

Automated Deployment of Software Containers Using Dockers

Dr. Latika Kharb

Associate Professor (IT), JIMS, Sector-05, Rohini, New Delhi. India.

Abstract – Over time, due to increased product functionalities, software projects have become more and more complex and along with increasing work completion pressures, the software projects are required to be accomplished in lesser amount of time but with fewer people. In this paper, our focus is on exploring the docker technology: from defining the basic characteristic features, their advantages and limitations to the question behind our minds related to its performance analysis: how docker could help in building better software and lead to some good reproducible research as compared to virtual machines. IT organizations could shrink application delivery from years to months and can achieve up to fifteen times greater efficiency in their use of computing resources by using Docker. Docker containers bind a block of module that contains everything it needs to run: the source code, output and libraries - anything that one can be installed on a domain: this ensures that it will always run similarly without any environmental constraints.

Index Terms – Docker Technology, virtual machines, software projects, module, domain, constraints.

1. INTRODUCTION

Delivering a high quality reliable product is the main focus in any software development [1]. In San Francisco at DockerCon 2015 held on June 22, 2015; the Docker project was introduced for having powerful new capabilities that strengthen the portability of multi-container distributed applications i.e. Docker multi-host SDN. Docker multi-host SDN is now available as native platform functionality to ensure that multi-container distributed applications can seamlessly communicate over various IP networks, while being portable over any network infrastructure. This easy-to-use multi-host SDN functionality provides a new level of consistency in terms of how applications are networked through their full lifecycle: where a development team can initially define the topology of its distributed application, while the networking team can, at a later stage, apply the sophisticated networking policy necessary to run an application with the highest level of availability and security in production.

Docker has rapidly become the standard management tool and image format for containers[2]. Docker is an open platform for developers and system administrators to build, ship and run distributed applications. With Docker, IT organizations shrink application delivery from months to minutes, frictionlessly move workloads between data centers and the cloud and can achieve up to 20X greater efficiency in their use of computing

resources. Docker is now used by millions of developers across thousands of the world's most innovative organizations, including eBay, Groupon, ING etc. Within the last two years, Docker [3] has emerged as a standard runtime, image format, and build system for Linux containers. Docker allows you to package an application with all of its dependencies into a standardized unit for software development. Docker containers wrap up a piece of software in a complete file system that contains everything it needs to run and install on a server. This technology guarantees to always run the same, regardless of the environment it is running in.

2. ADVANTAGES OF DOCKER TECHNOLOGY

In this section, we've shortlisted some of the major advantages of using Docker.

- **Lightweight**

In Docker Technology, as containers run on a single machine and share the same operating system kernel; so they start up instantly and thus make efficient usage of RAM.

- **Open**

Docker containers are based on open standards that allow containers to run on all major Linux distributions and Microsoft operating systems with support for every infrastructure.

- **Secure**

Containers isolate applications from each other and the underlying infrastructure while providing an added layer of protection for the application.

- **Build and Collaborate on Applications**

Docker subscription includes Docker Trusted Registry/ Hub that helps to store/ manage and distribute the Docker image content within the team.

- **Run Your Application Anywhere**

Docker Engine installs on any physical or virtual server infrastructure running in our private data center/ cloud service provider.

- **Responsive, Secure and Stable**

Docker subscriptions include commercial support for both the registry service and Docker Engine to ensure a responsive, secure and stable environment.

- Flexible Deployment Options**

Docker Subscription is available for Cloud or Server to meet our deployment requirements and with 12 or 24 hour support to align to our application service levels.

3. DOCKER HELPS FOR BETTER SOFTWARE

When the application is in Docker containers, one does not have to worry about setting up and maintaining different environments/ tools for each language. Our focus is concentrated only on creating new features and/or fixing any existing issues and then shipping the software.

- Accelerate Developer On boarding**

With Docker, one can easily take copies of our live environment and run on any new endpoint running Docker.

- Empower Developer Creativity**

The isolation capabilities of Docker containers have freed developers from the worries of using “approved” language and tools. Now, the developers can use the best language and tools for their application service without worrying about any conflicts.

- Eliminate Environment Inconsistencies**

By packaging up the application with its configurations and dependencies together and shipping as a container, we don’t have to worry about having to install the same configurations into a different environment.

- Easily Share and Collaborate on Applications**

Docker develops a similar framework for developers and system administrators to work together on distributed applications.

- Distribute and share content**

We can store, distribute and manage Docker images in the Docker Hub with the team. Image updates, changes and history are automatically shared across your organization.

- Simply share your application with others**

Ship one or many containers to others or downstream service teams without worrying about different environment dependencies creating issues with our application.

- Ship 7X More**

Docker users on average ship software 7X more after deploying Docker in their environment.

- Easily Remediate Issues**

Docker makes it easy to identify issues and isolate the problem container, as it quickly rolls back to required changes.

