

Moby Project and Linux Kit

Aditi Singh, Adhishi Pokharna, Aastha Sharma

College of Engineering and Technology, Mody University of Science and Technology, Lakshmangarh (Rajasthan), 332311, India

adsingh2401@gmail.com, adhishi.pokharna@gmail.com, aasthacse@gmail.com

+91 8963012401, +91 9530488101, +91 7597741745

ABSTRACT: LinuxKit packages lean, secure and portable Linux subsystems that can serve Linux container functionality as an element of a container platform in accordance with Kernel Self Protection Project (KSPP) principles. Since all system services are containers, hence they can be replaced and removed. It is built from containers and for containers and, the basic result is a 35MB busybox (read-only root file-system) initrd for containers.

Moby Project is an open framework that empowers anyone to put together their own container system in an extremely powerful manner that leaves no scope for redundant or wasted efforts. In addition to the 80+ core libraries and framework, the Moby Project offers ready-to-go reference assembly that anyone can use as per the requirements and it also gives an option to "bring your own components" (BYOC) packaged as containers with the choice to mix and match among all of the components to build a specialised container system.

KEYWORDS: *Portable, Backend, KSPP, containerd, Kubernetes, Azure*

I. INTRODUCTION

At DockerCon in Austin, Texas, Docker Founder and CTO Solomon Hykes said, "(Docker) is a bunch of projects, not a monolith." Two of the newest of these projects are open source projects Moby Project and LinuxKit that was developed by a collaboration between the Linux Foundation and leading companies such as silicon partner ARM, infrastructure providers like HPE and cloud companies including Microsoft and IBM.

Docker project was evolving into an enormous monolithic ecosystem around various user requirements, and use cases where all of the features presented may not be needed, therefore, the best solution was to split everything into tinier pieces (components, now available: infrakit, linuxkit, swarmkit, datakit, vpnkit, buildkit, containerd, etc) and later assemble the components as per the run requirements.

WHY THE NEED FOR LINUXKIT AND MOBY PROJECT?

Traditionally, containers were consequent to an operating system – once the system boots, the container engine loads and then, the container hosted applications like databases and web servers load. To build custom operating systems, Moby intends to containerize things like DHCP servers, DNS servers etc that enables the pulling and plugging in a similar way manner as that of conventional containers.

Users fancied a Docker-native experience with Linux container support on their platform which varied from cloud platforms such as AWS, Google Cloud, to server platforms such as Windows Server and IoT platforms but in many cases

such as Mac OS and Windows, the platforms themselves did not ship with Linux included. Hence, LinuxKit is a solution that assembles custom Linux subsystems with the goal to create a more native Linux experience for Windows and Mac OS and, cloud platforms.

It is imperative to believe in every one of the components so that we can be assured that our LinuxKit subsystem is assembled securely. More appropriately, we should be able to discern where the components come from, what they are supposed to include, and when we should cease using them (in case they become out of date).

II. ADVANTAGES

Having a complete operating system with its own kernel may seem a source of additional security because it is difficult to escape from virtual machines. Since the Linux distribution built is immutable, all the kernel binaries are read-only and cannot be affected.

Secure, Lean and Portable

The base LinuxKit Linux distribution is tiny, at its smallest, LinuxKit Linux takes up merely 35MB with a fast boot time to match because of its container native approach. Attack surfaces are much less in container-specific OS than with a general- purpose OS, so there are seldom opportunities to attack and jeopardize a container-specific OS. It's considerably portable and can thus, work on desktops, IoT, mainframes, bare metal, servers and also on virtualized systems as containers are reachable into all areas of computing.

- Immutable infrastructure employed to building

custom Linux distributions

- Though completely stateless, persistent storage can be attached as per the need
- Although designed for building and running clustered applications but clearly not limited to container orchestration such Kubernetes

III. MOBY PROJECT

Moby Project provides a CLI tool, called Moby which is used for construction of a fully customized, container-centric Linux distribution with Docker components.

Moby consists of the following elements:

1. A library comprising of vital containerized backend components like orchestration, infrastructure

management, networking, storage, image distribution, low-level builder, volume management, logging facility(SwarmKit, containerd, runC, InfraKit, SwarmKit, HyperKit, VPNKit)

2. A framework for putting together the required components into a standalone container platform, and an efficient tooling to build, test and deploy artifacts for diverse platforms like executables for Mac and Windows; VM images for cloud and virtualization providers.
3. Moby Origin, which is a reference assembly or an open base for the Docker container platform, which can be used as it is, altered, or used as an inspiration to create your own.

