

中国云计算开源产业发展白皮书

第二部分基于容器技术的产业

云计算开源产业联盟

OpenSourceCloudAllianceforindustry, OSCAR

(内容陆续补充, 最终版请以 12 月 2 日发布版为准)

2016 年 X 月

目录

版权声明	I
前言	II
参与编写单位	IV
主要撰稿人	IV

版权声明

本调查报告版权属于云计算开源产业联盟，并受法律保护。转载、摘编或利用其它方式使用本调查报告文字或者观点的，应注明“来源：云计算开源产业联盟”。违反上述声明者，本联盟将追究其相关法律责任。

云计算开源产业联盟

前言

随着近几年云计算的爆发式增长，开源技术也在云计算领域得到了新的发展契机，如 OpenStack、Docker、DC/OS 等。云计算开源产业联盟经过深入市场调研，对基于各种开源技术的产业及其各自在中国的市场的的发展进行了梳理，分析了发展瓶颈，并对发展方向做出了预测汇总形成白皮书。

《云计算开源产业发展白皮书第二部分：基于容器技术的产业》首先阐述了近两年异常火爆的容器技术的发展历程，分析了容器技术的特点和优势，介绍了容器技术主要的应用场景，最后分析了容器技术的发展方向。《云计算开源产业发展白皮书第一部分：基于 OpenStack 技术的产业》已经上传至联盟开源项目 GitHub：<https://github.com/opensourcecloud/manual>。

云计算开源产业联盟，是在工业和信息化部信息化和软件服务业司的指导下，2016 年 3 月 9 日，由中国信息通信研究院牵头，联合各大云计算开源技术厂商成立的，挂靠中国通信标准化协会的第三方非营利组织，致力于落实政府云计算开源相关扶持政策，推动云计算开源技术产业化落地，引导云计算开源产业有序健康发展，完善云计算开源全产业链生态，探索国内开源运作机制，提升中国在国际开源的影响力。

联盟目前由中国信息通信研究院、华为技术有限公司、北京易捷思达科技发展有限公司、联想（北京）有限公司、国际商业机器（中国）公司、中国惠普云计算集团、中国电信股份有限公司云计算分公司、中国移动通信集团公司、联通云数据有限公司、中兴通讯股份有

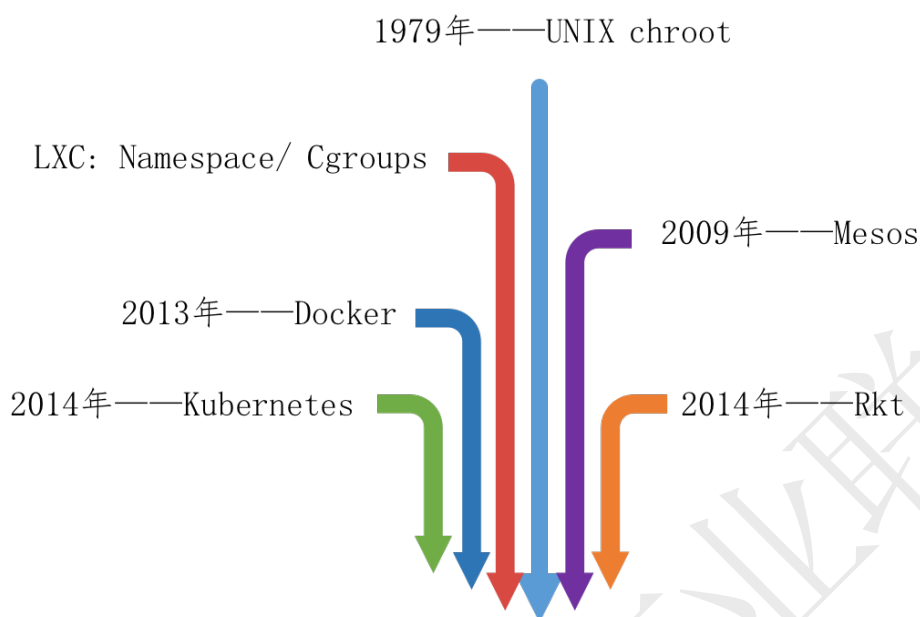
限公司、九州云信息科技有限公司、北京云途腾科技有限责任公司、烽火通信科技股份有限公司、上海优铭云计算有限公司、上海浪潮云计算服务有限公司、杭州华三通信技术有限公司（北京研究所）、杭州云雾科技有限公司、北京奇安信科技有限公司、北京中联润通信息技术有限公司、云栈科技（北京）有限公司、华云数据技术服务有限公司、航天信息股份有限公司、北京云基数技术有限公司、深圳市深信服电子科技有限公司、大唐高鸿数据网络技术股份有限公司、上海宽带技术及应用工程研究中心、天津南大通用数据技术股份有限公司、苏州博纳讯动软件有限公司、北京亚信智慧数据科技有限公司、国家新闻出版广电总局广播电视规划院、北京国电通网络技术有限公司、携程计算机技术（上海）有限公司、乐视云计算有限公司、中国银联电子商务与电子支付国家工程实验室 34 家单位组成。

