DECEMBER 18, 2025

The State of Trusted Open Source: December 2025

Ed Sawma, VP of Product Marketing, and Sasha Itkis, Product Analyst

At Chainguard, we have a unique view into how modern organizations actually consume open source software and where they run into risk and operational burdens. Across our growing customer base and extensive catalog of over 1800 container image projects, 148,000 versions, 290,000 images, and 100,000 language libraries, and almost half a billion builds, we can see what teams pull, deploy, and maintain day-to-day, along with the vulnerabilities and remediation realities that come hand in hand.

That's why we created *The State of Trusted Open Source*, a quarterly pulse on the open source software supply chain. As we analyzed anonymized product usage and CVE data, we noticed common themes around what open source engineering teams are actually building with and the risks associated.

Here's what we found:

- **AI is reshaping the baseline stack:** Python led the way as the most popular open source image among our global customer base, powering the modern AI stack.
- **Over half of production happens outside of the most popular projects:** Most teams may standardize on a familiar set of images, but real-world infrastructure is powered by a broad portfolio that extends far beyond the top 20 most popular, what we refer to in this report as longtail images.
- **Popularity doesn't map to risk:** 98% of the vulnerabilities found and remediated in Chainguard images occurred outside of the top 20 most popular projects. That means the biggest security burden accumulates in the less-visible part of the stack, where patching is hardest to operationalize.
- **Compliance can be the catalyst for action:** Compliance takes many forms today: from SBOM and vulnerability requirements to industry frameworks like PCI DSS, SOC 2, and regulations like the EU's Cyber Resilience Act. FIPS is just one example, focused specifically on U.S. federal encryption standards. Even so, 44% of Chainguard customers run a FIPS image in production, underscoring how frequently regulatory needs shape real-world software decisions.
- **Trust is built on remediation speed:** Chainguard eliminated Critical CVEs, on average, in under 20 hours.

Before we dive in, a note on our methodology: This report analyzes 1800+ unique container image projects, 10,100 total vulnerability instances, and 154 unique CVEs tracked from September 1, 2025,

through November 30, 2025. When we use terms like “top 20 projects” and “longtail projects” (as defined by images outside of the top 20), we’re referring to real usage patterns observed across our customer portfolio and in production pulls.

Usage: What teams actually run in production

If you zoom out, today’s production container footprint looks exactly like you’d expect: foundational languages, runtimes, and infrastructure components dominate our most popular list.

Most popular images: AI is reshaping the baseline stack

Across all regions, the top images are familiar staples: Python (71.7% of customers), Node (56.5%), nginx (40.1%), go (33.5%), redis (31.4%), followed by JDK, JRE, and a cluster of core observability and platform tooling like Grafana, Prometheus, Istio, cert-manager, argocd, ingress-nginx, and kube-state-metrics.

This indicates that customers operate a portfolio of critical building blocks – including languages, gateways, service mesh, monitoring, and controllers – that collectively form the foundation of their business.

It’s not surprising to see Python leading the way globally, as the default glue language for the modern AI stack. Teams typically standardize on Python for model development, data pipelines, and increasingly for production inference services as well.

The most popular image projects

Chainguard

1.	python	71.7%	11.	cert-manager-cainjector	19.9%
2.	node	56.5%	12.	cert-manager-webhook	19.9%
3.	nginx	40.1%	13.	cert-manager-controller	19.9%
4.	go	33.5%	14.	argocd	19.3%
5.	redis	31.4%	15.	ingress-nginx-controller	18.9%
6.	jdk	28.3%	16.	kube-state-metrics	18.6%
7.	jre	26.4%	17.	istio-proxy	18.6%
8.	grafana	23.0%	18.	kubectl	18.3%
9.	prometheus	22.7%	19.	prometheus-node-exporter	18.0%
10.	istio-pilot	20.5%	20.	aspnet-runtime	17.7%

