGINX Plus instances can be managed using NGINX Controller. The agent gathers performance data about these instances and sends it back to NGINX Controller for analysis. With the agent in place, Controller is able to provide robust monitoring and reporting capabilities.

Controller provides an intuitive wizard-style interface to configure NGINX Plus instances as load balancers and API gateways. You can configure features such as URL routing and SSL/TLS termination for your load balancers. Controller allows you to define and publish APIs and push these configurations to API Gateway instances

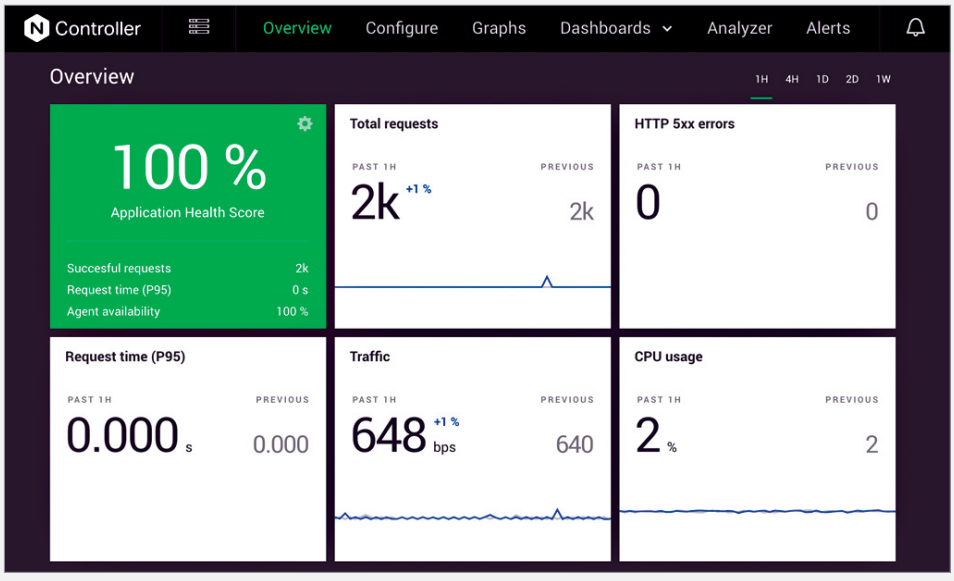
NGINX Controller is delivered as a Docker package. Docker offers complete flexibility in terms of deployment. You can deploy Controller in any environment that supports Docker containers – be it private or public cloud, virtual machines, or bare metal – to manage NGINX Plus instances operating across a multi-cloud environment.

NGINX Controller Features:

- 200+ metrics
- Alerting
- API management
- Customizable dashboards
- Configuration analysis
- Instance inventory
- Load balancing
- Multi-cloud
- REST API for monitoring

NGINX Controller provides real-time visibility as well as historical trending for more than 200 health and performance metrics. You can set thresholds for metrics and receive alerts via email when a threshold is exceeded. Or view the metrics you want to see on custom dashboards.

With NGINX Controller, you can centrally configure features such as load balancing and API management with just a few clicks in the graphical user interface. NGINX Controller supports a policy-based approach to management, so you can create configurations for test, staging, production, or line-of-business–specific applications, enabling developers to deploy applications faster.



NGINX Controller overview dashboard includes an Application Health Score

NGINX Plus

Dynamic Application Gateway

Ask anyone in IT and they'll tell you complexity is the enemy. By combining multiple tools into one, NGINX Plus reduces complexity. With a simpler infrastructure to manage, application teams can achieve greater agility and feature velocity. Whether you're modernizing legacy apps or building new, microservices-based apps, NGINX Plus helps you reliably scale digital services.

Many organizations today are dependent on legacy hardware application delivery controllers (ADCs). Hardware ADCs are a roadblock to creating scalable applications. NGINX Plus can be used to replace hardware ADCs, to give you significant cost savings combined with the agility of open source software.

"I evaluated NGINX Plus and was amazed at its power. The benchmarks just blew me away. The amount of traffic that NGINX Plus can handle is unreal – even beyond our needs."

– James Ridle,
IT Operations Manager,
State of Montana

Why NGINX Plus?

