Answering Calls to Move Away from VMware —The Tight Relationship Between Kubernetes and Virtual Environments

3.1 The VMware Issue Shaking the Industry

The debate around calls to move away from VMware has been raging since last year. Some say VMware license prices have surged and organizations should consider switching to other virtualization platforms. Others say vendor logic is being given precedence while existing customers are left neglected. Is this really true?

While various conditions make it difficult to generalize, a dispassionate look across the entire virtualization market would indicate that the company's actions are aimed at positioning suite packages, which have become the main battleground, as VMware's core offering as well, in an attempt to guide customers toward higher value-added, more competitive packages. Depending on how you look at it, you could say it is now less strategically differentiated from competitors. Not to favor any specific competitor's products, but I think it's reasonable to describe this as rationalization, whether in terms of lineup or pricing. The fact remains, however, that users of simple packages, most likely those in the volume zones, have borne the brunt and been hit by cost increases.

This leads people to consider taking some form of action, but it's highly questionable whether simply focusing on license fees and considering a switch to alternative products would achieve the desired outcome. After all, it's not as if VMware's appeal as a product has diminished. It may be a different story if version upgrades had introduced unwanted feature changes or problems that hurt customer trust, but this is purely a cost issue. Even if options that offer advantages in terms of license fees can be found, they would still need to be on par with VMware, which remains the virtualization market leader, in terms of functionality, stability, future prospects, and many other factors. Plus, moving away from VMware involves significant migration costs. And such costs might even offset the expected cost reduction benefits.

In other words, to successfully move away from VMware, one needs to think about what sort of platform would be even better than VMware, rather than fixating on the costs. Achieving both could not only deal with the licensing issue but also bring a competitive advantage to the business by modernizing the platform.

One answer to this extremely difficult challenge that IIJ is working on is the migration from VMware to Kubernetes. By using open source to the maximum extent and positioning our internally developed Kubernetes distribution, IKE (IIJ Kubernetes Engine), as a VMware alternative, we are significantly reducing license costs while also achieving operational efficiency and quality improvements unattainable at the IaaS layer.

3.2 Kubernetes Swallowing Up All Kinds of Workloads

Even engineers fairly well versed in cloud-native technology mostly consider Kubernetes a container orchestrator, I think. Indeed, it surely is the case that the vast majority of Kubernetes deployments are being used for container orchestration. But are you aware that in recent years, Kubernetes is gradually being adopted more and more not just for containers but for VMs as well? If you do know this, you are probably already quite in the know. I consider myself to be a cloud-native advocate, and even I felt that, while the technology is maturing, it would still be some time before we'd see adoption in production environments.

And yet, amid all this debate sparked by the VMware situation about reconsidering what platforms are being used, the tide seems to have turned dramatically, with Kubernetes having suddenly emerged as the leading potential successor. I'm not only talking about the mood here at IIJ. This is a trend I sense across the entire IT industry.

The reasons Kubernetes has come to attract attention as a VMware alternative are too numerous to list. While there certainly are technical reasons, I feel reliability plays an even bigger role. For example, it's an open source project operated in a vendor-neutral manner by the Linux Foundation, and there is little concern about vendor lock-in. It's widely supported across the entire IT industry, by both hardware and software vendors, with a rapidly growing ecosystem. As a result, much like Linux in the server OS space, Kubernetes has come to be regarded as the de facto standard for orchestrators, something organizations can commit to for long-term investment. It is this confluence of factors that has earned it trust from users.

That said, if Kubernetes's role had remained limited to being a container orchestrator, it might never have been regarded as mainstream. While container adoption is increasing rapidly on server-side systems, if one asks whether VMs or containers are the primary tool, the answer, at present, is clearly VMs. But on the other hand, it seems unlikely that platforms only capable of managing VMs will continue to be the mainstream forever. Given its ability to handle VMs and containers as equivalent workloads, it seems inevitable that Kubernetes would attract attention.

3.3 IKE, IIJ's Kubernetes Platform

IIJ is gradually migrating from VMware to Kubernetes, and what enabled us to select Kubernetes as our means of moving away from VMware at such an early stage is that we already had sufficient operational experience with Kubernetes as a container platform. If the move away from VMware had spurred us to tackle Kubernetes operations for the first time, we may have hesitated quite a bit.