参与编写单位

中国信息通信研究院、华为技术有限公司、北京易捷思达科技发展有限公司、联想（北京）有限公司、国际商业机器（中国）公司、中国惠普云计算集团、中国电信股份有限公司云计算分公司、中国移动通信集团公司、联通云数据有限公司、中兴通讯股份有限公司、九州云信息科技有限公司、北京云途腾科技有限责任公司、烽火通信科技股份有限公司、上海优铭云计算有限公司、上海浪潮云计算服务有限公司、杭州华三通信技术有限公司（北京研究所）、杭州云雾科技有限公司、北京奇安信科技有限公司、北京中联润通信息技术有限公司、云栈科技（北京）有限公司、华云数据技术服务有限公司、航天信息股份有限公司、北京云基数技术有限公司、深圳市深信服电子科技有限公司、大唐高鸿数据网络技术股份有限公司、上海宽带技术及应用工程研究中心、天津南大通用数据技术股份有限公司、苏州博纳讯动软件有限公司、北京亚信智慧数据科技有限公司、国家新闻出版广电总局广播电视规划院、北京国电通网络技术有限公司、携程计算机技术（上海）有限公司、乐视云计算有限公司、中国银联电子商务与电子支付国家工程实验室

主要撰稿人

一、 容器技术发展历程



主流容器技术演进路线图

容器技术是在单一主机上提供多个隔离的操作系统环境的操作系统级虚拟技术。容器概念最早可追溯至 1979 年的 UNIX chroot，它是一个 UNIX 上的系统调用，用于将一个进程及其子进程的根目录改变到文件系统中的一个新位置，使这些进程只能访问到该目录¹。

第一个较为完善的 Linux 容器解决方案是 LXC（Linux Container），它可以提供轻量级的虚拟化，以便隔离进程和资源，而且不需要提供指令解释机制以及全虚拟化的其他复杂性。LXC 有两个关键组件 Namespace、Cgroups。Namespace 是 Linux 内核的一种分类模式，方式函数和变量因名称相同出现冲突。Cgroups 是 Control Groups 的缩写，是 Linux 内核提供的一种可以限制、记录、隔离进程组所使用的物理资源的机制，是 LXC 为实现虚拟化所使用的资源管理手段。

¹ Linux 容器技术史话：从 chroot 到未来 <https://linux.cn/article-6975-1.html>（编译于 <https://dzone.com/articles/evolution-of-linux-containers-future>）

2009 年，Benjamin Hindman、Andy Konwinski 等人联合发起了 Mesos 项目，旨在尝试和提高集群的利用效率和性能。该项目目前设在 Apache 基金会，Twitter、eBay、Netflix 等均为 Mesos 的用户。2012 年，Mesos 的创始人之一 Benjamin 创立商业公司 Mesosphere，该公司提出了基于 Mesos 的 DCOS，用于像管理一台计算机一样使用整个数据中心。

2011 年，VMware 发布了业界首个开源 PaaS 云平台 Cloud Foundry，它支持多种框架、语言、运行时环境、云平台及应用服务，使开发人员能够在几秒钟内进行应用程序的部署和扩展。Cloud Foundry 可以简化现代应用程序的开发、交付和运行过程，在面对多种公有云和私有云选择、符合业界标准的高效开发框架以及应用基础设施服务时，可以显著提高开发者在云环境中部署和运行应用程序的能力。

2013 年，dotCloud 公司（后更名为 Docker）推出 Docker 项目，在容器技术的基础上，引入分层式容器镜像模型、全局及本地容器注册表、精简化 REST API；并推出了 Swarm 编排工具。2015 年，Docker 与 Linux 基金会联合成立 Open Container Initiative (OCI) 组织，并将容器引擎 RunC 捐献给基金会。2016 年，Docker 整合进 Windows 容器项目，成为最早以原生方式支持 Windows 平台上运行的容器技术之一。从 Docker 开始，容器技术开始具备了标准的格式和容器单元的统一大小。

2014 年，CoreOS 开发并推出 Rocket (rkt)，致力于打造比 Docker 更开放的技术。rkt 以组件的形式出现，目前已经得到了谷歌等企业的支持。

同年，谷歌公司推出开源项目 Kubernetes，提供了以容器为中心的部署、伸缩和运维平台。Kubernetes 支持 Docker、rkt 以及 OCI 等容器标准，能够实现在各种云环境中快速部署 kubernetes 集群。

