# 《代码英雄》第一季（5）：容器竞赛

代码英雄讲述了开发人员、程序员、黑客、极客和开源反叛者如何彻底改变技术前景的真实史诗。

什么是《代码英雄》

Command Line Heroes

代码英雄是世界领先的企业开源软件解决方案供应商红帽（Red Hat）精心制作的原创音频播客，讲述开发人员、程序员、黑客、极客和开源反叛者如何彻底改变技术前景的真实史诗。该音频博客邀请到了谷歌、NASA 等重量级企业的众多技术大牛共同讲述开源、操作系统、容器、DevOps、混合云等发展过程中的动人故事。

本文是《代码英雄》系列播客第一季（5）：容器竞赛的音频脚本。

容器的兴起为开发者们打开了一道新的大门，它简化了在机器与机器之间传递项目的成本。随着它变得广受欢迎，一场大战也悄悄拉开帷幕。这场战斗的奖品是容器编排的控制权，参赛者包括这个行业最快最强的玩家。

容器是开源运动中最重要的一项突破之一。在这一集里，特邀嘉宾 Kelsey Hightower、Laura Frank 和 Clayton Coleman 将告诉我们容器如何为未来添砖加瓦，以及编排技术为何如此重要。

**Saron Yitbarek：**

你有看过赛马吗？赛马们排成一行，蹄子刨着脚下的土壤。你可以想象出这么一副画面。比赛即将开始，在这些竞争者中脱颖而出的将是优胜者。

**00:00:30：**

不同的是，比赛的不是马。而是科技世界的诸侯。那么是什么让比赛如此重要？是怎样的珍贵的奖励，才会让这些参赛者们排着队，迫不及待地想要得到它？这是一场赢家将掌握容器编排技术规则的竞赛，而且胜利者只有一个。重要的是，不同于其他的比赛，赢得这场比赛，你不仅仅会成为今天的冠军，更有可能在来持续领先。

**00:01:30**:

我是 Saron Yitbarek，这里是代码英雄，一款红帽公司原创的博客。

第五集，容器竞赛。上一集我们见证了 DevOps 的崛起，以及一组新工具如何影响了其他人对开发者这一概念的看法。在这一集栏目中，我们会追溯容器技术崛起的历史，讲述容器技术如何通过拥有支持全新工作的可能性，来进一步扩展开发者这一角色的概念。然后我们会一起见证容器标准化是如何为容器编排奠定比赛基础的。

**00:01:30**：

这是一场严肃的比赛，也是一场全球性的比赛，吸引了行业里最快，最强大的选手。他们都准备好了为冲刺终点线而奋力一搏。准备好了吗？比赛开始了！

现在，随着这些“赛马”离开起点，也许我们应该弄清楚为什么这场比赛如此重要。谁会关心容器呢？好吧，算我一个。但是实际上，一开始我也并不知道容器是什么。以下我将讲述一个小故事 —— 我是如何醒悟容器之美的。

**00:02:00**：

不久之前，我还在为我网站写代码，然后有天我让我的朋友 Nadia 过来实现一些新的功能。我在保持代码干爽和可读性方面做得很好，当然，代码也经过了很好的测试。所以再加入一个新的网站开发者也不是一件难事。对吗？如果你也这样以为，那就错了。这是一个非常繁琐的过程，特别是当我们跑规范化测试时，这个问题尤为明显。

**00:02:30**：

代码运行正常，但我们不能在两台电脑上同时通过所有测试。我们有很奇怪的电脑时区设置问题，而且她的 Ruby on Rails 版本跟我的不同。就是一个很经典的问题：“我的电脑上可以代码运行”，“可是在我的电脑上就是不行”。我只好对代码做一些修改，直到它在我这里正常运行，但当我把它发送给 Nadia 时，程序又会崩溃。

**00:03:00：**

我很清楚，我和 Nadia 所碰到的这些问题，对于所有的开发者来说都或多或少经历过，甚至他们把这种经历当作玩笑来讲。有时候，我只能把这个当做是在我工作时必须要忍受的一部分。我没有意识到的是，这个问题有个最终解决办法。想象有一种方式可以降低人与人之间的隔阂；想象有一种方法可以让我们在开发中使用任意喜欢的工具，并且在传递工作成果时毫无阻碍；想象一下有一种办法，无论有多少人同时进行一个项目的开发，不管这些人散布在世界何地，都可以让项目从开发到测试，再到生产环境，保持连贯性。如果在我浪费好几周，用最笨的方式传递工作成果前就想到了容器该多好。

