HashiCorp
Nomad

Nomad, Kubernetes, and a Pragmatic Look at Choosing Orchestrators

WHITEPAPER

Overview

While Kubernetes is leading the orchestration market, this whitepaper explains why a growing number of customers choose Nomad as an alternative or use each tool where most appropriate.

Back in March 2019, Matthias Endler from Trivago posted a blog "Maybe You Don't Need Kubernetes," explaining his company's decision to use HashiCorp Nomad for orchestration instead of Kubernetes. His article garnered more than 500 comments on Hacker News and reminded the community that even when one technology seems to dominate headlines, one tool never fits all.

In the past two years, we have seen a big rise in the number of enterprises that deploy Nomad in production. There are remarkable stories from companies running Nomad across the world: in traditional on-premises datacenters, cloud instances, agricultural smart farms, industrial factories, the world's largest ground-based solar telescopes, and even biology laboratories. Yet no matter the environment, organizations choose Nomad as an alternative to Kubernetes for its two unique core strengths:

- Simplicity in usage and maintainability
- Flexibility to deploy and manage containerized and non-containerized applications

Simplicity at Any Scale

Operating as a single lightweight binary (roughly 50MB as of the 1.0 release), Nomad excels on-premises and at the edge, providing the same ease-of-use as it does in the cloud. Its architectural simplicity, native federation capabilities, and operator-friendly design enable companies to scale and manage an orchestrator with little operational overhead.

" Nomad is easy to cluster up. We converted our Kubernetes deployment manifest to Nomad job files, then tested it. And since it's a single binary, it's simple to configure to our specific needs, which eliminates much of the complexity we faced with Kubernetes. More importantly, Nomad's agnostic infrastructure resource pool and automated workflows let us deploy and manage our containers and apps across on-prem and any private or public cloud environment, which dramatically expands our datacenter options while still meeting our data residency obligations."

— From AmpleOrganics. Read the full article here.

" I'm a complete beginner when it comes to distributed computing and orchestration. Nomad virtually eliminates barriers to entry for developers who don't have cloud computing expertise and makes it really easy to connect to the cluster, configure it, and run my jobs while having full visibility into the jobs' status so I can restart them if need be."

— From Autodesk. Read the full article here.

Organizations are capable of translating the simplicity of Nomad into real business results. Our customer interviews showed that it takes teams 1 to 3 weeks on average to get Nomad from a technical proof of concept into production, no matter the company size.

Its simplicity also carries over into maintainability. It is quite common to hear stories about lean teams of 1 to 4 people being able to service hundreds of developers and applications, and achieve high uptime with self-hosted Nomad across organizations. Some of the recent great examples include:

- Cloudflare, who routes 10% of the world's internet traffic
- Roblox, who serves more than 150 million monthly active video game players globally
- Pandora, who streams audio content to more than 60 million users monthly

Bridging Legacy Applications with Modern Orchestration

While seeing the growing trends of containerization and app modernization, we observe that most companies are taking an incremental approach and require multi-year efforts to fully migrate. Nomad's flexible workload support beyond Docker containers allows companies to migrate at their own pace and be able to maintain a single, unified deployment workflow during the whole journey.

Another key benefit is to bring modern orchestration capabilities to legacy applications without the need for rewrites, such as zero-downtime deployment, improved resource utilization, and automation with a self-service experience.

" Our system runs several Docker containers as well as Java applications that access hardware resources and require root. Nomad makes it easy to add machines like mass spectrometers by first fingerprinting through the exec driver, and then rolling out more specific drivers. This gives us a unified control plane, enabling hardware and driver rollouts using vendor's drivers — be it a centrifuge, incubator, or mass spectrometer."