二、 容器技术发展现状、特点和优势

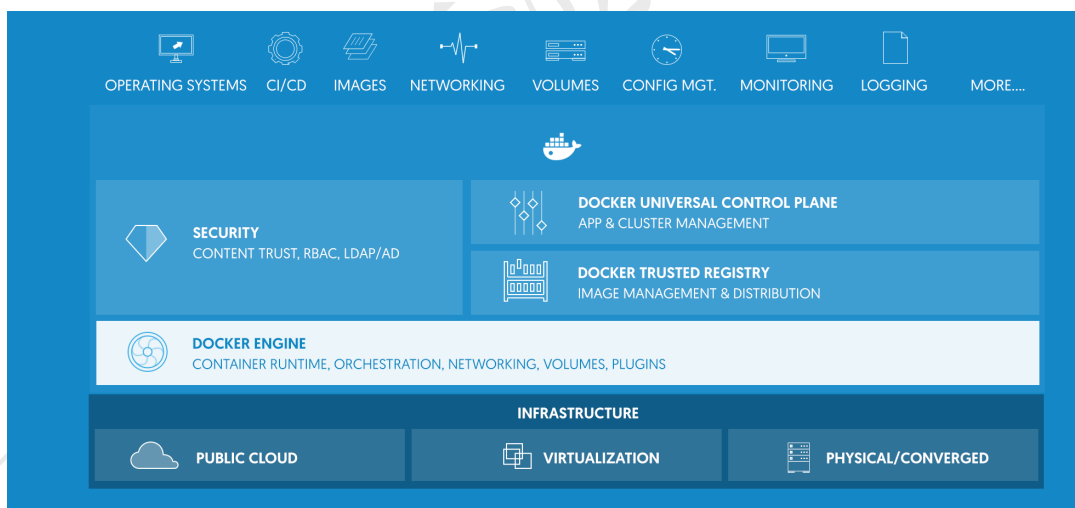
（一）容器技术分类

容器技术主要可以分为容器运行技术和容器编排技术。其中：

- 容器运行技术：主要包括 Docker、rkt、Cloud Foundry 等。

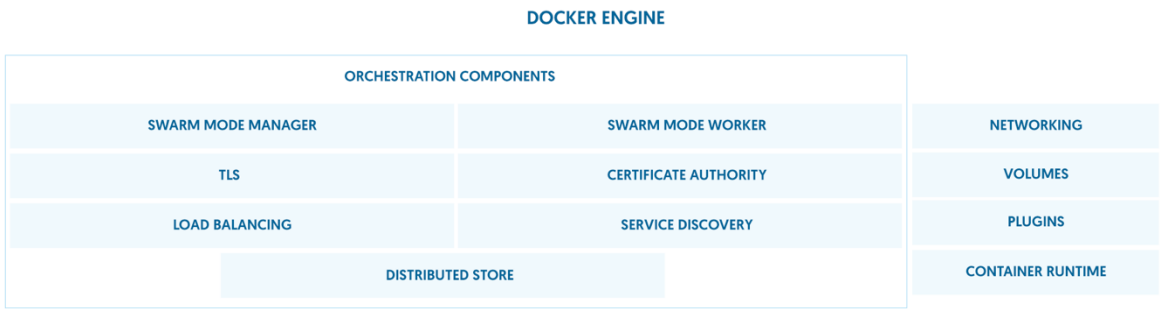
➤ Docker

目前，Docker 已经成为发展最快的容器技术。2014 年，Docker 围绕开源、轻量级和标准的 API 展开技术布局，随着 Docker 社区的不断壮大，Docker 的兼容性、隔离性、稳定性不断提高，并开始大规模投入生产环境中。目前，Docker 已经原生支持跨主机通信，技术架构趋于完善，生态体系已经基本形成。



Docker 架构图 (<http://www.docker.com/>)

Docker 包括 Docker 引擎（Docker Engine）、可信镜像注册（Trusted Registry）、控制面板（Universal Control Plane）等核心组件。其中，Docker Engine 为核心，部署在基础设施之上，支撑上层应用。Docker Engine 都具备内置的业务流程（Built in Orchestration），所以通过 Docker Engine 可以通过简单的指令，完成集群或工作流的管理。

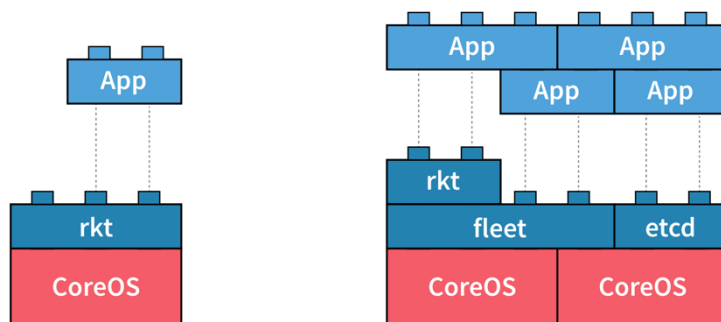


Docker Engine 架构图 (<https://www.docker.com/products/docker-engine>)

Trusted Registry 用于 Docker 镜像的注册，通过该模块，用户可以保证本地部署的私有云镜像安全可靠；通过 Universal Control Plane，用户可以管理和部署容器化的应用，并使这些应用运行在公有云或私有云的基础设施上。

➤ rkt

2014 年底，CoreOS 创立 rkt 项目，意在打造一套能够真正实现安全性、组合性与标准遵循效果的容器引擎。2016 年 2 月 rkt 发布了 1.0 版本，在镜像签名、权限隔离等方面做出了优化；2016 年 9 月，rkt 发布了截至目前（2016 年 11 月）的最新版 V1.19，对接口做出了优化。



rkt 架构图 (<https://coreos.com/>)