**00:03:30 - Liz Rice：**

一个容器实际上就是一个进程。

**Saron Yitbarek：**

Liz Rice 是 Aqua Security 的一名技术布道师。她描述了为何容器会如此实用。事实上容器把一切打包到了一个整洁、并且可以迁移的包中。

**00:04:00 - Liz Rice：**

这就像任何其他的进程一样，不同的是容器的世界非常小。比如，如果你启动一个容器，进程会被授予它自己的根目录。然后它认为自己在查看的是整台计算机的根目录，但实际上它只是在查看这个文件系统很小的一个子集。

**00:04:30 - Saron Yitbarek：**

通过打包一个可执行文件及其所有的依赖，容器可以在任何笔记本或者云中的虚拟机上运行。带着它自己的执行文件、库和依赖。所有的一切都包含在了容器中。所以，这就是容器神奇之处，容器在每个环境中的运行都会完全一样。这也就意味着开发者可以轻松地分享并协作应用开发，而不用担心计算机之间相互不兼容这个老问题。

**00:05:00：**

举一个类比的例子希望能够帮助你理解。你有听说过蓝围裙(Blue Apron)这个服务吗？该服务提供你做饭所需的一切，包括精心按照菜谱卡片搭配好的，所有做饭需要的原料。好的，想象一下如果蓝围裙所能带给你的不仅仅只是还没有处理过的食材，而是一整个厨房，有煤气灶，还有你所需要的全部餐具，一切你需要的都会装到小盒子里，放在门阶上。这就是一个容器。在我提到的那种情况下，容器技术就可以很好地解决 Nadia 加入进来时所碰到的问题，简单到像使用蓝围裙服务做一顿晚餐一样。虚拟机同样也可以提供一个预装好的环境。但要解释这个，我们就不得不抛弃蓝围裙这个比喻，让我们来看一看具体的细节。

**00:05:30 - Liz Rice：**

许多人都认为容器是某种轻量级的虚拟化技术、轻量级的虚拟机，事实上并不是。容器与虚拟机有很大不同。虚拟机有独属于自己的一整个操作系统，相比起来容器是共享操作系统的。一个计算机上的所有容器共享同一个操作系统的。

**00:06:00 - Saron Yitbarek：**

最后一点，容器和虚拟机可以并肩工作。容器不能替代虚拟机。虚拟化技术仍然可以提高过时系统的效率，并且对于服务器整合非常关键。但容器技术的兴起也为我们打开了新的大门。不妨这样想，如果我们全部依靠虚拟机的话，运行所有仿真服务器将产生大量的额外负担。

**00:06:30：**

一台虚拟机的大小至少是以 G 为单位的，然而一个容器可能也就只有 20 M 左右。一台虚拟机可能会需要若干分钟来启动，如果我尝试用它

部署一个网页应用的话，这可不是一个好消息。很长时间以来，人们都期盼一个轻量级的、更快速的完整机器虚拟化替代方案出现。

**00:00:07：**

回顾一下历史，1979 年就出现了容器的原型。Unix V7 的开发者们设计了一种根系统调用，使环境中只包括特定的程序。该突破为我们现在看到的容器技术指明了道路。另一个巨大的进展来源于 2008 年的 Linux 容器技术。现在，我们有了操作系统级的虚拟化技术。

**00:07:30：**

我们终于可以在一个单独的 Linux 内核上运行多个容器，而无需使用完整的虚拟机。这也就意味着程序对于基础架构的需求逐渐减少，但不是每一个人都能立马看到容器技术的潜力。

**Laura Frank：**

容器化真的是前所未有的、崭新的一个天才般的想法。

**Saron Yitbarek：**

Laura Frank 是 CloudBees 的技术总监。

**00:08:00 - Laura Frank：**

只有少部分人了解容器技术的来龙去脉，并可以运用它。不过相信随着时间的推移越来越多的人会接触到容器化的概念，随着越来越多的人开始使用这项技术，并且这些知识通过工程团队和工程组织，通过社区进行传播，就会变得更容易获得。