Cost savings – Save 80% over hardware ADCs while exceeding performance and functionality.

Flexibility – You can run NGINX Plus software in bare metal, virtual machines, container-based environments, and any cloud environment with no vendor lock-in.

Community – NGINX Plus is built on NGINX Open Source and backed by an enthusiastic community of over 400 million users, with thousands of community-contributed articles available to help you in your journey.

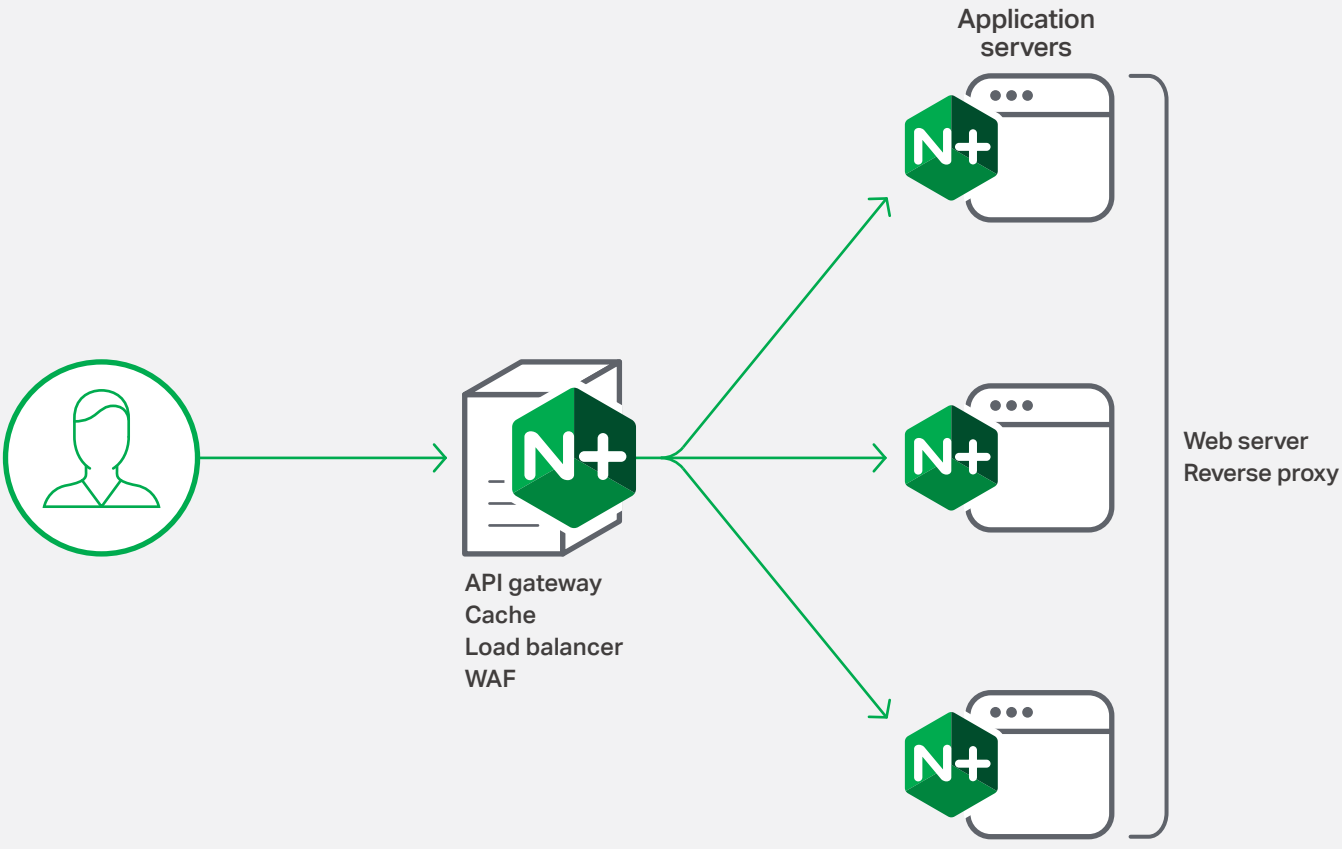
How NGINX Plus Works

NGINX Plus installs on x86, ARM, and Power8 servers running a Unix-like system, such as Linux or FreeBSD. It's installed using standard package management tools (`apt`, `yum`). NGINX Plus typically runs one worker process per CPU core for maximum performance.