与 Docker 的运作模式相似，rkt 也是以 Engine 为核心，通过对镜像的下载、执行和监控，完成对应用的一系列操作。但 rkt 也与 Docker 存在较大的不同：

一方面，rkt 不仅支持 Docker 镜像格式，还支持遵从 Appc 规范的 aci (App Container Images) 镜像格式。

另一方面，rkt 与 Docker 的执行过程不同。在 1.11 版本之前，Docker 镜像的下载、运行、API 调用、日志监控等都由 Docker daemon 负责，虽然有利于快速的部署和实施，但是也导致了 Docker 的运行必须具备操作系统的最高权限；1.11 版本后，Docker daemon 虽然新增了 containerd 组件负责运行镜像文件和创建容器，使 Docker 容器在运行镜像和创建容器的过程中无需最高权限，但仍要通过具备最高权限的 Docker daemon 完成一系列的工作。rkt 在运行过程中始终采用标准的 UNIX 组权限分割不同操作，这使得 rkt 无需取得系统最高权限，即可完成镜像的下载和运行。

➤ Cloud Foundry

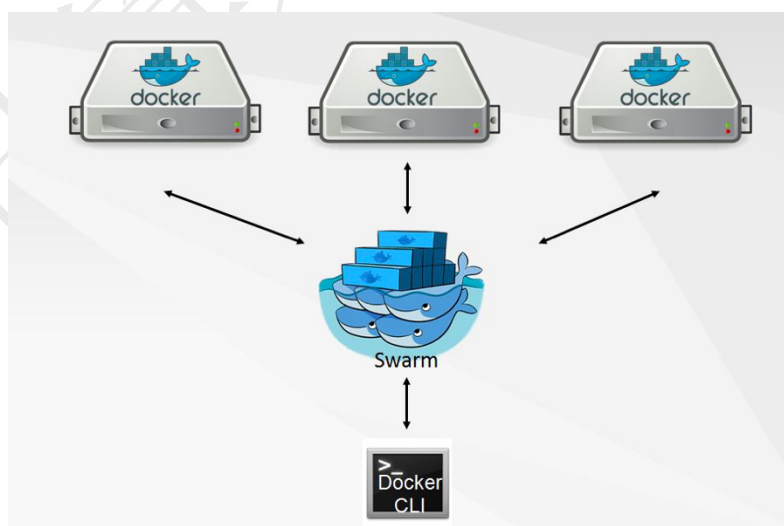
Cloud Foundry 是 VMware 推出的开源 PaaS 云平台，支持 .NET，Ruby on Rails，Node.js 等多种框架、Java、PHP、Python 等语言、运行时环境、云平台及应用服务。

Cloud Foundry 由 Router 、 DAE 、 Healthy Manager 、 CloudController、 Services 等模块组成。其中，Router 负责管理 Cloud Foundry 的全部通信；DAE 是 Cloud Foundry 的核心组件，负责将提交的源代码和需要的运行环境打包存储，当 Cloud Foundry 启动某一应用时，部署了该应用的全部 DAE 服务器都将根据设定好的源码和流程运行；Health Manager 负责收集每个 DAE 的运行信息，并根据收集到的信息作出分析、统计和报警；CloudController 负责管理 Cloud Foundry 的运行，保证各个组件正常运行；Services, Cloud Foundry 的服务层，第三方用户可以通过此部分将自己的服务整合进 Cloud Foundry。

- 容器编排技术：主要包括 Swarm、Kubernetes、Mesos 等

- Swarm

Swarm 项目是 Docker 集群的管理工具，将多个 Docker Engine 聚集在一起，对外提供 Docker 的集群服务。同时这个集群对外提供 Swarm API，用户可以像使用 Docker Engine 一样使用 Docker 集群。

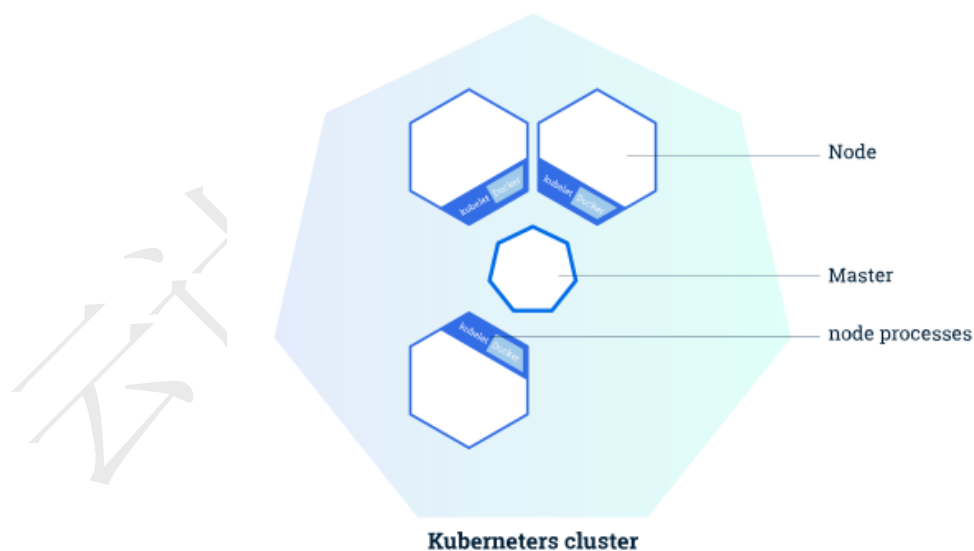


Swarm 运行示意图 (<http://dockone.io/article/662>, 作者：线超博)

