



SH **C** Cassandra cloud computing cloud Infrastructure cloud-native ap  
cations Cloud Spanner cluster CNCF CNI Concourse CoreDNS containe  
ntainerize containerized application containers as a service container h  
ntainer registry controllers CI/CD continuous integration continuous de  
ery continuous deployment converged infrastructure **D** day one day two  
sired state DevOps developer-ready infrastructure digital transformat  
cker Docker Swarm **E** elastic ELK stack etcd **F** fault tolerance Fluentd  
P open service broker Gemfire Google Cloud Platform Google Kubern  
gine Greenplum Database GRPC **H** Hadoop Harbor Helm Chart horizon  
d autoscaler hybrid cloud hyper-converged infrastructure **I** image infra  
structure as a service (IaaS) ingress **J** Jaeger JSON **K** K8s KaaS Kafka kop  
bernetes kubectl kubelet Kubo **L** LDAP Lightwave linkerd **M** Memcache  
crosservices microsegmentation Minikube MongoDB MySQL **N** namespa  
dePort NoSQL NSX **O** OCI OpenTracing opinionated **P** platform orches  
tiation PaaS PCF PKS Photon OS platforms platform as a service platform  
veloper platform operator pod PostgreSQL prescriptive platform priva  
ud Prometheus pull **Q** quality of service **R** RabbitMQ RBAC Redis regis  
lica set repository refactoring repackaging replatforming rkt runC **S** so  
r service service discovery software-defined data center (SDDC) softw  
a service (SaaS) Spanner Spark spec Spring Cloud Data Flow StatefulS  
arm **T** tag the cloud traditional application **V** Vagrant Volume **W** workl  
XML **Y** YARN YAML **Z** ZooKeeper 12-factor app **A** ACID Active Directory  
S API server Azure Container Registry **B** build BOSH **C** Cassandra clou  
mputing cloud Infrastructure cloud-native applications Cloud Spanner  
CNCF CNI Concourse CoreDNS container containerize containerized a  
cation containers as a service container host container registry control  
CD continuous integration continuous delivery continuous deployment  
nverged infrastructure **D** day one day two desired state DevOps develo  
dy infrastructure digital transformation Docker Docker Swarm **E** elasti  
K stack etcd **F** fault tolerance Fluentd **G** GCP open service broker Gem  
e Google Cloud Platform Google Kubernetes Engine Greenplum Datab  
PC **H** Hadoop Harbor Helm Chart horizontal pod autoscaler hybrid clo

# 简介

本词汇表为云原生领域的术语提供了定义。这些定义并不是公理化的字典式定义，而是采用通俗易懂的语言描述术语的含义，并解释与术语相关联的技术为什么重要。部分术语的含义因用法、情况、角度或语境而异。



# 术语表



A  
ACID, 8  
Active Directory, 8  
Admiral, 8  
AKS, 8  
API 服务器 (API server), 8  
API 优先 (API-first), 9  
Azure 容器注册中心  
(Azure Container  
Registry), 9

---

B  
BOSH, 9  
编排 (orchestration), 9  
标记 (tag), 9

---

Calico, 10  
Cassandra, 10  
CI/CD, 10  
Clarity, 10  
Cloud Foundry 容器  
运行时 (CFCR) (Cloud  
Foundry Container  
Runtime), 10  
Cloud Spanner, 10  
CNCF, 10  
CNI, 10  
Concourse, 11  
CoreDNS, 11  
CredHub, 11  
超融合基础架构  
(hyper-converged  
infrastructure), 11  
持续部署 (continuous  
deployment), 11  
持续集成 (continuous  
integration), 11  
持续交付 (continuous  
delivery), 11  
初始 (day one), 11  
传统应用 (traditional  
application), 12  
重构 (refactoring), 12  
重新打包 (repackaging), 12  
存储库 (repository), 12

---

C

---

## D

DevOps, 12  
Diego, 12  
Dispatch, 12  
Docker, 13  
Docker Swarm, 13  
单元 (pod), 13  
底层网络 (underlay network), 13  
叠加网络 (overlay network), 13  
多云 (Multicloud), 13  
调度程序 (scheduler), 14

---

## E

ELK Stack, 14  
etcd, 14

---

## F

Flannel, 15  
Fluentd, 15  
服务 (service), 15  
服务发现 (service discovery), 15  
服务网格 (service mesh), 15  
服务质量 (quality of service), 15  
副本集 (replica set), 15