NGINX Plus functions as a reverse proxy, accepting TCP connections and making new TCP connections to upstream servers. As a web server it serves static content directly and reverse proxies to PHP-FPM, uwsgi, and other application servers. As a cache, NGINX Plus handles both static and dynamic content. In addition, NGINX Plus can be used to stream media.

NGINX Plus Features:

- HTTP/TCP/UDP load balancer
- Content cache
- Web server
- SSL/TLS offload with dual-stack RSA/ECC
- Content compression
- Rate limiting
- JWT authentication
- High availability
- Reverse proxy for HTTP, FastCGI, memcached, SCGI, and uwsgi
- 20+ dynamic modules
- OpenID Connect SSO



Cache, load balance, and serve static content with a single tool, NGINX Plus.

WAF

Layer 7 Attack Protection

Even when you understand security, it is difficult to create secure applications given the pressures on today's companies. The NGINX Web Application Firewall (WAF module) protects applications against sophisticated Layer 7 attacks that might otherwise lead to loss of sensitive data, downtime, and reputation damage. The NGINX WAF module is based on the widely used ModSecurity open source software.

ModSecurity is one of the most trusted names in application security, protecting over a million websites today. As open source software, ModSecurity is backed by a large, enthusiastic community of security experts. Community backing extends beyond ModSecurity itself to the OWASP Core Rule Set (CRS), which protects against the most common and devastating attacks.

Like ADCs, WAFs have typically been in the realm of hardware appliances within the enterprise. And like ADCs, hardware WAFs suffer from the same shortcomings. They are costly and inflexible.

"Web applications – yours, mine, everyone's – are terribly insecure on average. We struggle to keep up with the security issues and need any help we can get to secure them."

– Ivan Ristić, creator of ModSecurity

Why NGINX WAF?

Cost savings – PCI compliance at a fraction of the cost of the hardware WAFs.

Battle tested – Used by more than a million websites, ModSecurity is the most trusted name in application security.

Agility – Respond quickly to emerging security threats with virtual patching.

How the NGINX WAF Module Works

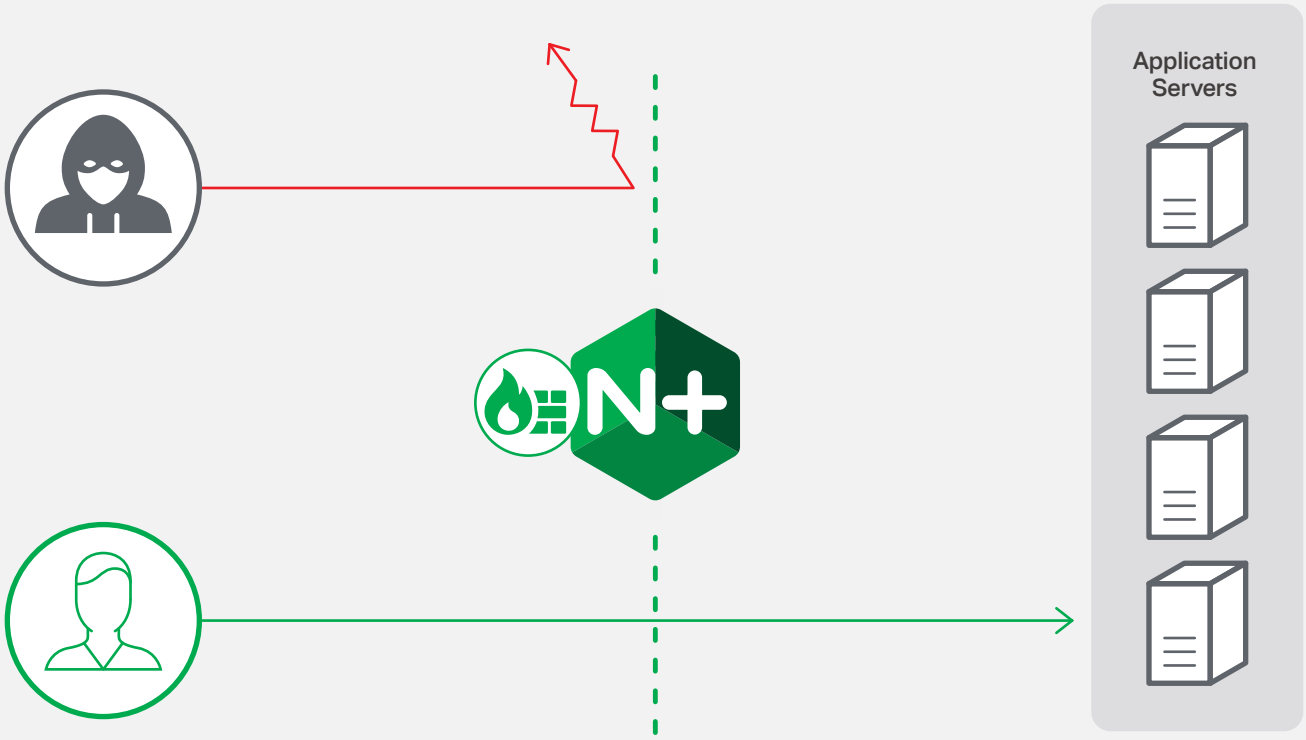
NGINX WAF is a dynamic module for NGINX Plus. It plugs into a running NGINX Plus instance. NGINX Plus still terminates connections and performs redirects and rate limiting as usual, but requests are sent to the NGINX WAF module before being forwarded to the backend server.