**Saron Yitbarek：**

因为和我们之前提到的与虚拟机的相似性，Laura 认为，因为我们之前提到的容器技术与虚拟机的相似性，容器的潜力被低估了。

**00:08:30 - Laura Frank：**

我在回想我的职业生涯，那是我还只是个普通的日常技术人员。如果你不是一个系统管理员或者 Linux 资深用户的话，容器还是一个你刚刚了解到的全新概念。我把它理解为使用一台虚拟机模式类似的东西，我可以去建立一个可以用完即弃的环境，而且这个环境完全独立，清理之后不留痕迹。

**Saron Yitbarek：**

容器除了能保持系统整洁之外，其实还大有可为。容器将会革新整个行业，并且随着开源项目和社区的兴起，在不久之后，容器标准化的充分实施将变为可能。

**00:09:00 - Scott McCarty：**

整个界面已经变得非常简单。

**Saron Yitbarek：**

Scott McCarty 是红帽的一名资深的容器策略顾问。他称得上是这个行业的资深人士，他在容器出现前，甚至是虚拟机出现前，就在做这方面的工作了。

**00:09:30 - Scott McCarty：**

在互联网 1.0 时代，我在一家线上零售商工作，我们有上千台实体机，我们用不同的方式，在所有这些不同的服务器上一遍又一遍地安装相同的软件。我们尝试了所有的方法。当你从原始的操作系统迁移到虚拟机，然后再到 Linux 容器、Solaris 容器，同样的问题一再出现，你仍然不得不在不同的虚拟机，或者类似操作系统实例的结构体之间管理配置。

**Saron Yitbarek：**

一旦容器变的规范化，一切都将改变。

**00:10:00 - Scott McCarty：**

比如，有了很多非常标准化的方式可以去处理现在这些打包好的应用，我认为容器技术的出现，从根本上改变了一切。它使得那些应用

非常容易使用，而且容器还不会对系统本身造成损害，同时相比虚拟机更加小巧快捷。

**00:10:30 - Saron Yitbarek：**

借助 Linux 容器带来的进步，这些新的开源项目和社区使得开发者们可以更好地携手合作。很多我们对于后端的焦虑都被一扫而光。突然间，容器和由它促进的微服务变得十分有吸引力。一旦一种共同的容器语言出现了，障碍就消失了，与此同时容器技术改变了我们的工作方式，也改变了我们学习新技术的步伐。

**00:11:00：**

还记得之前我和同事 Nadia 遇到的反复出现的问题吗？“在我这代码能跑”的场景？在容器的世界，这个问题将不复存在。相比于我们之前使用的标准的操作系统，开发者社区见证了容器是如何变得更加快速，成本低廉，并且容易使用的 —— 比传统操作系统更加容易。容器技术被采纳的速度十分惊人。但是要记得：容器标准的出现仅仅是容器编排这场竞赛的热身。

赛马们已经整齐排列好，随着信号枪一声令下，它们为了这场比赛的冠军而拼尽全力。竞争的不是容器本身，而是我们部署和管理容器所使用的工具。

**00:11:30：**

我是 Saron Yitbarek，这里是代码英雄。在这场标准容器编排竞赛中，哪位会胜出成为管理所有容器的平台呢？起初有两位竞争者处于领先地位。

**00:12:00：**

由 Apache 驾驭的 Swarm，和 Docker 驾驭的 Mesos。但是等等，怎么？现在出现了一匹黑马改变了这个格局，那就是谷歌。Linux 设立了云原生计算基金会（CNCF），随后 CNCF 推动了谷歌开源的编排引擎 Kubernetes。

**00:12:30：**

现在，相比 Kubernetes，Mesos 和 Swarm 已经抢占了先机，对吗？它们得到了 Apache 和 Docker 的支持，已经入场了一段时间了。但是 Kubernetes 有其他的“赛马”所不具备的优势。Clayton Coleman 会告诉我们这个秘密是什么。Clayton 是红帽负责 Kubernetes 和 OpenShift 的一名架构师。

