Tero Ala-Hulkko

**Virtualization**

Virtualization can be used to optimise hardware utilization. Virtualization can take many forms such as virtual machines and containers. This essay explores the ways in which virtualization is used on servers, how it came to be and where is the industry today.

## History of virtualization

Before virtualization was implemented, each server typically ran a single operating system and a single application. This meant that a lot of an organization’s mainframe processing power was wasted. And since organizations tend to need more than one application, you needed several physical servers.

### Virtualization emerges

In the 60’s IBM developed the first mainframe with time-sharing capabilities. This allowed mainframes to support multiple users simultaneously. Time-sharing works by assigning milliseconds of CPU time for each user. Because mainframes were so powerful, this could power several users.

A few years later Cambridge Scientific Center developed a system that could support full virtualization. Soon it was implemented on to a product system, as source-code, without support. In 1972, the first mainframe with hypervisor access became available for purchase.

### Containers

Another important milestone in virtualization is the emergence of containers. Containers allow splitting computers into several chunks that perform as a complete operating system. Containers or guests, use the computing power of the main computer, the host computer. Compared to virtualization, containers are a lighter weigh solution. Another one of the benefits of containerization is the portability of software.

The main idea behind containers was developed by Bill Joy in 1979. He created a Linux operation called chroot. Chroot allows the operating system to corner off certain parts of the system by setting a specific root location. While chroot lacks many of the features of modern containers, such as helping with portability, not offering resource management tools, and having a very limited isolation, modern containers history can be traced directly back to chroot.

Containers did not see widespread adoption for a long time. They developed further in the late 90’s when BSD jails started to be developed. They offered a more comprehensive containerization. Many of the features we see today were developed.

Containers had a niche role in software architecture until 2013 when Docker was released. Docker has further improved upon several features in the container world. However, the biggest improvement it offered was simplifying the container and standardized container images. Standardized images made the container more portable and deployable.

### Container orchestration

For all the good containers did, they also brought several disadvantages. These include, among other things, difficulties with service discovery, having to power up and down containers every time they’re updated, and having lots of them for every little thing. A special container orchestrator was needed to do away with the problems. While several such tools exist, Kubernetes is currently the most widely used tool, according to Stack Overflow and CNCF surveys.

Kubernetes added several layers of abstraction to the deployment process. The developer could now take advantage of the YAML files Kubernetes has. These files can be used to setup containers in a way that they can communicate with each other by poking holes in the container walls, so to speak.

Another abstraction Kubernetes offers is deploying containers. Developers could now just tell Kubernetes how many replicas they need to run and which pods are they running. Scaling the application became much simpler for the software developer. These groups could then easily be transformed into services, that can be accessed through the internet.

### Infrastructure as code

Infrastructure as code- tools allow teams to automate server environment configurations. For example, Terraform is able to automate the installation of Docker, creation of users, and setting up the network settings. This environment can then be copied to production environment, and it should be the same working environment as you had in testing environment.

One of the benefits of Terraform is that automating configurations reduces human error. Where a small lapse in judgement can mean non-functional production, automated systems do not need to worry about it.

Another great feature in Terraform specifically is the declarative nature of the config file. Often managing systems involve changing the running system by removing, adding, or changing a certain element. In Terraform, you don’t need to remove components manually using a console, you just edit the end result you want in the configuration file.

Some disadvantages IaC solutions have is that it’s yet another thing to master. For example, Terraform has its own configuration file syntax. Some say there is potential for over-engineering by automating simple configuration tasks. I would argue that it’s better to have some kind of document where anyone who needs that information can get it. Too often there is one guy in the company who knows how to setup things, at least with Terraform, you have the YAML file for clues.

### Kubernetes on bare metal

Kubernetes is great at managing miniature virtual machines, containers, that much is clear, but should Kubernetes itself run on a VM? It depends on the needs of the software. Both methods have distinct advantages and disadvantages that can all be mitigated to some extent, but the differences are still clear.

Let’s have a look at advantages running of Kubernetes on VM over BM. Virtual machines offer better reliability and availability. In case a server goes down, typically more pods are affected, but importantly recovery is quicker. Additionally, security is increased in a VM environment, as the attack surface may only affect a single node. In some ways resource utilization can be optimized using VMs, however a lot of those resources go towards the hypervisor.

Running Kubernetes directly on top of the operating system brings its own advantages. Namely power, and lots of it. According to thenewstack.io, running Kubernetes on bare metal performs 2-4 times better in every metric, in addition to having network bandwidth that makes VM look crippled. Some cost savings can be achieved by not having VM providers to pay for.

As an immaterial benefit that will materialize very quickly, Kubernetes is much simpler to implement on VM. For a smaller application and a smaller team, using Kubernetes on VM might be the right choice. Additionally, if security and reliability are of high concern, VM provides some benefits. Say if you wanted to run your hospital systems on Kubernetes (for some reason), and money is not an issue when adding more iron to the border, maybe VM is the right choice.

Kubernetes on bare metal is the right choice for mid to large size businesses. It offers savings on server costs and the business can acquire the skills needed to support BM implementation. Additionally, you are safe from a very specific current development in the VM space.

### When your VM bricks

Late 2023 Broadcom managed to complete their purchase of VMWare, the market leader in virtualization. In doing so, they discontinued dozens of applications, perpetual license, and hiked prices by sometimes hundreds or even thousands of percents. Even if justified these price hikes are not something companies can afford in today’s economy.

It's important for companies to prepare for situations where one of their technology providers becomes unavailable. One of the ways to prepare is to avoid being married to any single technology. Or alternatively rely on open-source software that has traction and if worst comes, you can operate yourself. This is obviously easier said than done, it’s why we pay for the convenience.

Another important point is to have back-up plans. Some preplanned loose guidelines about what to do when a provider does disappear. This should be done for mission critical components, and it should always be up to date.

All that being said, there will be lots of companies in panic mode trying to come up with a good solution. And I’m sure there are VM companies ready to help them out. Let’s just hope it doesn't cost them too much, and the next time even less.