After doing initial processing, NGINX Plus passes the traffic to the NGINX WAF module, which inspects all parts of the request for malicious content or other anomalies. If the request is deemed malicious it can be blocked, logged, or both, depending on configuration.

If the request is determined to be acceptable, it is returned to NGINX Plus, which then satisfies the request.

The NGINX WAF module provides the following protections:

- **Layer 7 attack protection**
  SQL injection (SQLi), cross-site scripting (XSS), Local File Inclusion (LFI), Cross-site request forgery (CSRF), Remote File Inclusion (RFI), and more.
- **IP reputation**
  Block known bad IP addresses in real time using Project Honey Pot.
- **Scanner and bot detection**
  The NGINX WAF module can detect and block most scanners in use today.
- **Virtual patching**
  Respond to emerging threats in real time with a flexible PCRE regex-based rules language.



The NGINX WAF module protects against a broad range of Layer 7 attacks.

Unit

Dynamic application server

Most application servers we use in production environments today were written for static, monolithic architectures. Most have been retrofitted with sidecars and other add-ons for the dynamic functionality and consistent behavior needed in today's application environments. Unfortunately these add-ons increase complexity, decrease performance, and create a greater surface for failure.

NGINX Unit is a new dynamic web and application server built to meet the demands of both monolithic and distributed applications. NGINX Unit combines the functions of a web and application server into a single binary. NGINX Unit is dynamic by design and configured via a RESTful JSON API. All configuration changes are handled in memory, so there are no process reloads, no service disruptions, and no downtime.

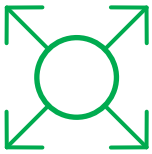
Many organizations today have applications written in multiple languages. With NGINX Unit you can run multiple languages on the same server, enabling you to consolidate application servers and reduce complexity. Go, Node.js, Perl, PHP, Python, and Ruby are supported, with more language support to come.

"Modern web applications are more complex than before. I see the lack of a simple and flexible base that is easy to use and reduces the complexity of modern web stacks. I created NGINX Unit to be this base."

– Igor Sysoev, creator of NGINX and NGINX Unit

Why NGINX Unit?

Reduce complexity – No need to tangle with multiple application servers and software stacks. Run your apps written in Go, Node.js, Perl, PHP, Python, and Ruby using the same consistent and powerful application server.



Dynamic by design – NGINX Unit was created to adapt in real time to the demands of distributed applications, with a RESTful JSON API, and immediate in-memory changes without process reloads or service disruptions.

Deploy with confidence – NGINX Unit is developed by the team behind NGINX, the most reliable and trusted name in application delivery.