**00:13:00 - Clayton Coleman：**

在 Kubernetes 诞生之初，谷歌就在项目的开放上做的很好，它降低了项目的贡献和参与的难度。谷歌极其关注让开发者和运维人员能更加容易地开展工作。有这样一个强烈的关注点，就是要做一个能让大多数开发者和运维的生活更轻松的东西。我觉得 Kubernetes 和围绕着Kubernetes 的社区找到了一个足够好的方式，让大部分人参与进来，他们让 Kubernetes 具有足够的可扩展性，还可以解决一些极端的用例。

**Saron Yitbarek：**

在早期，来自于红帽、CoreOS 和谷歌的工程师们都参与到了 Kubernetes 的开发中。随着 Kubernetes 开发到 1.0，不管是初创公司还是大公司都参与其中，一起构建和完善它。关键的是，所有这些增长从来都不是只归功于谷歌或者任何一方。

**00:13:30 - Clayton Coleman：**

在这个例子中，我喜欢以 Linux 打比方。Linux 并不是始于 Linus 开始编写内核，然后告诉所有人，在用户空间如何写 GCC，如何去建立 NGINX 或者 Apache。相反，内核团队专注于建立一个高效的操作系统内核，并与其他诸如 GNU 项目的开源社区合作，并且将可以在其他 Unix 系统上工作的工具引入 Linux。

**00:14:00：**

因此，我们如今所使用的许多工具，都不是 Linux 核心团队交付的。

但是 Linux 作为一个整体，相比于其内核涵盖的范围要宽泛得多，而且我认为这种模式的优势是 Kubernetes 取得现在成就所不可或缺的。当我们建立社区并且专注于 Kubernetes 范围时，我们可以试图

从“Kubernetes 内核”的角度来考虑它，这是分布式集群操作系统的内核。

**00:14:30 - Saron Yitbarek：**

Kubernetes 证明了自己在开源世界中建立社区的能力，令人难以置信。正如我们在操作系统之战中谈到的 Linux 崛起一样，现如今这场关于容器的战争中，获胜者往往懂得如何借助社区力量。事实上，尽管谷歌可能开创了 Kubernetes，但目前它属于每一位开发者，并由云原生计算基金会（CNCF）管理。

**00:15:00：**

在 GitHub 上，Kubernetes 有大约 3 万的星标数，而 Swarm 和 Mesos 只有数千，这已经很能说明问题了。这就是由社区所生，为社区所用的技术。

我想了解谷歌的态度，一个如此庞大并且以效益为导向的大公司，是怎么做到如此擅长跟其他开发者合作的呢？我找到了很适合回答这个问题的人 —— Kelsey Hightower，他是谷歌负责容器技术支持的技术专家。

**00:15:30：**

想想谷歌的地位：它在分布式系统领域具备丰富的经验，还运行着分布在世界各地的许许多多的服务器，因此它开发的 Kubernetes 似乎有着很大的优势，并且有信心一定能在这场容器竞赛中胜出。那么，当你想到 Kubernetes 和开源时，你是如何看待这种关系的？

**00:16:00 - Kelsey Hightower：**

我想当谈到基础架构工具，甚至编程语言时，大家没有什么选择 —— 你不可能用个专有工具，即使它很棒。如果它不是开源的，大多数人可能甚至都不会想去了解。而且我认为这也是大多数人会采用像 Kubernetes 这样的基础架构工具的原因，你可能会对自己说：“好吧，我们就要坚持使用这个版本四、五年，也可能我们需要根据自己的一些独特需求来对其进行修改。”

**00:16:30：**

一旦走到这一步，就很难说服企业接受，“嘿，每台服务器使用程序的价格是 200 美元，而且你看不到源代码，所以有需要的话也必须等我们来修改”。

那样的日子一去不复返了，所以我不确定是否真的可以在没有开源的情况下建立基础架构。开源的另一个意味是拥有一个与项目紧密联合的社区，所以我认为 Kubernetes 一开始就锁定了胜利。