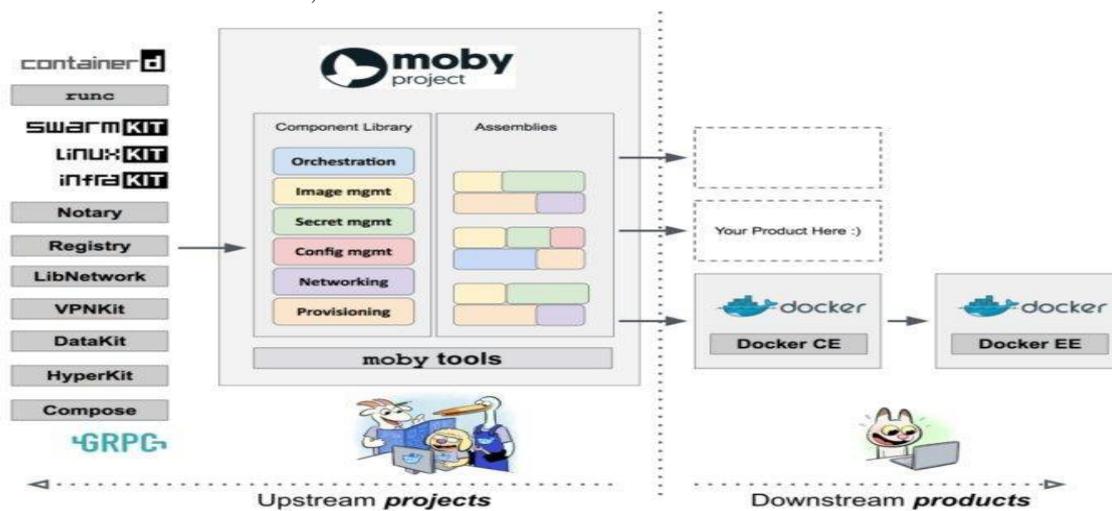


Fig 1

To minimize duplication of efforts, different models of cars share the same essential components, like engines, wheels, and chassis, but more precisely, they share or collude in the assembly process. This process is analogous to how we use a

web framework like Ruby on Rails to create several different types of web applications. The essence (Rails) remains the same, but the end product (i.e., the web app) is entirely diverse and one of its kind.

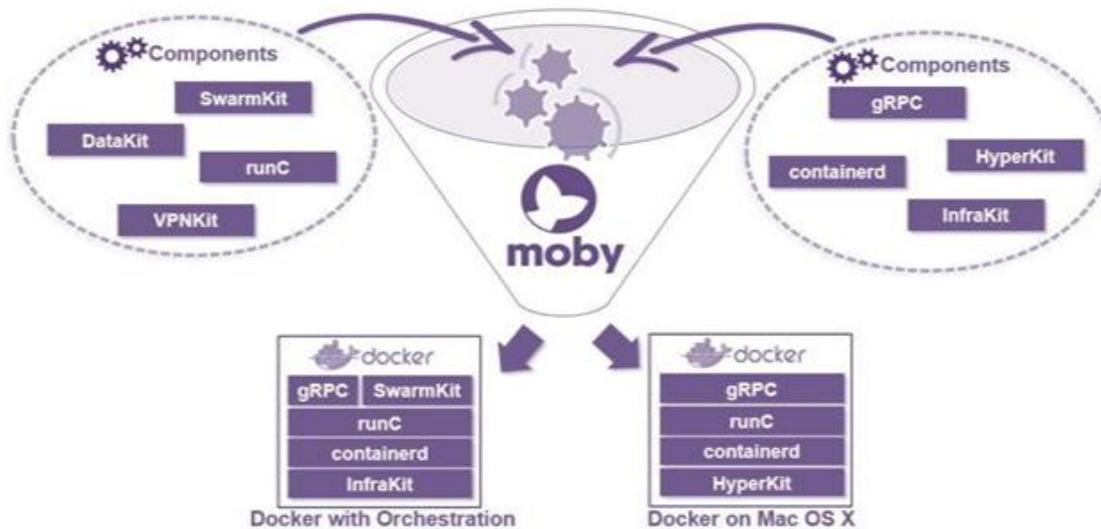


Fig 2

IV. LINUXKIT

LinuxKit is a builder for default highly secure Linux operating system such that everything is based on containers. All of the processes, including system daemons, run in containers, allowing users to create a Linux subsystem with only the necessary and required services. It renders the Linux components otherwise missing for a container platform on non -

