Docker Machine and Swarm

SHREE GOVIND MISHRA[1]

[1] School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
[*] Corresponding authors:shremish@indiana.edu

project-000, March 27, 2017

Docker is a container-based technology which helps in the packaging and shipping of applications quickly and across networks.The usage of Docker is improving in the Agile Software workforce and among the DevOps because of its attributes of Continous Integration and testing. To create a cluster of Docker Engines into a single docker virtual engine is used the Docker-Swarm which is tested for more than 50,000 containers.

Keywords: Cloud, I524, Big Data, Virtualisation, Docker, Docker Swarm, Docker Machine , Virtual Machines, VMs, Containers,LInux Containers, lightweight containers,LXC

https://github.com/Govind273/sp17-i524/blob/master/paper1/S17-IR-2021/report.pdf

1. INTRODUCTION

Docker is an open-source container-based technology. It is an extension of Linux Containers (LXC): which is a unique kind of lightweight, application-centric virtualization that reduces overhead and makes it easier to deploy software on servers[1]. A container allows a developer to package up an application and all its part including its code, stack it runs on, dependencies it is associated with, system tools and everything the application requires running within an isolated environment. This makes it easier for programmers and developers to run more apps on the same server and also to package and ship the apps very frequently. Docker has been able to popularize the container approach in part because it has improved the security and simplicity of container environments. Plus, interoperability is enhanced by its association with major companies – such as Google, Canonical, and Red Hat – on its open source element libcontainer.

2. HOW DOCKER DIFFERS FROM VMS

The virtualization of application can be obtained with Hypervisors or Virtual Machine Manager(VMM) which makes it easy for applications to run in isolation with one another while sharing the same underlying hardware architecture. VM hypervisors such as Hyper-V, KVM, and Xen are all based on emulating virtual hardware which means they are expensive in terms of system requirements as each VM requires the underlying VMM, Memory, and RAM. Instead of virtualizing hardware, containers rest on top of a single Linux instances. The only requirement is to keep the virtual machine image small in containers. Thus, the smaller the image is the less is needed to be stored and the lesser information is needed to be sent around the network, which makes them lightweight in comparison to the Virtual Machines. Thus a large number of containers can be hosted on the same host OS in comparison to the Virtual Machines. A full virtualized system usually takes a time to start whereas the Docker container does take even less than seconds to upstart and running with a lower overhead than the VMs[2]. Containers are potentially much more efficient than VMs because they are able to share a single kernel and share application libraries.

3. USES OF DOCKER

3.1. Docker for DevOps

Another major use of Docker is its use in the DevOps community."DevOps is the practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support[3]". Docker is not there to replace other configuration management tools and instead can be incorporated with other configuration management tools like Chef, Puppet, Salt or ansible. The other major benefit of using Docker, Dockerfiles, the registry and the whole Docker ecosystem is that the development teams don't have to learn domain specific language, tools or technologies which are suitable to a specific platform. Docker will not be domain specific as it can be integrated with other configuration tools and software as many times Docker can be made to work with the other configuration management tools. Docker can also the eliminate the need for a development team to have the same versions of everything installed on their local machine[4]. While working on to incorporate Docker to their development workflow, Spotify created Helios, an application that manages Docker deployments across multiple servers and that alerts the development teams when the server isn't running the correct version of a container.

3.2. Docker as Virtualized Sandbox

Docker allows systems administrators and developers to build applications that can run on any Linux distribution or hardware in a virtualized sandbox without making custom builds for all the different environments[4]. It is easy to deploy Docker containers in a cloud scenario. Which can easily integrate it with typical DevOps environments seamlessly (Ansible, Puppet, etc.) or use it as a standalone.

3.3. Docker for continuous integration

Docker can be used as git for continuous integration. Docker is similar as changes in the system as changes can be tracked just like changes in the git. These collaboration features (docker push and docker pull) are one of the features which are important.The Docker pull/push are features which help the DevOps teams to easily collaborate and in quickly building infrastructures together.The application teams can thus share app containers with operation teams and they can share MySQL, PosgreSQL, and Redis servers with application teams[1].

4. DOCKER MACHINE

Docker Machine is a tool that lets you install Docker Engine on virtual hosts, and manage the hosts with docker-machine commands[5]. You can use Machine to create Docker hosts on your local Mac or Windows Machine, on your company network, in your data center, or on cloud providers like AWS or Digital Ocean. Docker Machine is the only way to run Docker Engine on Mac or Windows previous to Dockerv1.12 and starting with the Dockerv1.12 we have Docker for Mac and Docker for Windows. Thus we can create a cluster of Docker hosts which is called a Swarm using Docker Machine.