How NGINX Unit works

NGINX Unit is software installed on a Unix-like system, such as Linux or FreeBSD. NGINX Unit creates a group of separate processes on one system. For security purposes only the *Main process* is run as *root*. Client connections, application processes, and code are run in separate isolated processes with limited rights.

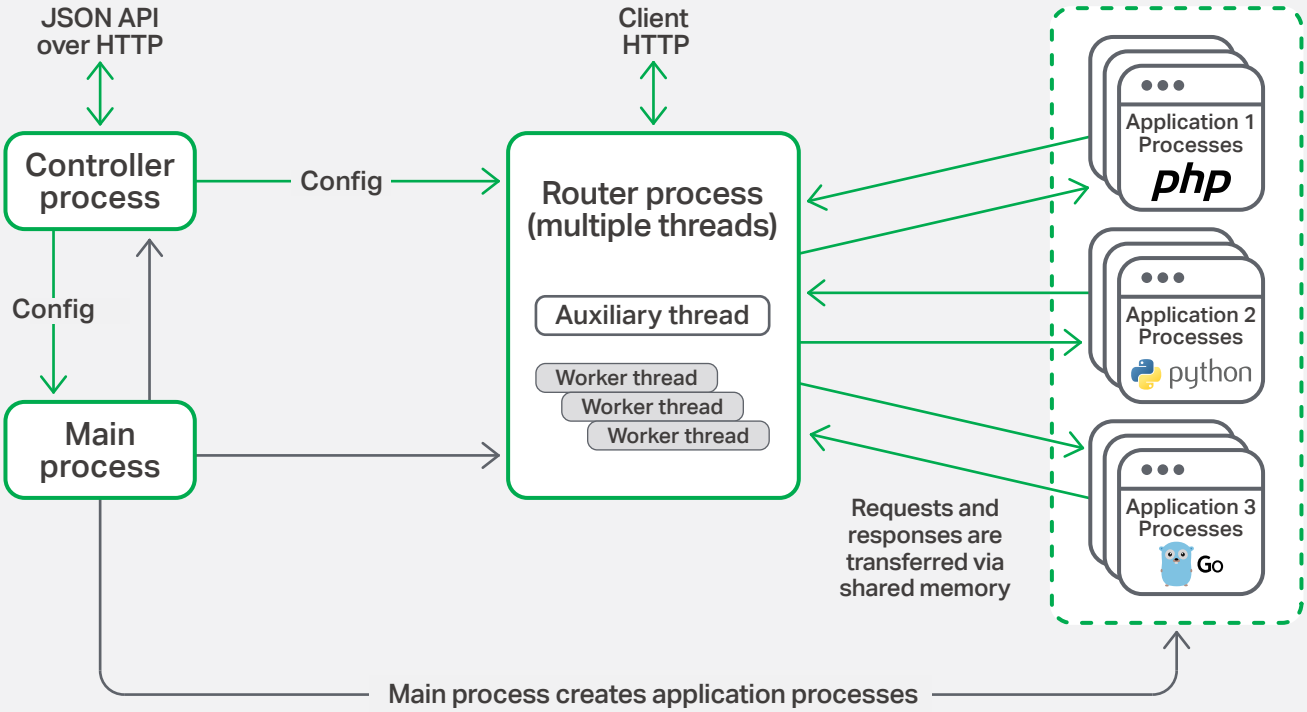
The *Controller process* is responsible for configuration. NGINX Unit is dynamically configured using a RESTful JSON API. You can upload the whole configuration at once, or just a part of it. NGINX Unit does not reload the whole configuration for every change, instead performing relevant changes directly in memory and reloading only the necessary parts. NGINX Unit configuration can be updated as frequently as needed without worrying about taking up additional system resources.

Supported languages:

- Go
- Node.js
- Perl
- PHP
- Python
- Ruby

The *Router process* interacts with clients. It accepts client requests and passes them to the application processes. It then gets responses back from the applications, and forwards them to the clients. Each worker thread in the Router process can simultaneously handle thousands of connections.

The *Application processes* run the application code. New application processes are created on demand by the Main process.



NGINX Unit architecture.

For more information, visit nginx.com or send us an email at nginx-inquiries@nginx.com