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.

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.