5. DOCKER SWARM

Docker Swarm provides native clustering capabilities to turn a group of Docker engines into a single, virtual Docker Engine[6]. These helps to scale up the applications as if these are all running on a single, huge machine. It does so by providing a standard Docker API where if any tool which communicates with the Docker, the daemon can use Docker Swarm to transparently scale to multiple hosts: Dokku, Docker Compose, Krane, Flynn, Deis, DockerUI, Shipyard, Drone, Jenkins and, of course, the Docker client itself. Docker Networking, Volume, and plugins can also be used through their respective Docker commands via Swarm. Swarm has been tested and is production ready to scale up to one thousand (1,000) nodes and fifty thousand (50,000) containers with no performance degradation in spinning up incremental containers onto the node cluster Swarm also comes with a built-in scheduler, but you can easily plugin the Mesos or Kubernetes backend while still using the Docker client for a consistent developer experience. To find nodes in the cluster, Docker Swarm can use either a hosted discovery service, static file, etcd, consul and zookeeper depending on what is best suited for the environment[5]

6. ECOSYSTEM SUPPORT TO DOCKER

Finally, it is easy to deploy Docker containers in a cloud. [4]. It can be integrated in a typical DevOps environments seamlessly (Ansible, Puppet, etc.) or use it as a standalone. You can do local development within a system that is identical to a live server, deploy various development environments from your host that each uses their own software, OS, and settings. easily run tests on various servers and create an identical set of configurations, so that collaborative work isn't ever hindered by parameters of the local host.Ecosystem support for Docker is improving with time.

- Operating Systems support to Docker: Is compatible with virtually any distribution with a 2.6.32 + kernel RedHat Docker collaboration to work with across fedora and other(2.6.32)+.
- It is compatible with Private PaaS(Platform as a Service) technologies: Openshift, Solum(Rackspace and Openstack).
- Public PaaS technologies like Voxoz, Cocaine(Yandex), Baidu, etc.
- Ecosystem Support for IaaS is present via Rackspace, Digital Ocean, AWS(Amazon Web Services), AMI, etc.
- Orchestration tools Support and Integration with: Chef, Puppet, Jerkins, Travis, Ansible, etc.
- In Openstack Docker integration into NOVA(compatibility with Glance, Horizon)are also present.

7. SHORTFALLS OF DOCKER

Though dockers are legacy virtualization techniques, they are not the goto solution for all kinds of virtualization and cannot be considered as a replacement of the VMs.

- VMs are self-constrained with as they have a unique operating system(OS), drivers and application components whereas the containers are dependent as containers under Docker cannot run on Windows server.
- VMs provide high level of isolation as the system's underlying hardware resources are all virtualized and thus any bugs or viruses could not affect the other VM in the virtualized environment.
- The kinds of tools and technologies required to manage the containers are still lacking in the industry and thus only a few management tools from companies like Google and Docker such as Kubernetes and Swarm respectively are avalible which are not appropriate for an open-source product.

8. FUTURE WITH DOCKER

Docker Inc has set a clear path to the development of core capabilities (libcontainer), cross service management (libswarm) and messaging between containers (libchan). Docker's key open source project is libcontainer. Libcontainers enables the containers to work with the Linux namespace, control groups, capabilities, AppArmor security profiles, network interfaces and firewalling rules in a consistent and predictable way. Libcontainers is receiving help from Google, RedHat, and Parallels to build the program as they will work with docker as core maintainers of the code[4]. Libconatiner which is written natively in Google's Go is also being ported to other languages. Parallels' libct, which includes libcontainer's functionality, has native C/C++ and Python bindings. Microsoft is bringing Docker to their Azure platform, a development which would integrate the Microsoft products with Linux applications.

9. CONCLUSION

According to a survey by Docker community it is observed that there are more than 1200 Docker Contributors, 10,000 Dockerized Applications at index.docker.io, 300 Million Downloads, and 32,0000 Docker-related Projects[7]. At Docker Ecosystem there is a potential for some advanced deployment tools that combine containers, configuration management, continuous integration, continuous delivery, and service orchestration in the coming days.

REFERENCES

[1] CenturyLink, "What is docker and when to use it," Web Page, Apr. 2014, accessed 2017-2-20. [Online]. Available: https://www.ctl.io/developers/blog/post/what-is-docker-and-when-to-use-it/

[2] Ken Cochrane, "How is docker different from a normal virtual machine?" Web page, apr 2013, accessed 2017-02-23. [Online]. Available: http://stackoverflow.com/questions/16047306/how-is-docker-different-from-a-normal-virtual-machine

[3] the agile admin, "What is devops," Web Page, accessed 2017-3-13. [Online]. Available: https://theagileadmin.com/what-is-devops/

[4] opensource.com, "Who's using docker?" Web Page, Jul. 2014, accessed 2017-2-19. [Online]. Available: https://opensource.com/business/14/7/docker-through-hype

[5] Docker Inc, "What is docker machine," Web Page, accessed 2017-2-23. [Online]. Available: https://docs.docker.com/machine/overview/

[6] ——, "Docker swarm," Web Page, accessed 2017-2-21. [Online]. Available: https://www.docker.com/products/docker-swarm

[7] DATADOG, "8 surprising facts about real docker adoption," Web Page, accessed 2017-2-23. [Online]. Available: https://www.datadoghq/com/docker-adoption/