EBOOK

Secure, Scale, and Monitor Kubernetes with AWS and F5 NGINX

NGINX in the AWS Marketplace for Containers allows you to deliver high-performance, cloud-native apps.

Contents

Containers Versus Other Modern App Technologies

Microservices architectures – which generally make use of containers and Kubernetes – fuel business growth and innovation by reducing time-to-market for digital experiences. Whether alongside traditional architectures or on their own, these modern app technologies enable superior scalability and flexibility, faster deployments, and increased cost savings.

NGINX's 2020 survey of NGINX users found that 60% of respondents are using microservices in production, up from 40% in 2019, and containers are more than twice as popular as other modern app technologies.

Kubernetes and Culture

Kubernetes is the de facto standard for managing containerized apps, as evidenced by the Cloud Native Computing Foundation's (CNCF) 2020 survey, which found that 91% of respondents are using Kubernetes – 83% of them in production. When adopting Kubernetes, many organizations are prepared for substantial architectural changes but are surprised by the organizational impacts of running modern app technologies at scale. If you're running Kubernetes, you've likely encountered all three of these business-critical barriers: culture, complexity, and security.

Even as app teams adopt modern approaches like agile development and DevOps, they usually remain subject to Conway's Law, which holds that when organizations design systems they will do so in a way that re lects their own communication structure. In other words, distributed applications are developed by distributed teams that operate independently but share resources. While this structure is ideal for enabling teams to run fast, it also encourages the establishment of siloes. Consequences include poor communication (which has its own consequences), security vulnerabilities, tool sprawl, inconsistent automation practices, and clashes between teams.

Infrastructure Complexity Leads to Operational Debt

To implement enterprise-grade microservices technologies, organizations must piece together a suite of critical components that provide visibility, security, and traffic management.

Typically, teams use infrastructure platforms, cloud-native services, and open source tools to fill this need. While there is a place for each of these strategies, each has drawbacks that can contribute to complexity. And all too often different teams within a single organization choose different strategies to satisfy the same requirements, resulting in "operational debt." Further, teams choose processes and tools at a point in time and continue to use them regardless of the changing requirements around deploying and running modern microservices-driven applications using containers.

The CNCF Cloud Native Interactive Landscape is a good illustration of the complexity of the infrastructure necessary to support microservices-based applications. Organizations need to become proficient in a wide range of disparate technologies, with consequences that include infrastructure lock-in, shadow IT, tool sprawl, and a steep learning curve for those tasked with maintaining the infrastructure.

Security Considerations for Modern Apps

Security requirements differ significantly for cloud-native and traditional apps, because strategies such as ring-fenced security aren't viable in Kubernetes. The large ecosystem and the distributed nature of containerized apps means the attack surface is much larger, and the reliance on external SaaS applications means that employees and outsiders have many more opportunities to inject malicious code or exfiltrate information.

Further, the consequences outlined in the culture and complexity areas – tool sprawl, in particular – have a direct impact on the security and resiliency of your modern apps. Using different tools throughout your ecosystem to solve the same problem is not just inefficient – it creates a huge challenge for SecOps teams who must learn how to properly configure each component.

The Solution: Production-Grade Kubernetes

As with most organizational problems, the answer to overcoming the challenges of Kubernetes is a combination of technology and processes. We're going to focus on the technology component.

Because Kubernetes is an open source technology, there are numerous ways to implement it. While some organizations prefer to roll their own basic Kubernetes, many find value in the combination of flexibility, prescriptiveness, and support provided by services such as Amazon Elastic Kubernetes Service (Amazon EKS).

Kubernetes platforms can make it easy to get up and running; however, they focus on breadth of services rather than depth. So, while you may get all the services you need in one place, they're unlikely to offer the feature sets you need for true production readiness at scale. Namely, they don't focus on advanced networking and security, which is where we see Kubernetes disappointing many customers.

How Do You Make Kubernetes Production Grade?

To make Kubernetes production grade, you need to add three more components in this order:

1. A scalable ingress-egress tier to get traffic in and out of the cluster

This is accomplished with an Ingress controller, which is a specialized load balancer that abstracts away the complexity of Kubernetes networking and bridges between services in a Kubernetes cluster and those outside it. This component becomes production grade when it includes features that increase resiliency (for example advanced health checks and Prometheus metrics), enable rapid scalability (dynamic reconfiguration), and support self-service (role-based access control [RBAC]).

2. Built-in security to protect against threats throughout the cluster

While "coarse-grained" security might be sufficient outside the cluster, "fine-grained" security is required inside it. Depending on the complexity of your cluster, there are three locations where you may need to deploy a flexible web application firewall (WAF): on the Ingress controller, as a per-service proxy, and as a per-pod proxy. This flexibility lets you apply stricter controls to sensitive apps – such as billing – and looser controls where risk is lower.