Linux systems like Mac and Windows computers. To facilitate an immutable infrastructure, it renders a read-only root file-system to deployments enabled by InfraKit. If Moby is used to create an ISO image, LinuxKit takes a charge of pushing it and running on diversified platforms. By Hyper-V isolation techniques, one can run both Windows and Linux containers parallelly and build Linux/Windows hybrid clusters.

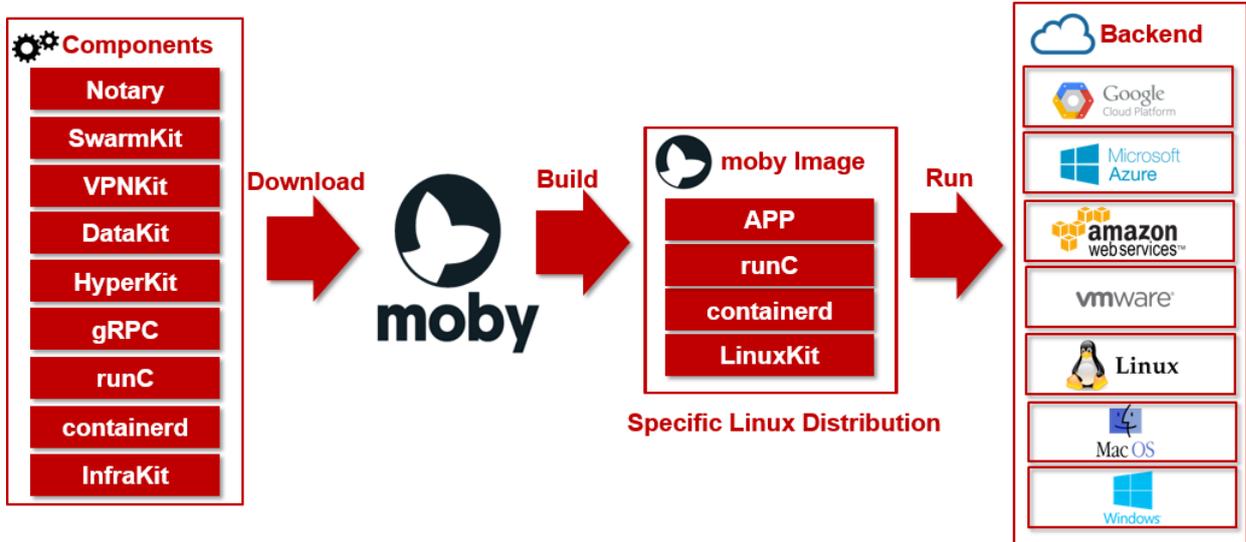


Fig 3

V. FEATURES

BUILD	RUN
<ul style="list-style-type: none"> On the basis of build configuration in YAML file which specifies Docker images (kernel, boot, init or services), build pipeline, downloading, unpacking, archiving, compressing are performed. On the criteria of the backend on which Moby image is supposed to run, Build generates an image with specified image format. 	<ul style="list-style-type: none"> Supported Moby image is run on a backend (gcp image format for Google Cloud Platform, vhd for Azure, raw for AWS etc). Since LinuxKit provides Linux minimal functionality bundled in a Moby image, Linux native container can run on any hypervisor which is, in turn, running on a specific platform.

VI. USE CASE

The Moby Project is to Docker what Fedora is to Red Hat Enterprise Linux and, Electron is to VSCode. Moby will convert Docker from a monolithic engine into a toolkit to assemble its components into distinct configurations and encourages reuse of each of these elements while accelerating the software containerization movement. Docker for Windows servers has been possible by LinuxKit, bringing Linux Subsystem on a different OS/Platform. Moby is used

to build the distro images and LinuxKit tool to run them.

