

Rocket（也叫 rkt）是 CoreOS 推出的一款容器引擎，和 Docker 类似，帮助开发者打包应用和依赖包到可移植容器中，简化搭环境等部署工作。Rocket 和 Docker 不同的地方在于，Rocket 没有 Docker 那些为企业用户提供的“友好功能”，比如云服务加速工具、集群系统等。反过来说，Rocket 想做的，是一个更纯粹的业界标准。

CoreOS 把它的容器称为 App Containers，里面包含 app container image、runtime、container-discovery 协议等。其中，App Container Image 和 Docker 里的 Image 比较类似，包含应用必需的元素组成，如源代码和二进制文件。Rocket runtime 则是依照 App Container 标准规格打造的，旨在将容器真正的变成一款命令行工具。

Rocket 0.8 利用了 Intel 的 Clear Containers 项目，它在英特尔芯片中使用 VT-x 指令集为容器增加硬件隔离。事实上，Intel 利用 Rocket 为其建立一个概念验证项目，目前的工作只不过是 CoreOS 和 Intel 之间的合作。一个容器在 Rocket 0.8 执行其整个进程按层级封装在一个 KVM 中，这意味着容器的 contents 被防火墙从主机隔离。

这么多隔离听起来好像有些过分，但是这是对容器安全的持续关注。大多数容器环境中声称提供容器技术、cgroups 和 namespaces 隔离。在多租户环境中，隔离度是至关重要的。

最大的问题是，Rocket’ 新特性是否被 OCI 采用。根据 CoreOS CTO Brandon Philips 表示，由 CoreOS 提议的最初的 appc 容器规范涵盖了容器管理的四个不同元素：packaging、signing、naming（sharing the container with others）以及 runtime。。

“当前 OCI 的焦点主要集中在 runtime。虽然我们也在努力让 appc 与 OCI 协调，但他同时表示也希望 OCI 规范应该为用户提供一个完整的 container image story。”

CoreOS 想做个带头人，但 Docker 也提供 Philips 概述的一些功能。比如 Docker 最近公布的 Docker Content Trust，一个为 Docker 容器设计的签名和验证机制。

通过使用 Content Trust 作为一种选择性机制来验证内容添加到官方 Docker Registry，并提供它作为一个开源的标准，Docker 希望以身作则，鼓励采用。

Rocket（rkt）发布 1.0 版本，这意味着从现在开始该引擎可用于生产环境。其命令行工具和磁盘格式等都认为是稳定的，可以再次开发。从该版本开始，任何接口方面的改动都会考虑向后的兼容性。