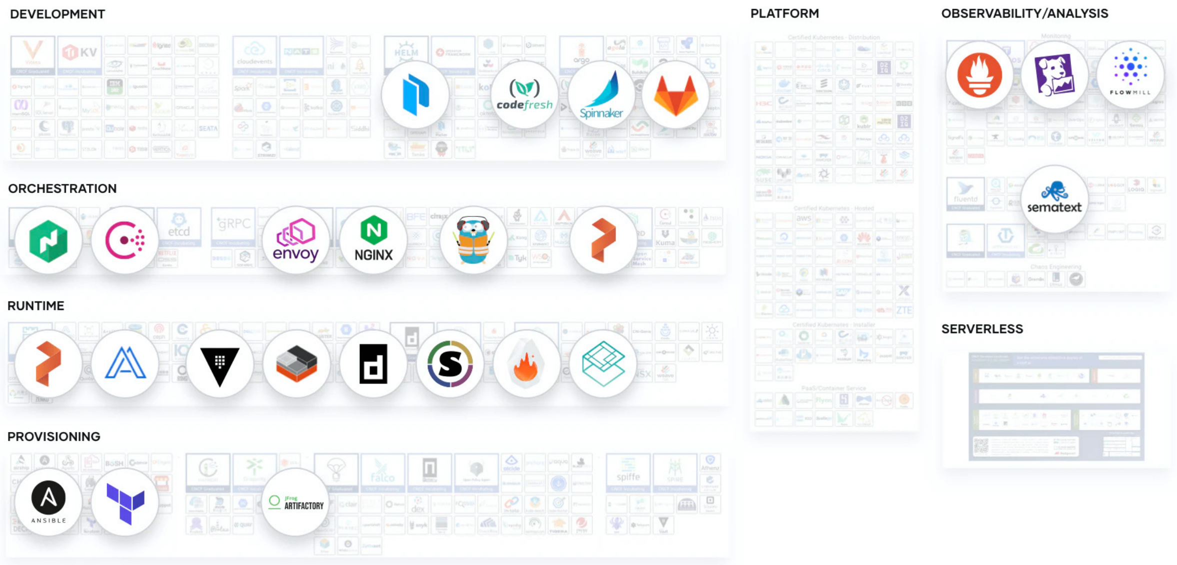
— From Radix. Read the full article here.

" A large portion of our applications are Windows-based, so we need both Windows and Linux support. Although we do prefer running containers, we don't necessarily want a hard requirement to have to use them and we like the idea of directly running applications on VMs if the use case calls for it. We wanted to make improvements to the current workflow without massive and time-consuming application rewrites. Ultimately we chose Nomad because it met all of our requirements and made the most sense to our environments."

— From Q2. Read the full article here.

A Prescriptive Approach to Ecosystem

The ecosystem can be a deciding factor for companies when it comes to choosing orchestrators. While Nomad's strengths lie in simple, flexible core scheduling, it will be hard to replicate the ever-growing ecosystem built around Kubernetes. The CNCF diagram showcases the breadth of the Kubernetes landscape. With additions like Helm charts, it is much easier for Kubernetes users to run some popular open source frameworks than it is on Nomad today.