Swarm 由集群管理、服务发现、集群调度等模块组成。其中，集群管理模块用于创建和管理集群，将各节点加入到 Docker 集群中，用 Docker 的命令实现 Docker 集群的统一管理；服务发现节点用于实现节点的发现，当单个 Docker engine 接入到网络中时，该模块获取节点的 IP 端口信息，管理模块根据发现模块获取的信息，将新发现的节点纳入 Docker 集群中管理；集群调度模块用于在 Docker 运行时，获取各个节点的状态，从中选取符合使用要求的多个节点，并从筛选出的节点中，根据资源最优或随机选择等策略选取出最优节点。

➤ Kubernetes

Kubernetes 是谷歌公司发布的一款编排工具，具有简洁、兼容、可扩展等特点。搭建 Kubernetes 至少需要 6 个步骤：创建集群、打包 app、测试 app、发布 app、集合 app、升级 app，在 Kubernetes 中，资源以 2 种方式存在：管理节点（Master）和子节点（Node），二者通过服务（service）建立联系。



Kubernetes 集群示意图

(<http://kubernetes.io/docs/tutorials/kubernetes-basics/cluster-intro/>)

其中，管理节点（Master）用于管理集群，提供资源调度、app 管理、发布更新等，由 3 个组件构成：apiserver，Kubernetes 的入

口，封装了对核心对象的增删改查操作，以 RESTful 接口实现外部用户对内部组件的调用；scheduler，负责集群的资源调度，为新建的容器组（pod）分配物理资源；controller-manager，负责保证服务和容器间的映射都是最新且有效的。

子节点（Node）用于Kubernetes的具体运行，由2个组件组成：kubelet，负责管理容器的状态，包括启动、停止等基本操作，及资源的监控；proxy，负责为容器组提供代理，当不同的容器组之间发起通讯请求时，由 proxy 进行转发，当 service 发起请求时，由 proxy 建立代理。

➤ Mesos

Mesos 是 Apache 基金会下的分布式资源管理框架，可用于计算、存储、网络资源的分配，可运行 Hadoop、MPI、Spark、Hypertable 等多种框架（Framework）。Mesos 由主（Master）、从（Slave）节点组成，其中，主节点（Master）负责从节点（Slave）与框架间的通讯，并根据策略决定分配多少资源给 Slave；Slave 负责启动本地进程，并向 Master 反馈空闲资源。在容器的编排方面，Mesos 负责资源的管理，Docker/rkt 负责进程的管理，二者与文件系统、分发系统等共同组成一套容器系统。

（二）容器规范

目前，主流的容器规范有 OCI 和 appc：

Open Container Initiative（OCI）：2015 年 6 月，Docker 公司与 Linux 基金会联合成立的项目，旨在“制定并维护容器镜像格式和容器 runtime 的正式规范，以达到让一个兼容性的容器可以在所有主要的具有兼容性的操作系统和平台之间进行移植，没有人为的技术

屏障的目标”。目前，该规范已经得到了谷歌、微软、华为等厂商的支持。RunC、TODO 等是基于 OCI 的技术。

App Container Specification (appc)：由 CoreOS 发布，得到谷歌、VMWare、Redhat 等公司的支持，rkt、Jetpack、Kurma 等是基于 appc 标准的技术。

该两项规范具有很多共性²：

- 可组合性：容器由独立运行的个体组成，容器的框架和 runtime 不绑定客户端和运行环境；
- 可移植性：容器可以在不同的硬件平台、操作系统和云平台上运行；
- 安全性：支持强大的密码算法、镜像审计和应用认证；
- 分散性：容器镜像的分发和发现（discovery）是简单的、支持联合命名空间和分布式检索，使容器能够实现快速部署；
- 开放性：容器的格式和 runtime 由社区统一设计，保证容器运行工具能够同时运行多个容器；
- 极简性：规范规定了容器的开发必须围绕简洁和稳定；
- 兼容性：新的容器格式需要实现向下兼容。

两种规范虽然同样致力于建立简洁、易用的框架，并消除厂商锁定，但两种规范间也存在一些不同，如：appc 规定了容器的打包规范，OCI 仅定义了目录结构；appc 定义了镜像的传输机制，如 Https、bittorrent 等。