That's because Kubernetes operational knowhow isn't something you can acquire overnight. Kubernetes is often referred to as the OS for the cloud era since it mediates system resources like servers, networks, and storage. Moreover, unlike OSes that control resources within the confines of a single server, a single Kubernetes instance can manage all manner of large-scale systems housed in a data center so that they are coordinated with each other. To accomplish this, Kubernetes abstracts away the physical systems (hides the physical divisions between them), making the data center appear as if it were one giant resource pool. And users perform all operations by issuing commands through Kubernetes, just as they would with a public cloud. This is indeed one of the motivations for using Kubernetes, but when it comes to designing and operating such abstracted systems, even experienced engineers may find that their existing knowledge is not enough. On the other hand, becoming well-versed in Kubernetes alone only confers operational abilities, not the ability to manage systems. Even with Kubernetes, you still ultimately end up having to deal with hardware control, so unless you understand how Kubernetes operations are reflected in the actual systems, you may find it difficult to provide sufficient quality of operations.

While similar issues exist with virtualization platforms like VMware, Kubernetes is even more complex due to its higher level of abstraction.

IIJ, meanwhile, introduced Kubernetes as a service platform back in 2018. At the time, we were using Kubernetes v1.9, which was a far simpler system than what we have today. The 23 upgrades since then bring us up to v1.32. Over the intervening time, we have also upgraded or replaced many of the plugins and controllers we were initially running, leaving barely a trace of the components of the original Kubernetes cluster. Nevertheless, the many workloads that have been running on this Kubernetes cluster since the early days continue to operate stably.

This is an extremely important point. It means that, even while countless features have been added to Kubernetes with each minor version upgrade, not once in seven years has there been a fatal loss of compatibility. Moreover, the fact that the impact on systems running on Kubernetes has been minimal despite implementations, both hardware and software, having changed dramatically at the lower levels is evidence of the effectiveness of Kubernetes's abstraction. This speaks to Kubernetes's high level of continuity, stability, extensibility, and future potential, and we can no doubt expect that to continue ahead.

That said, this didn't happen by accident. It is the result of making upgrades in an appropriate manner based on a detailed understanding of the content of each upgrade and the scope of its impact. The experience gained through that process has been tremendously helpful in our move away from VMware. Of course, there are any number of ways to acquire Kubernetes operational skills in less time, but it still takes some time to gain confidence.

3.4 The Networking Challenges of VMs on Kubernetes

While we had accumulated the requisite knowledge, we actually only started deploying VMs on Kubernetes at IIJ a year ago. When we began our evaluation, we were considering separating our infrastructure into Kubernetes for containers and Kubernetes for VMs, but we quickly realized this sort of thing was completely unnecessary. Indeed, one could even say that a real advantage of Kubernetes is the ability to mix and match containers and VMs according to the use case and build systems that combine both. That's how compelling the environment made possible by KubeVirt is. At the risk of

oversimplifying, most Kubernetes functionality applies to VMs as well if you mentally substitute VirtualMachine (the resource used to launch VMs) for Pod (the resource used to launch containers).

Treating Pods and VirtualMachines as equivalent workloads provides substantial benefits.

- As Kubernetes evolves, you enjoy the benefits for both containers and VMs without doubling your effort.
- For on-premises Kubernetes, you can maintain high utilization rates since there's no need to split infrastructure for containers and VMs.
- Pods and VirtualMachines attach to the same pod network, making interoperability a breeze.
- Replacing systems built as VMs with containers is easy. This is useful as a migration path from VMs to containers

Yet Kubernetes as a path away from VMware is not all smooth sailing. In many existing Kubernetes environments, the common pattern is to attach all containers and VMs to a single pod network per cluster. By contrast, VM-centric environments often provision multiple L2 networks and assign independent networks for each application. While you can provision equivalent networks on Kubernetes, this is not yet common practice, and L2 implementations frequently depend on the underlying infrastructure or proprietary networking products, so careful evaluation is needed. Where complex network topologies are required, simply extending an existing Kubernetes cluster may not suffice; you may need a Kubernetes cluster dedicated to the VMware exit.

That said, it's encouraging that the substantive challenges are largely confined to networking. For users, the skills needed to use VMs are truly minimal. Only a small amount of training is needed to get started. While it's a different story on the operations side, if the burden falls mainly on the platform engineers, that's actually gratifying because it's exactly what we're here for.

Our move away from VMware is still a work in progress, but there's no doubt that Kubernetes can be one of the answers. That said, requirements obviously vary widely by workload, and it's equally true that there can be multiple right choices. To deliver value to our customers through our services, we aim to set aside preconceived notions and choose the best option for each case.

Keisuke Taguchi
General Manager, SRE Promotion Department, Network Division, Network Service Business Division, IIJ
Mr. Taguchi has been involved in the launch of numerous services including email, DNS, server hosting, and cloud IaaS services. In recent years, he has drawn on his past experience to establish the platform engineering department, nurturing it into a platform that hosts over 100 services and projects. He is a strategist who believes staying abreast of changes in markets and technology and continuously updating one's knowledge and skills is the key to business success.