**Saron Yitbarek：**

让我们回到这场容器竞赛。在这里不仅仅有你提到的 Kubernetes，还有 Docker 的 Swarm Apache 的 Mesos……

**00:17:00 - Kelsey Hightower：**

所以，我想当人们谈论容器竞赛时，我不确定竞争是否发生在我们和 Mesos、Docker 使用者之间。我认为，真正的竞争发生在争取目前没有使用容器的潜在用户身上。是的，你还在使用原生 Bash 脚本，你迷茫着，不知道自己该归属何方。这些尚未选择编排工具和平台之人的市场，比起已选择了 Mesos 或 Swarm 的一方，要多得多。

**00:17:30：**

这就是容器战争存在并将继续的原因，真正的关键点在于如何帮助最终用户。Mesos、Kubernetes 或 Docker Swarm 是否会成为寻求更好解决方案的人们的首选？这一切都还悬而未决（SIG 译注：现在已经尘埃落定，Kubernetes 取得了全胜），但我会告诉你，像我一样，在这个领域工作的工程师来说，如果你不考虑市场营销和供应商，我会使用这个短语“不同的公司，相同的团队。”

**00:18:00：**

我们为彼此开发了许多工具，最终以某种方式出现在其他产品中。没错吧？好主意就是好主意。没有理由说，“哦，这是 Mesos 的人正在做的事情，那就忽略吧”，这有点愚蠢。所以从技术和社区的角度来看，我们的想法需要交流。同时也需要竞争来迫使我们来进行独立思

考，然后最棒的点子就会浮出水面，接着我们再选择采用哪种方式来正确满足用户的需要。

**00:18:30：**

因此，就这场竞赛而言，仍处于初期阶段，而且这个事情本身不会带来利润。明白我的意思吗？我们不是直接向任何人销售这个产品，这更像是一个平台之间的游戏，对所有人开放，然后用户会选择满足他们需求的那个，这就是我认为 Kubernetes 在社区方面做得很好的地方，真正开放，真正能解决实际问题。

**Saron Yitbarek：**

听起来很棒啊。我喜欢这个想法：在同一个球队踢球，而不要管球队是在什么地方。你对于容器和编排工具，还有 Kbubernetes 的未来有什么展望吗？

**00:19:00 - Kelsey Hightower：**

是的，我在 KubeCon 上做了一次主题演讲。所有这些工具都很棒，它们就像是乐高积木，我们有 Kubernetes，你可以选择一种产品用于安全，选择另一种产品用于网络，但最终，作为开发人员而言，你所想要的只是检查你的代码，并希望你的代码可以某种方式以呈现在客户面前。而我认为 Kubernetes 还有容器都会作为底层技术或者成为类似 Serverless 这种技术的基础平台。

**00:19:30：**

这是我的代码片段，已经打包完毕了。所有的平台都会把你的代码片段，用容器包装起来，然后帮你运行，但是不需要向你公开所有这些过程。因此，在未来，我认为随着 Kubernetes 变得普及，容器的应用场景将从大大小小的供应商或个人，提升到云供应商，因为这些事情往往需要专业知识和软件投资。容器将会遍布各个角落，但同时也就此隐藏起来。它会随着应用场景的扩展而渐渐隐形。

**00:20:00 - Saron Yitbarek：**

Kelsey Hightower 是 Google 的员工开发人员。在 2017 年秋天，Docker 宣布支持 Kubernetes。他们并不是说就放弃 Swarm 了，只是决定与容器编排竞赛的明显赢家和解。

**00:20:30：**

并不只有它一方，Azure 和 AWS 都宣布了对 Kubernetes 的支持。与此同时，像 OpenShift 这样的 Kubernetes 发行版仍在不断发展。我们得到的是一个可以扩展，支持新的用例的 Kubernetes 内核，这些用例包括微服务或持续集成项目。

**00:21:00 - Clayton Coleman：**

这个生态系统在类似 Linux 的模式下能得到最好的发展，而且我认为我们正朝着这条道路迈进。因此，就像所有优秀的开源项目一样，相对于单打独斗，让每个人都能够参与进来构建更好的东西，那就算是成功了。