（三）容器技术特点及优势分析

² OCI 官方社区：www.opencontainers.org

与虚拟化技术不同，容器需要部署在 Linux、Windows 等操作系统上，部署在相同物理资源上的容器共享资源。相比虚拟化技术，容器技术具有以下优势：

➤ 部署便捷。传统虚拟化的部署模式通常是安装到配置，再到运行，容器的部署则是根据镜像模板进行复制，通过一行代码完成部署。此外，容器的镜像通常在几十 M 左右，但虚拟机的镜像文件一般在数个 G 至数十个 G 之间，同时，除在本机部署镜像外，容器还支持调用远程的镜像文件，仅需占用少量的带宽资源。因此，相比虚拟化技术，容器技术提升了对内存、存储、带宽资源的利用效率；在运行过程中，由于体量较小，容器通常可以实现秒级启动，虚拟机则需调用大量的资源，启动时间在 20 分钟左右。

➤ 管理便利。容器技术可通过简单的命令行，完成对单一容器的管理，完成对镜像的快速打包和迁移；同时也能通过 Kubernetes、Swarm 等工具，实现对大规模容器集群的管理。

➤ 利于微服务架构的实现。微服务架构是指将单一应用程序划分成一组小的服务，通过拆分之后的服务之间的配合，为用户提供最终服务。拆分后的每个服务运行在独立的进程中，服务间采用 RESTful 接口进行通信。使用容器部署微服务，有助于整体架构的实现，将容器作为一个微服务的独立运行环境，能够节约搭建微服务运行环境的成本；容器的秒级创建和运行有助于微服务的构建；容器的编排工具有助于实现微服务的快速组合和调度。

➤ 弹性伸缩。由于容器单元间相互独立，切由统一的编排工具管理，且编排工具具备发现容器节点的功能，所以容器的弹性扩容可以在短时间内自动完成；同时，由于每个容器均为独立的个体，容

器调用的资源和容器的使用由编排工具管理，所以减少某一容器节点不影响整个容器系统的使用。

➤ 高可用。与弹性伸缩类似，在某一容器节点出现故障时，容器编排工具能够及时发现节点的变化，并根据外部请求情况及时作出调整，不影响整个容器系统的使用，实现系统的高可用。

（四）与 OpenStack 的结合

OpenStack 已经占据了开源云计算平台的绝大部分市场，早在 2014 年，OpenStack 在市场中的占有率就已经达到 69%，远超 CloudStack、Eucalyptus；根据 OpenStack 基金会的统计，绝大多数大中型企业都在使用和部署 OpenStack 解决方案。OpenStack 在快速占领市场的同时，也加入了许多与容器相关的重要新特性：

- Magnum - 提供容器环境的编排、部署与管理，即：COE(Container Orchestration Engines) as a service，目前支持 Kubernetes、Mesos 和 Docker Swarm；
- Kuryr - 集成 Libnetwork 等原生的容器网络组件，打通容器网络与 Neutron 网络；
- Murano - 提供应用目录服务，实现服务与应用程序的一键发布、快速部署和生命周期管理；
- Solum - 简化云应用程序从研发到交付的生命周期管理，为云应用开发者提供持续集成的能力；
- Manila - 文件共享服务，可以为容器应用的 Replication 和多读多写提供持久化存储方案；
- Kolla - 实现 OpenStack 容器化，即在持续集成环境中使用容器方式部署 OpenStack 组件，利用容器的快速部署和易于升级或

回退的优势，提升持续集成的效率，便于开发人员对集成环境的各组件进行快速调整或替换。

三、 国际容器社区及生态发展现状

目前，国际容器社区包括基于容器规范建立的 OCI 项目，以及 Docker 公司设立的 Docker 社区等。

OCI 项目 (<https://www.opencontainers.org>)，2015 年 6 月发布，是 Linux 基金会中的一个项目，用于指定容器及其运行状态、工具的规范，并将规范代码存储在 GitHub 上 (<https://github.com/opencontainers/>)，截止 2016 年 11 月 14 日，共有 Commits 2600 余次，活跃贡献者超过 200 名。社区每周三会通过电话举行 Open Weekly Meetings，讨论 OCI 相关的技术问题。目前，CoreOS 公司已经正式加入 OCI 项目，支持容器统一标准制定。

Docker 社区 (<https://www.docker.com>)，由 Docker 公司运行管理，主要依托社区网站和 GitHub (<https://github.com/docker>) 组织活动和收集贡献。目前，Docker 在 GitHub 上共有项目 104 个，包括 Docker、VPNkit、machine、Swarm 等，Commits 总数近 10 万次，活跃贡献者超过 1000 名。Docker 社区不定期在全球各地组织 Docker Global Mentor Week、MeetUp 等活动；每年组织 1-2 次 DockerCon，用于发布 Docker 最新成果，2016 年 6 月，DockerCon 在美国西雅图举行，会上 Docker 宣布了与 Windows、微服务的结合，并正式发布了 Docker 的 1.12 版本；2017 年的 Docker 将在美国奥斯丁举行，以工作坊、研讨会等形式重点开展技术方面的宣传。

原生云计算基金会 (The Cloud Native Computing Foundation，CNCF，<https://www.cncf.io>)，由 Linux 基金会于 2015 年发起成立，推动 Linux 容器技术在开发应用领域的实践，目前