Most popular by region: Similar foundations, different longtail mix

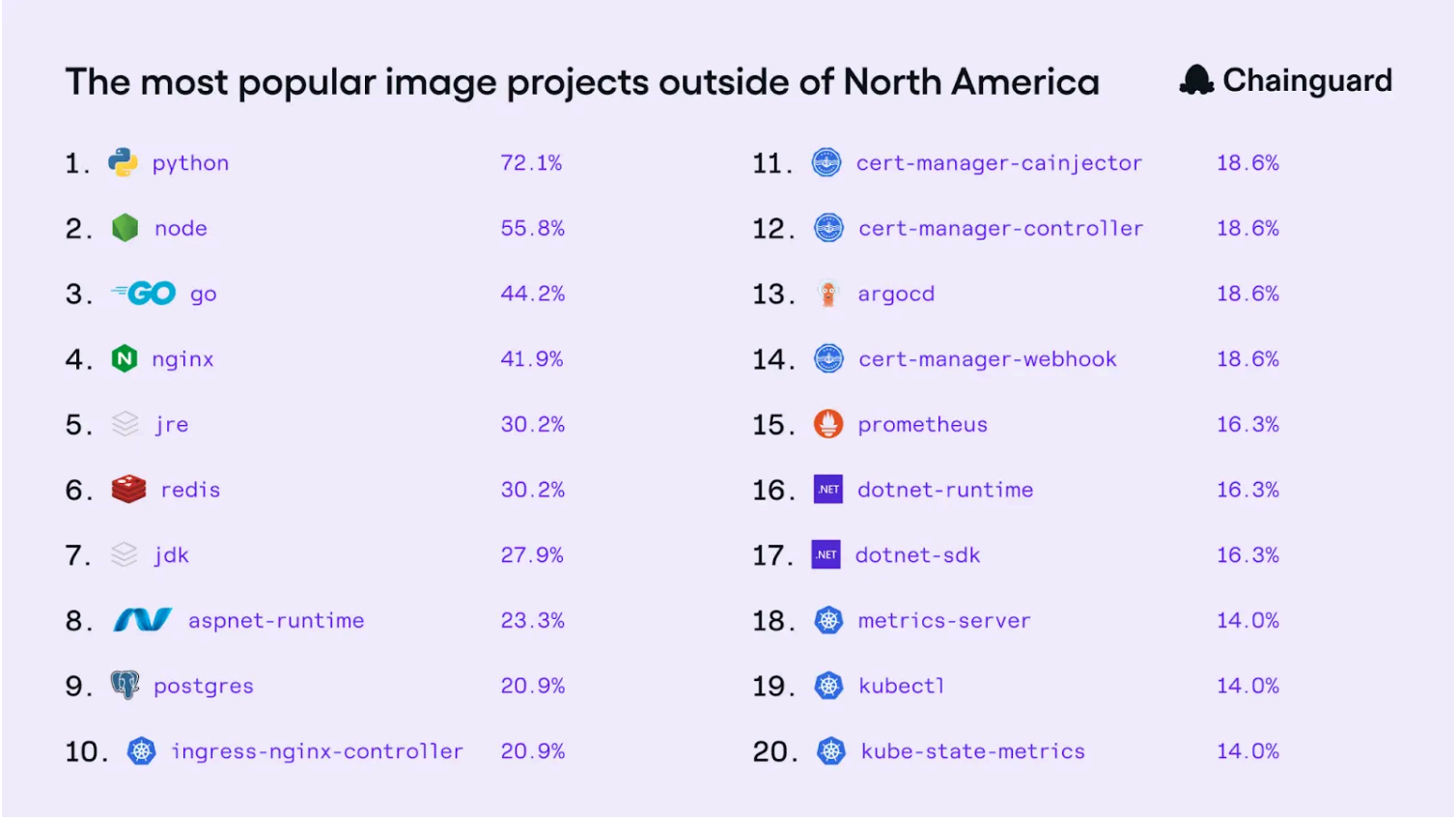
North America shows a broad and consistent set of default production building blocks: Python (71.7% of customers), Node (56.6%), nginx (39.8%), go (31.9%), redis (31.5%), plus strong penetration of Kubernetes ecosystem components (cert-manager, istio, argocd, prometheus, kube-state-metrics, node-exporter, kubectl). Notably, even utility images like busybox show up meaningfully.

Outside North America, the same core stack appears, but the portfolio spreads differently: Python (72% of customers), Node (55.8%), Go (44.2%), nginx (41.9%), and a noticeable presence of .NET runtimes (aspnet-runtime, dotnet-runtime, dotnet-sdk) and PostgreSQL.