**00:21:30 - Saron Yitbarek**:

所有这一切都在快速发生着，毕竟，这是一场竞赛，而这正是我们期望能从开源中获得的东西。在我们才刚刚理解什么是容器时，第一轮几乎就结束了，

这是来自 Red Hat 的 Scott McCarty。

**Scott McCarty：**

回想一下两年前，容器镜像格式还是一个巨大的战场，然后回到六个月至一年前，容器编排就成为了下一个巨大的战场。紧接着，如果你看看 2017 年的 KubeCon 及前几周，几乎每个主要供应商都宣布支持 Kubernetes。因此，很明显 Kubernetes 在这一方面上获胜了。

**00:22:00 - Saron Yitbarek：**

这章关于容器战争的故事即将结束。就像容器技术的开始一样迅速。

**Scott McCarty：**

因此，Kubernetes 已经成为标准，其美妙之处是，现在的应用定义已经变得标准化了。因此，任何人都可以在这些 YAML 文件中使用 Kubernetes 对象并定义应用，这就是我们共同所追求的事情。事实上，对于容器技术足够处理处理大型扩展系统这件事，我已经期待了 20 年。

**00:22:30 - Saron Yitbarek：**

Kubernetes 的成功看起来板上钉钉，但即使竞赛尘埃落定，我们仍然面临更大的一些问题。容器是否会成为未来几年的默认选择？是否会促使更多的云原生开发？这些转变将促生哪些工具和服务上？以下是我们目前所知道的。

**00:23:00：**

社区将通过 CNCF 继续改进 Kubernetes，并作为它最重要的使命之一，我们将建立一套全新的容器技术。

容器已经催生了大量新的基础设施，伴随而来的是全新的服务的需求。举个例子让你感受下容器的整合程度和发展速度，仅 Netflix 每周就运行超过一百万个容器。毫不夸张得说，容器是未来的构件。

**00:23:30：**

这一整季的栏目中，我们一直在追踪开源运动的演变。首先看到 Linux 如何主导战场，以及开源理念是如何改变商业、工作流程和每日使用的工具。容器真的是开源运动中最重要的里程碑之一。它们具有很好的迁移性、轻量、易于扩展。

**00:24:00：**

容器技术很好地体现了开源的优势，开源项目自然而然也推动了容器技术的发展。这是一个全新的世界，我们不用再担心从不同计算机或者云间的迁移产生的隔阂。

**00:24:30：**

容器的标准化比任何人预测的都要快。接下来的一集，我们将转向另一场悬而未决的战争。这场云间战争史无前例地催生者行业重量级人

物。微软、阿里巴巴、谷歌和亚马逊四家云供应商的摩擦正在升温，随之而来的将是一场暴风骤雨。我们将会追随它们激发的闪电，和广受欢迎的几位代码英雄一起探讨云间战争。

**00:25:00：**

《代码英雄》是红帽公司推出的原创播客栏目。想要了解更多关于本期节目和以往节目的信息，请访问 redhat.com/commandlineheroes 。在那里，你还可以注册我们的新闻资讯。想免费获得新剧集的自动推送，请务必订阅该节目。

只要在苹果播客、Spotify、Google Play、CastBox 中搜索“Command Line Heroes”，或者通过其他方式收听，并点击订阅，这样你就能在第一时间知道最新剧集。我是 Saron Yitbarek。感谢您的收听，编程不止。

### 什么是 LCTT SIG 和 LCTT LCRH SIG

LCTT SIG 是 LCTT Special Interest Group 特别兴趣小组，LCTT SIG 是针对特定领域、特定内容的翻译小组，翻译组成员将遵循 LCTT 流程和规范，参与翻译，并获得相应的奖励。LCRH SIG 是 LCTT 联合红帽（Red Hat）发起的 SIG，当前专注任务是《代码英雄》系列播客的脚本汉化，已有数十位贡献者加入。敬请每周三、周五期待经过我们精心翻译、校对和发布的译文。

欢迎加入 LCRH SIG 一同参与贡献，并领取红帽（Red Hat）和我们联合颁发的专属贡献者证书。

---

via: https://www.redhat.com/en/command-line-heroes/season-1/the-containers-derby

作者：RedHat 选题：bestony 译者：lujun9972 校对：acyanbird

本文由 LCRH 原创编译，Linux中国 荣誉推出