项目包括 Kubernetes、OpenTracing（容器等厂商开放标准）等，会员包括华为、IBM、Redhat、Docker 等。基金会现有 16000 余名成员，已在北京、上海、柏林、伦敦等城市举行 Meetup54 场。

四、 我国容器技术产业情况

（一） 我国基于容器技术的产业发展情况

我国的主要容器厂商包括云栈科技（cSphere）、博纳讯动（博云）、灵雀云、时速云、DaoCloud 等。

云栈科技，全称云栈科技（北京）有限公司，是国内最早专注 Docker 容器的技术企业，定位于大型数据中心的 Docker 管理平台和 PaaS 解决方案，为客户提供一整套的开发运维自动化管控平台，产品主要包括容器 PaaS 解决方案、企业级容器操作系统等，解决方案支持微服务、互联网等架构、支持多基础设施，可在私有云、混合云或传统 IDC 环境下运行，解决方案具备编排引擎、部署引擎、健康引擎、监控引擎、调度引擎、配置引擎、构建引擎、服务发现引擎 8 个模块，为用户提供一站式的解决方案。

BoCloud 博云，全称苏州博纳讯动软件有限公司，为银行、证券、保险等企业级客户提供基于 Docker 的 PaaS 云架构产品和实施服务以及自动化运维平台，帮助企业客户降低成本、提升效率、简化运维、提高系统可靠性和安全性。产品具备自主、稳定、安全、可靠的特点，主要产品包括混合云管平台产品 BeyondSphere、企业级 PaaS 容器管理平台 BeyondContainer、自动化运维产品 BeyondBSM 等，已在民生银行、中国银联、江苏电力、江苏国资委等电力、政务、金融、IDC 等行业领域的生产系统中落地实施。

灵雀云，全称北京凌云雀科技有限公司，成立于 2014 年，由原微软 Azure 云平台的核心创始团队创立，在西雅图和北京设有研发中

心，旨在为客户提供最专业的企业级容器云平台（CaaS）。主要产品涉及自动化运维、开放测试管理、跨云管理等。

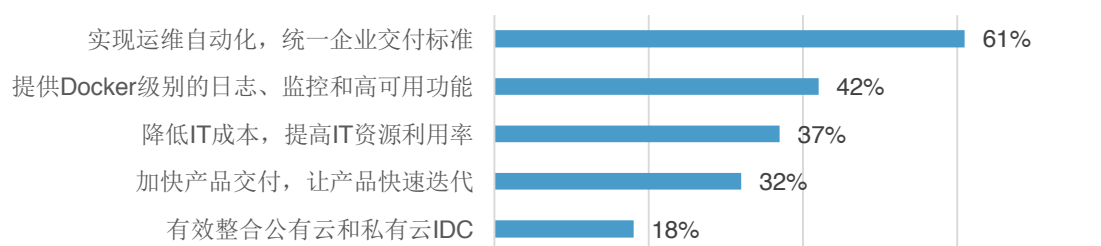
时速云，全称北京云思畅想科技有限公司，是国内领先的容器云平台和解决方案提供商。基于 Docker 为代表的容器技术，为开发者和企业提供应用的镜像构建、发布、持续集成/交付、容器部署、运维管理的新一代云计算平台。其中包括标准化、高可用的镜像构建，存储服务、大规模、可伸缩的容器托管服务，及自有主机集群混合云服务。

数人云成立于 2014 年 9 月，致力于为客户提供企业级容器解决方案，帮助传统企业实现 IT 业务转型，更好地应对业务变化。数人云重点聚焦打造轻量级 PaaS 平台，使用户能够在云主机、虚拟机或物理机上快速建立并稳定运行一个高扩展性的生产环境，并实现了一站式的微服务架构集群系统，最大化地帮助客户实现应用业务在云端的快速部署，解决应用上云的最后一公里。

DaoCloud 成立于 2014 年，是新一代容器云计算领域企业，产品线涵盖互联网应用的开发、交付、运维和运营全生命周期，并提供公有云、混合云和私有云等多种交付方式。覆盖金融、汽车、制造、教育、广电等多个领域，主要客户包括国家电网、东风汽车等。

（二）我国目前容器技术的主要应用场景

根据灵雀云 2016 年的调研，国内互联网企业希望借助容器解决运维自动化、节约 IT 成本等问题：



数字来源：灵雀云《2016 上半年中国互联网行业 Docker 和容器服务使用调查报告》

目前，我国容器厂商提供的容器产品、解决方案主要涵盖以下应用场景：

➤ 运维自动化

运维自动化需要使用自动化工具链，实现构建和测试的标准化流程。与传统的 Jenkins 等运维工具相比，容器将运维所涉及到的运维流程和工具链全部打包交付，由机器完成自动运维。

请厂商补充案例和架构图

➤ 弹性扩容/灾备

依托轻便的容量，容器可以实现秒级启动和释放。利用此特性，容器技术被广泛使用在了金融、电商行业，用于解决业务高峰期访问量快速提高导致资源不足的问题。

请厂商补充案例和架构图

➤ 持续集成和持续交付（CI/CD）

相比传统的虚拟化平台，使用容器技术搭建的 CaaS 平台具有开发效率高、资源调度和管理便利、版本升级和回滚快速等特点。容器化 CI/CD 解决方案能够实现源码的快速构建、镜像打包和应用部署。