The most popular image projects in North America

Chainguard

1.	python	71.7%	11.	cert-manager-controller	20.1%
2.	node	56.6%	12.	cert-manager-webhook	20.1%
3.	nginx	39.8%	13.	cert-manager-cainjector	20.1%
4.	go	31.9%	14.	istio-proxy	19.7%
5.	redis	31.5%	15.	argocd	19.4%
6.	jdk	28.3%	16.	kube-state-metrics	19.4%
7.	jre	25.8%	17.	prometheus-node-exporter	19.0%
8.	grafana	24.4%	18.	kubectl	19.0%
9.	prometheus	23.7%	19.	ingress-nginx-controller	18.6%
10.	istio-pilot	21.5%	20.	busybox	17.6%



The longtail of images is crucial to production, not edge cases

Chainguard's most popular images represent only 1.37% of all available images and account for roughly half of all container pulls. The other half of production usage comes from everywhere else: 1,436 longtail images that make up 61.42% of the average customer's container portfolio.

In other words, half of all production workloads run on longtail images. These aren't edge cases. They're core to our customers' infrastructure. It's relatively straightforward to keep the top handful of images polished, but what trusted open source requires is maintaining that security and velocity across the breadth of what customers actually run.

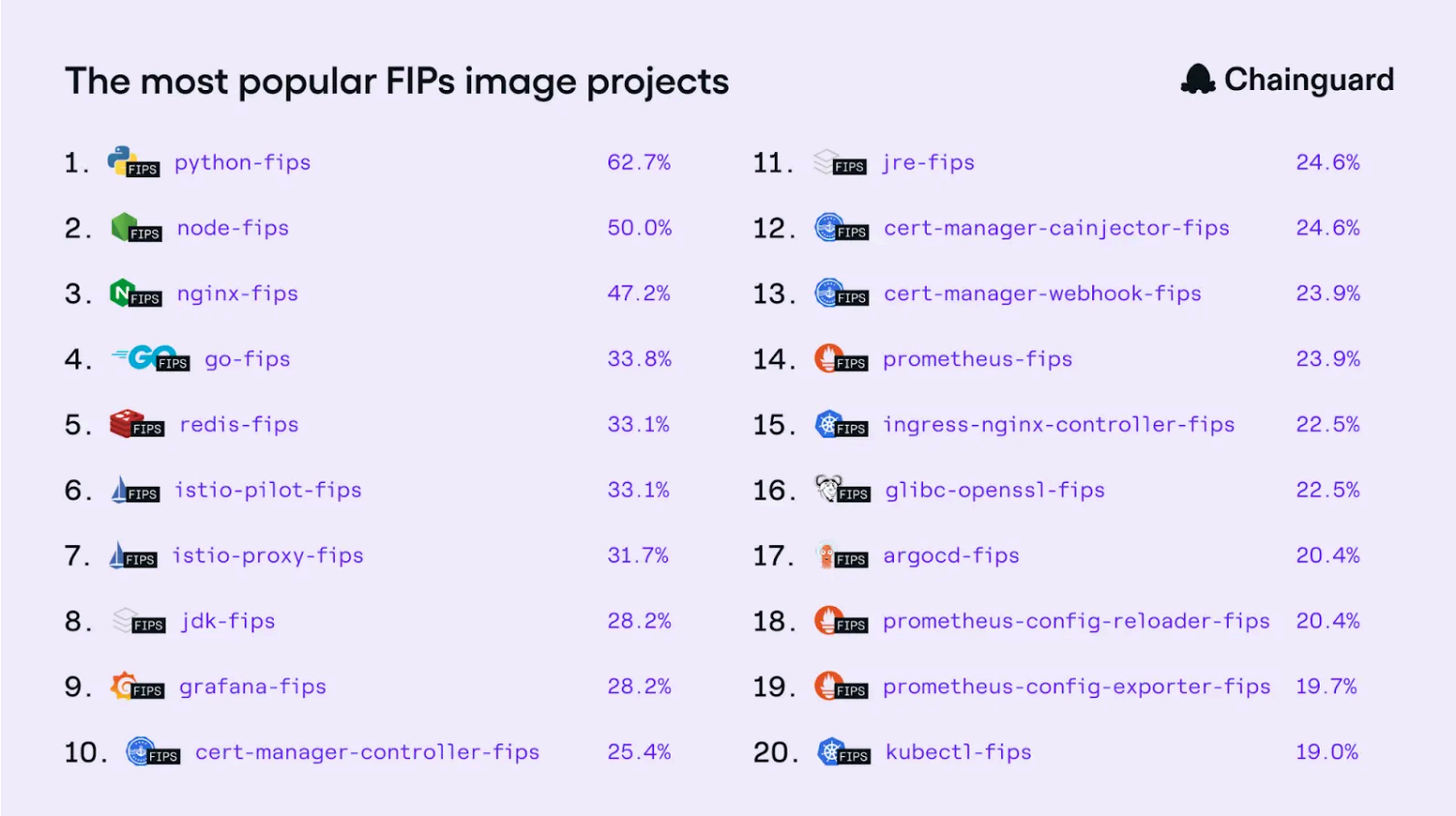
FIPS usage: Compliance is a catalyst for action