4. CHALLENGES BEHIND USAGE OF DOCKER

The entire Docker holds the potential to address the shortcomings of certain existing approaches as well as reproducible research challenges that stem from complex computational environments [4]. Docker has proved its worth in concepts like versioning, modular design, portable containers, and simple interface etc. However, these advances have some challenges of their own like:

- Docker does not provide complete virtualization but relies on the Linux kernel provided by the host. [5].
- Docker is limited to 64 bit host machines, making it impossible to run on older hardware (in today’s scenario).
- Potential computer security issues may still need to be evaluated. Among other changes, future support for digitally signing Docker images may make it easier to build off of only trusted binaries.
- Most importantly, it remains to be seen if Docker will be significantly adopted by any scientific research or teaching community.

5. PERFORMANCE ANALYSIS MEASURE (SOFTWARE WITH/ WITHOUT USING DOCKER)

The effectiveness of Docker approach for supporting reproducible research depends on how each of these features are adopted and implemented by individual researchers. Here, we have summarized a few of these practices here on the basis of what we have analyzed from literature study of Docker:

- Code executing inside a container on a local machine can appear identical to code running natively, but with the added benefit that one can simply recreate or snapshot and share the entire computational environment with a few simple commands. This works best if researchers set up their computational environment in a container from the outset of the project.
- Docker can be used in a purely interactive manner to record and distribute changes to a computational environment.
- Docker file commands need not be limited to installing software, but can also include execution.
- The Docker Hub significantly reduces the barriers for making even large images readily available to other researchers.

Docker is acting as a disruptive force that is forcing people to rethink many of the layers of a cloud stack - starting from configuration management, networking, build, deployment and even micro services [6].

6. PERFORMANCE ANALYSIS W.R.T DOCKER: WHEN TO USE CONTAINERS OR VIRTUAL MACHINES (VM)

Docker is the new technology started out as a project of The Next Platform-as-a-service company dot Cloud in 2013. While Docker utilizes virtualization technology in the Linux kernel, it uses the kernel to create containers, not virtual machines. Here, we have summarized each of three on performance analytics:

• The Virtual Machine

In 1999, VMware released Workstation and the entire technology industry changed. The virtual machine (VM) has been at the center of the cloud computing explosion. The rapid adoption of VMs led to significant changes in processor architecture. Even more than that, the cloud-based platform providers of today could not exist without the virtual machine. Amazon, Digital Ocean, Linode and Joyent are just some of the cloud providers that depend on virtual machines.

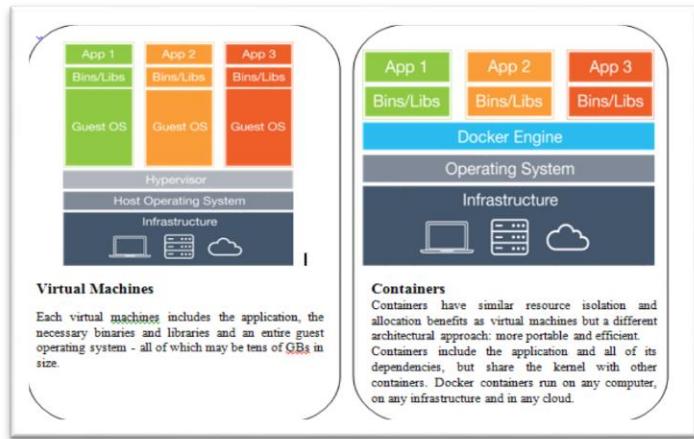


Figure 1: VM and Containers

• The Container

While a virtual machine abstracts away the hardware, container abstraction happens at the operating system level. Each type of container technology has an explicitly stated purpose that limits its scope. LXC, the initial technology Docker was built on, is scoped to specific instances of Linux. Containers are often used for running specific applications due to: less CPU and memory usage than a VM.

7. CONCLUSION

Using Docker, IT organizations could shrink application delivery from years to months and can achieve up to fifteen times greater efficiency in their use of computing resources. In a nutshell, here's what Docker can do for us: it can get more applications running on the same hardware than other technologies; it makes it easy for developers to quickly create, ready-to-run container applications; and it makes managing and deploying applications much easier. Combining all together one can see why Docker is high in demand.

REFERENCES

- [1] Proposing a Comprehensive Software Metrics for Process Efficiency: International Journal of Scientific & Engineering Research, Volume 5, Issue 9, September-2014.
- [2] An Updated Performance Comparison of Virtual Machines and Linux Containers, Wes Felter, Alexandre Ferreira, Ram Rajamony, Juan Rubio, IBM Research, Austin, TX.
- [3] What is Docker? Solomon Hykes and others <https://www.docker.com/whatisdocker/>
- [4] An introduction to Docker for reproducible research, Carl Boettiger, Center for Stock Assessment Research.
- [5] Our Troubles with Linux Kernel Upgrades and Why You Should Care, Harji, A.S. et al. 2013. ACM SIGOPS Operating Systems Review. 47, 2 (2013), 66–72.
- [6] <http://anandmanisankar.com/posts/container-docker-PaaS-microservices/>