Fine-grained access means granting limited AWS Lake Formation permissions to individual principals on AWS Glue Data Catalog resources, Amazon Simple Storage Service (Amazon S3) locations, and the underlying data in those locations. Coarse-grained means broader permissions on individual operations and on access to Amazon S3 locations.

3. A scalable east-west traffic tier to optimize traffic within the cluster

This third component is needed once your Kubernetes applications have grown beyond the level of complexity and scale that basic tools can handle. At this stage, you need a service mesh, which is an orchestration tool that provides even finer-grained traffic management and security to application services within the cluster. A service mesh is typically responsible for managing application routing between containerized applications, providing and enforcing autonomous service-to-service mutual TLS (mTLS) policies, and providing visibility into application availability and security.

Create a Unified "Single Pane of Glass" View of Your Infrastructure

When selecting the components discussed on the previous page, prioritize portability and visibility. Platform-agnostic components reduce complexity and improve security, with fewer tools for your teams to learn and secure, and easier shifting of workloads based on your business needs. The importance of visibility and monitoring is hard to overstate. Integrations with popular tools like Grafana and Prometheus create a unified “single pane of glass” view of your infrastructure, ensuring your team detects problems before they’re discovered by your customers.

In addition, there are other complementary technologies that aren’t necessarily required for production-grade Kubernetes but are integral parts of modern app development. For example, when organizations are ready to modernize traditional apps, one of the first steps is building microservices with an API gateway.

How NGINX Can Help

Our Kubernetes solutions are platform-agnostic and include the three components you need to enable production-grade Kubernetes:

- NGINX Ingress Controller as the ingress-egress tier
- NGINX App Protect as the WAF
- NGINX Service Mesh as the east-west tier.

These solutions can make Kubernetes your best friend by enabling you in four key areas:

1. **Automation – Get your apps to market faster and safer.**
   Deploy, scale, secure, and update apps using NGINX Ingress Controller's traffic routing and app onboarding capabilities paired with NGINX Service Mesh's automatic deployment of NGINX Plus sidecars.
2. **Security – Protect your customers and business from existing and emergent threats.**
   Reduce potential points of failure by deploying NGINX App Protect anywhere in the cluster while using NGINX Service Mesh and NGINX Ingress Controller to govern and enforce end-to-end encryption between services.
3. **Performance – Deliver the digital experiences your customers and users expect.**
   Effortlessly handle traffic spikes and security threats without compromising performance with NGINX solutions.
4. **Insight – Evolve your business and better serve customers.**
   Get targeted insights into app performance and availability from NGINX Ingress Controller and NGINX Service Mesh, with deep traces to understand how requests are processed across your microservices apps.

NGINX and AWS

You can now purchase the NGINX Plus-based version of NGINX Ingress Controller, with and without NGINX App Protect, directly from AWS Marketplace for Containers. With this addition, NGINX continues to strengthen our partnership with AWS, enabling you to deliver high-performance, cloud-native apps with ease.

NGINX Ingress Controller offers two options for making your Kubernetes deployment production-grade:

Option 1:

NGINX Ingress Controller—The first step towards making Kubernetes production-grade is the addition of an Ingress controller that increases resiliency, enable rapid scalability, and support self-service. Try free for 30 days.

Option 2:

NGINX Ingress Controller with NGINX App Protect—As your Kubernetes apps mature, you'll need to enhance security to reduce the risk of breaches and single points of failure. The deployment of a flexible web application firewall (WAF) at the point of Ingress moves the WAF closer to your Kubernetes apps and eliminates the need for a separate WAF device. Try free for 30 days.

Check out this episode of ***This is My Architecture*** video series from AWS to learn how NGINX Ingress Controller routes and protects traffic with Amazon EKS, as well as how to use Infrastructure-as-Code with Amazon Elastic Container Registry (Amazon ECR) to build an automated pipeline for application deployment and versioning.

Learn More About NGINX on AWS

F5 offerings are backed up by F5's Kubernetes experts:

Both NGINX Ingress Controller and NGINX Ingress Controller with NGINX App Protect include include premium support. The Premium Edition includes access to 24x7 support with a 1-hour response time when purchased on AWS Marketplace.

There are a number of resources available for you to learn more:

- **Review the sizing guide** – Get an overview of the performance levels you can achieve on specific AWS instance types, along with the estimated monthly costs.
- **Attend the webinar** – In ***Strengthen Security and Traffic Visibility on Amazon EKS with NGINX***, we chat with Zipwhip's principal architect to learn how the company achieved 99.99% uptime by moving to Kubernetes with NGINX Ingress Controller and AWS.
- **Watch demos** – In this YouTube playlist, we show numerous ways to improve performance, resiliency, and security with NGINX on AWS.

If you're ready to give it a try:

Request a private offer
Save up to 20% annually by negotiating a custom price for NGINX Ingress Controller with NGINX App Protect WAF.

Try NGINX Plus on AWS
In addition to our new container offers, you can also purchase NGINX Plus (with and without NGINX App Protect WAF) from the AWS Marketplace.

TRY IT NOW FOR FREE