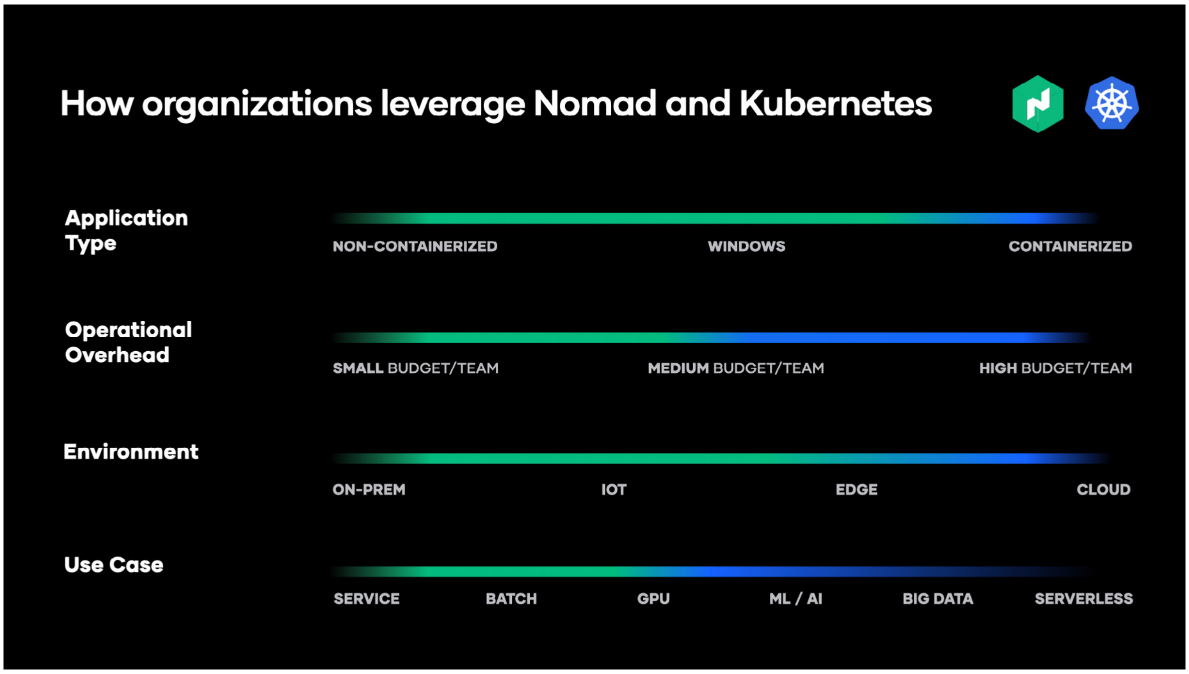


The goal of Nomad is to build a simpler, leaner, and more prescriptive path to the ecosystem. By building for depth instead of breadth, our goal is to strategically build integrations with leading partners from each category. In the past six months, Nomad has shipped integrations with Prometheus, held webinars on how to integrate with GitLab CI, added CSI and CNI support, and built our own application and cluster autoscaler for AWS. This is an area of improvement that Nomad invests in significantly today and plans to continue into the future.

Supplement to Kubernetes

As the orchestrator market matured rapidly over the past few years, organizations have gradually moved beyond hype-driven deployments. More organizations choose orchestrators based on their merit and how they fit into their specific use case and project. They leverage technologies based on their particular needs and constraints, such as the nature of their applications, infrastructure environments, technical competencies, team sizes, budgets, SLAs, etc.

Therefore, medium- to large-sized enterprises run into challenges when trying to standardize hundreds or thousands of software developers and operators onto one single orchestrator (Kubernetes, Nomad, or Docker Swarm) given that no scheduler today fits all applications, environments, projects, and teams. Companies today such as Intel, GitHub, and Autodesk have multiple products and business units organically running Nomad and Kubernetes to supplement each other.



They leverage each scheduler's strengths: Kubernetes for its cutting edge-ecosystem and Nomad for simple maintenance and flexibility in non-containerized applications.

These are the characteristics we see in teams that typically adopt self-hosted Kubernetes:

- Greenfield use-cases such as machine learning (ML), serverless, and big data that require the Kubernetes ecosystem and Helm chart
- High budget and full-time staffing to maintain Kubernetes
- High-profile projects with significant investment and long-term timeline (multi-year)
- Deploying and managing new, cloud-native applications
- Public cloud environment such as AWS, GCP, Azure

Characteristics of teams that typically adopt Nomad:

- Run a mix of containerized and non-containerized workloads (Windows, Java)
- Small/medium-sized teams with limited capacity to maintain an orchestrator
- Deploying and managing core, existing applications
- On-premises environment, or hybrid environments
- Require simplicity to move fast and fulfill business needs with hard deadlines

We continue to see that small and medium-sized enterprises continue to standardize on a single orchestrator given the natural staffing and organizational constraints.

In Conclusion

The multi-orchestrator deployments we are seeing in enterprises will continue to emerge as a pattern in the future, where teams see Nomad and Kubernetes used in areas where their specific strengths are needed. It's not a one-size-fits-all conversation about "Why you should use Kubernetes" or "Why Nomad," but rather *when* should each tool be used.

HashiCorp

www.hashicorp.com