Docker is building an ecosystem of technologies around containerization with Docker as the central unit, i.e., a LEGO set of containerized components and a framework for putting together these elements into a standalone, specialised container platform, more appropriately, a community centre for all container enthusiasts to explore and interact. Conclusively, Docker wants to use Moby and LinuxKit in order to fuel the Docker CE and Docker EE environments.

INTENDED FOR	NOT INTENDED FOR
<ul style="list-style-type: none"> • Hackers who want to customize or patch their Docker build • System engineers or integrators building a container system • Infrastructure providers looking to adapt existing container systems to their environment • Container enthusiasts who want to experiment with the latest container tech 	<ul style="list-style-type: none"> • Application developers hoping for quick and easy means to run the applications in containers, Docker CE is a more viable option. • Enterprise IT and development teams hoping for a ready-to-use, commercially supported container platform. Docker EE is recommended instead unless a highly secure, specialised and customised container solution is designed and orchestrated for increased security and performance. • Anyone inquisitive about containers and looking for an uncomplicated way to learn.

VII. COMPETITORS

Docker is a product of Moby. No choice for a Moby framework as a custom, bootable container system assembler is available yet. Alternatives are only available for Docker as a product: RKT, offering containerizing platform, but varies in security principles and cannot be a replacement and as reliable for production as Docker is.

VIII. FUTURE SCOPE

The projects are modelled around an idea for cross-ecosystem collaboration on architecture, design and experimentation by leveraging interchangeable, well-tested containerized components to build new sub-systems. Moby allows anything that can be containerized to become a Moby component, which will, in turn, create continuing possibilities for collaboration with additional projects outside of Docker, where contributors can leverage well-tested similar components to create highly specialized container systems.

IX. CONCLUSION

Docker as a product and a brand is not going away, it will still remain Docker. End users will not see any difference in the product, other than possibly seeing sprinkles of the Moby Project being stated in the documentation or the Moby Project being cited in technical write-ups as the implementation specifications of Docker itself.

Moby Project and LinuxKit are more of a developer kit than an end-user product like Rancher, Atomic, CoreOS, more specifically meant for system builders. to enterprise IT and development teams looking for a ready-to-use, commercially supported container platform. Though, there is a lot of interest in packaging up system level services inside of containers. Docker as a business may be biased towards Docker Swarm for cluster management but Moby

embraces swappable alternatives, hence kubernetes can easily be assembled in a release.

We use the latest Moby to build our own distro using a YAML file and can choose to use LinuxKit as the base. Then we can install additional items on top of it and it outputs various images types: IMG, ISO, Google cloud images, AMI, etc.

REFERENCES

- [1]. Stewart, Blaize.(2017, Jun 19).Quick Start for Moby and LinuxKit.[WWW document].URL <https://www.wintellect.com/quick-start-moby-linuxkit/>
- [2]. Calcote, Leen.(2017,May 4).What is Docker’s Moby Project?..[WWW document].URL <https://www.infoworld.com/article/3193904/containers/wh-at-is-dockers-moby-project.html>
- [3]. Bhartiya, Swapnil(2017, Apr 20).Why Docker created the Moby Project?.[WWW document].URL <https://www.cio.com/article/3191344/linux/why-docker-created-the-moby-project.html>
- [4]. Cormac, Justin.(2017, Apr 18).Announcing linuxkit: a toolkit for building secure, lean and portable linux subsystems.[WWW document].URL <https://blog.docker.com/2017/04/introducing-linuxkit-container-os-toolkit/>
- [5]. Tachibana Kazuhiro.(2017, Jun 26).New capability of Docker - Moby & LinuxKit.[WWW document].URL <https://techblog.rakuten.co.jp/2017/07/26/moby-and-linuxkit/#backend>
- [6]. Raina, Ajeet.(2017, May 6).Demystifying the Relationship between Moby & Docker.[WWW document].URL <http://collabnix.com/demystifying-the-relationship-between-moby-docker/>
- [7]. White, Jesse.(2017, Apr 24).A Docker aptain Goes to DockerCon 2017,Part 1: The Moby Project and LinuxKit.[WWW document].URL <https://www.contino.io/insights/a-docker-captain-goes-to-dockercon-2017-part-1-the-moby-project-and-linuxkit>