FIPS encryption is an essential technology in the compliance landscape, focused on satisfying U.S. federal encryption requirements. And it offers a useful window into how regulatory pressure drives adoption. In our data, 44% of customers run at least one FIPS image in production.

The pattern is consistent: when working within compliance frameworks like FedRAMP, DoD IL-5, PCI DSS, SOC 2, CRA, Essential Eight or HIPAA, teams need hardened, trusted open source software that mirrors their commercial workloads. Our most used FIPS images align with the broader portfolio, simply with cryptographic modules strengthened for audit and verification.

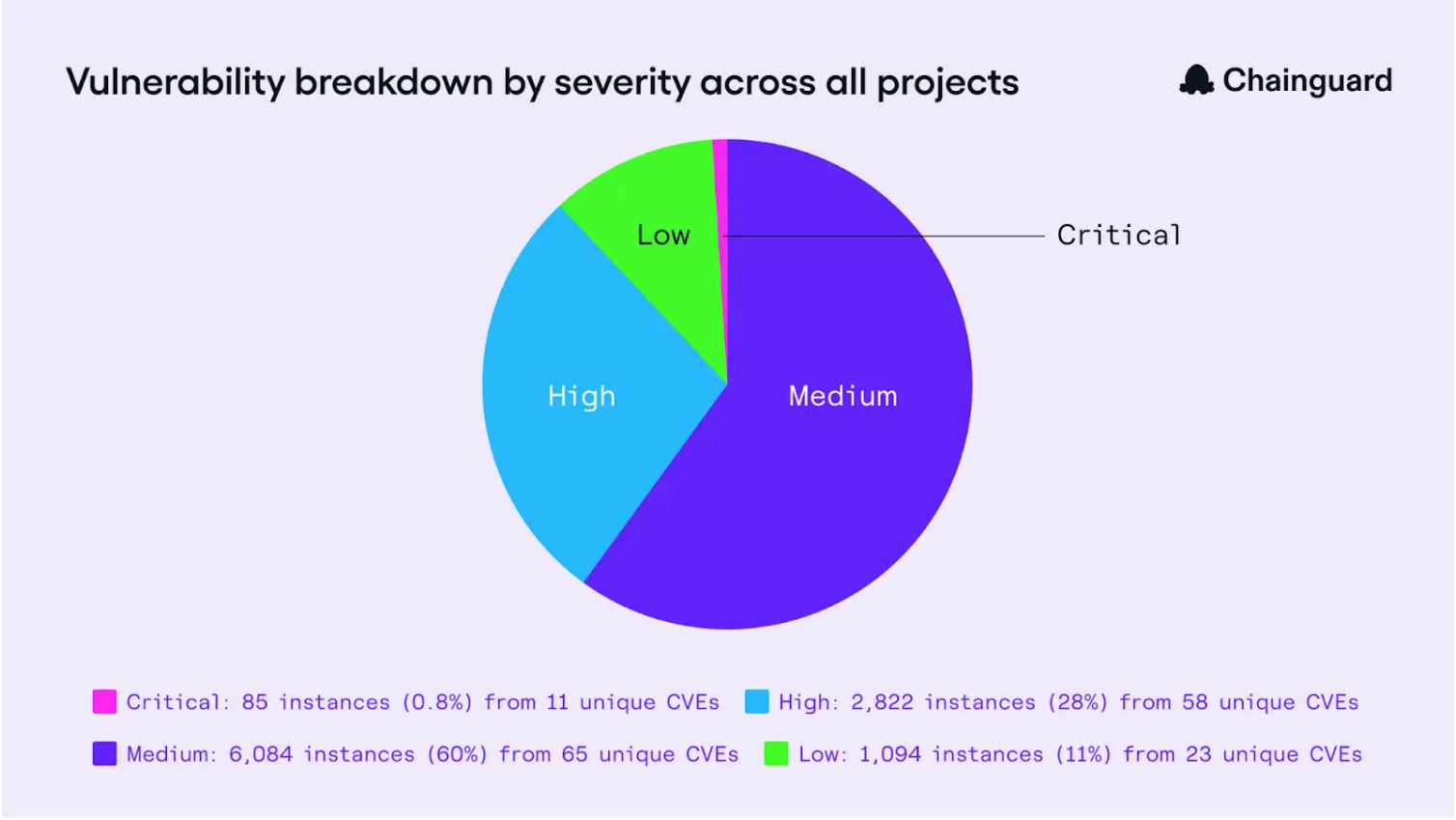
Top FIPS image projects include Python-fips (62% of customers with at least one FIPS image in production), Node-fips (50%), nginx-fips (47.2%), go-fips (33.8%), redis-fips (33.1%), plus platform components like istio-pilot-fips, istio-proxy-fips, and cert-manager variants. Even supporting libraries and crypto foundations show up, such as glibc-openssl-fips.