请厂商补充案例和架构图

➤ 容器化软件交付

请厂商补充案例和架构图

➤ 开发测试

容器依靠轻量、便捷等特性，对于实现开发测试环境和生产环境的统一有着得天独厚的优势，消除了从开发到测试再到最终上线中不同环境对代码造成的影响。

请厂商补充案例和架构图

➤ 微服务

微服务是指将服务拆分成一系列服务组，拆分后的服务相互配合完成最终服务。容器的细粒度松耦合架构能够使用一个容器装载一个场景功能，即将每个服务或应用封装到每个容器镜像中，并在一个服务器上运行多个容器，通过这种分布式架构实现分布式的微服务。

请厂商补充案例和架构图

（三）我国容器社区发展情况

云计算开源产业联盟（OSCAR）成立于 2016 年 3 月，由中国信息通信研究院牵头，联合各大云计算开源技术厂商成立的，挂靠中国通信标准化协会的第三方非营利组织，致力于落实政府云计算开源相关扶持政策，推动云计算开源技术产业化落地，引导云计算开源产业有序健康发展，完善云计算开源全产业链生态，探索国内开源运作机制，提升中国在国际开源的影响力。联盟目前有会员 34 家，其中包括云栈科技、博纳讯动等专注于容器技术的国内先进企业。对于推动容器技术在国内的落地，联盟一是通过组织 meetup、研讨会等形式，建立客户和容器厂商之间沟通的平台，使客户了解容器技术的优势，使厂商了解客户的真实需求；二是通过标准的制定，建立容器云平台、容器解决方案的规范，填补国际社区和项目组对于平台搭建和解决方案构建方面标准和规范的缺失；三是组织容器解决方案的评估，根据容器的不同应用场景，从容器的基本功能、服务规范等方面，对解决方案的质量和服务商的服务质量进行评估，为用户选购容器解决方案提供详尽的参考。

五、 容器技术发展趋势分析

近几年，Docker、Kubernetes 等容器技术得到了迅速的发展，依靠自身轻便、可靠等特性迅速占领市场。但与此同时，容器技术还面临一些挑战，这都将成为容器技术将来的主要改进和发展方向：

（一） 技术方面

安全性有待提高。由于直接调用物理资源，所以当采用不可信的容器镜像或容器内核遭受攻击时，将会直接对宿主机的物理资源造成影响。目前，**Docker**等容器社区对此开展了镜像审计等工作，但因难以维护等原因并未整合至社区版本中。

编排系统的完善。**Docker**、**rkt**等容器运行技术均需要编排技术提升自身使用效率。但除了为容器提供管理和编排支持外，以**Kubernetes**为代表的编排系统正在加入更多的功能，如自行搭建一个容器运行环境，满足运行多样化的需求。

（二） 整体架构方面

工具的完善。目前，围绕容器的工具仅有**Kubernetes**、**Swarm**等编排工具，虽然具备监控和管理功能，但在某些场景下不能满足用户的需求；同时，由于是新生的开源工具，发布时间较短，所以多数情况下，厂商需要根据用户的不同需求，对工具进行定制化的二次开发。所以，具备完善功能的编排工具，以及现有工具在功能方面的不断完善，将是容器技术在接下来一段时间内发展的方向。

跨平台支持。随着数据中心规模的不断扩大，能够在多操作中心中热迁移的需求越来越大，但由于设计原理的限制，容器对于操作系统的依赖过大，仅通过容器技术很难实现跨平台的支持。对此，很多企业正在研发相应的解决对策，如目前使用较为广泛的容器+虚拟化解决方案等。

（三） 产业生态方面

投入使用规模。虽然经过几年来的发展，在市场上的应用逐渐成熟，但仍有很多企业处于投资成本、数据迁移等方面的考虑，仍在使

用传统的虚拟化技术，容器技术未市场中得到广泛使用，市场仍有很大的发展潜力。

容器标准统一。随着 2015 年 CoreOS 加入 OCI 计划，国际容器公司和社区开始加强了容器标准的统一，当 Docker 公司、CoreOS 公司将 RunC、appc 贡献给 OCI 时，容器技术的开放程度进一步提高，Docker 和 CoreOS、谷歌等主流容器公司均在致力于制定统一的容器标准。

社区发展趋于完善。相比 Linux 社区、OpenStack 社区等大规模社区，容器社区还处在发展的初期阶段：虽然具有较大的贡献数量，但组织的各类活动及活动的规模仍然较小。因此，容器社区的影响力并没有对容器技术的推广和发展起到有力地推动作用。但目前，围绕容器技术的 IaaS、PaaS 的解决方案已经较为成熟，OpenStack 等软件和 Linux、Windows 等操作系统对容器的支持，以及容器本身对 Hadoop、Spark 等的承载，标志着围绕容器的生态体系已初步形成。