FIPS is not the whole story, but it illustrates a broader truth: compliance is a universal driver, emphasizing the need for trusted open source across the entire software stack.



CVEs: Popularity doesn't map to risk

When looking across Chainguard's catalog of images, risk is overwhelmingly concentrated outside of the most popular images. Of the CVEs we remediated in the past three months, 214 occurred in the top 20 images, accounting for only 2% of the total CVEs. Go beyond those top images, and you'll find the other 98% of CVEs we remediated (10,785 CVE instances). That's 50 times the number of CVEs in the top 20 images!



The largest volume of CVEs are categorized as Medium, but operational urgency often stems from how quickly Critical and High CVEs are addressed, and whether customers can rely on that speed across their entire portfolio, not just the most common images.

Trust is built on remediation speed

For us, trust is measured in time-to-fix, and we know this is most important when it comes to Critical CVEs. During the three-month period analyzed, our team achieved a less than 20-hour average remediation time for Critical CVEs, with 63.5% of Critical CVEs being resolved within 24 hours, 97.6% within two days, and 100% within three days.

In addition to Critical CVE remediation, our team addressed High CVEs in 2.05 days, Medium CVEs in 2.5 days, and Low CVEs in 3.05 days, notably faster than our SLAs (seven days for Critical CVEs and 14 days for high, medium, and low CVEs).

And this speed isn't confined to the most popular packages. For every single CVE remediated in a top 20 image project, we resolved 50 CVEs in less-popular images.

That longtail is where most of your real exposure hides and it can feel hopeless trying to keep up. Most engineering organizations simply can't allocate resources to patch vulnerabilities in packages that fall outside their core stack, but our data makes it clear that you have to secure the "quiet majority" of your software supply chain with the same rigor as your most critical workloads.

A new baseline for trusted open source

Across our data, one takeaway stands out: modern software is powered by a wide, shifting portfolio of open source components, most of which live outside the top 20 most popular images. That's not where developers spend their time, but it's where the bulk of security and compliance risk accumulates.

This creates a concerning disconnect: it's rational for engineering teams to focus on the small set of projects that matter most to their stack, but the majority of exposure sits in the vast set of dependencies they don't have the time to manage.

That's why breadth matters. Chainguard is built to absorb the operational burden of the longtail, providing coverage and remediation at a scale that individual teams can't justify on their own. As open source supply chains grow more complex, we'll continue to track usage patterns and shine a light on where risk truly resides, so you don't have to fight the battle against the longtail alone.

Ready to get started with the trusted source for open source? Contact us to learn more.

SHARE THIS ARTICLE

Related articles

OPEN SOURCE DEC 29, 2025

The only rule: Don't look at the code

PATRICK SMYTH, PRINCIPAL DEVELOPER RELATIONS ENGINEER

OPEN SOURCE DEC 22, 2025

Fork Yeah: We're keeping ingress-nginx alive

ADRIAN MOUAT, STAFF DEVELOPER RELATIONS ENGINEER