---

## G

GCP 开放服务代理 (GCP open service broker), 16  
Gemfire, 16  
Go, 16  
Google Cloud Platform, 16  
Google Kubernetes Engine, 16  
Greenplum 数据库 (Greenplum Database), 16  
GRPC, 16  
工作负载 (workload), 17  
构建 (build), 17  
固执平台 (opinionated platform), 17  
规范平台 (prescriptive platform), 17

---

## H

Hadoop, 18  
Harbor, 18  
Hatchway, 18  
Helm Chart, 18  
Horizontal Pod Autoscaler, 19  
函数即服务 (FaaS) (function as a service), 19  
后续 (day two), 19  
混合云 (hybrid cloud), 19

---

## I

Ingress, 20  
Istio, 20

---

## J

Jaeger, 20  
JSON, 20  
基础架构即服务 (IaaS) (infrastructure as a service (IaaS)), 20  
集群 (cluster), 21  
卷 (volume), 21

---

# K

K8s, 21  
KaaS, 21  
Kafka, 21  
Kops, 22  
Kubectl, 22  
Kubelet, 22  
Kubernetes, 22  
Kubo, 22  
开发人员就绪型基础架构 (developer-ready infrastructure), 22  
控制器 (controllers), 22

---

# L

LDAP, 23  
Lightwave, 23  
Linkerd, 23  
理想状态 (desired state), 23

---

# M

MANO, 24  
Memcached, 24  
Minikube, 24  
MongoDB, 24  
MySQL, 24  
敏捷软件开发 (agile software development), 24  
命名空间隔离 (namespace isolation), 24  
命名空间 (namespace), 25

---

# N

Node.js, 25  
NodePort, 25  
NoSQL, 25  
NSX, 25

---

# O

OCI, 26  
OpenStack, 26  
OpenTracing, 26

---

# P

PaaS, 26  
PAS, 26  
PCF, 26  
Photon OS, 26  
PKS, 26  
PostgreSQL, 26  
Prometheus, 27  
平台 (platforms), 27  
平台操作员 (platform operator), 27  
平台即服务 (platform as a service), 27  
平台开发人员 (platform developer), 27  
平台可靠性工程师 (platform reliability engineer), 27

---

# R

RabbitMQ, 28  
RBAC, 28  
Redis, 28  
Rkt, 28  
RunC, 28  
容错 (fault tolerance), 28  
容器 (container), 28  
容器化 (containerize), 28  
容器化应用 (containerized application), 28  
容器即服务 (containers as a service), 29  
容器主机 (container host), 29  
容器注册中心 (container registry), 29  
融合基础架构 (converged infrastructure), 29  
软件即服务 (SaaS) (software as a service (SaaS)), 29

---

## S

Sidecar, 29  
Software-Defined Data Center (SDDC), 29  
Spanner, 30  
Spark, 30  
spec, 30  
Spring, 30  
Spring Cloud Data Flow, 30  
Swarm, 30  
数字化转型 (digital transformation), 30  
私有云 (private cloud), 31

## T

Tern, 31  
Terraform, 31  
弹性 (elastic), 31  
提取 (pull), 31

## U

UID, 32

## V

Vagrant, 32  
Virtual Infrastructure Manager, 32  
VMware Integrated OpenStack, 32  
vSphere Cloud Provider, 32  
vSphere Integrated Containers, 33

## W

网络功能虚拟化 (NFV) (network functions virtualization), 33  
微分段 (microsegmentation), 33  
微服务 (microservices), 34

## X

XML, 34

## Y

YAML, 35  
YARN, 35  
映像 (image), 35  
有状态集 (StatefulSet), 35  
云 (the cloud), 35  
云计算 (cloud computing), 35  
云计算基础架构 (cloud infrastructure), 36  
云原生应用 (cloud-native applications), 36

## Z

ZooKeeper, 36  
注册中心 (registry), 36  
转移平台 (replatforming), 36

## #

12 要素应用 (12-factor app), 37

# A

## ACID

ACID 是 Atomicity（原子性）、Consistency（一致性）、Isolation（隔离性）和 Durability（持久性）这几项数据库事务属性的首字母缩写，这些属性共同保证了在出现电源故障或系统错误时数据的有效性。

## Active Directory

Microsoft Active Directory (AD) 是一项目录服务，可以对用户进行身份验证，并控制对个人计算机、服务器、存储系统、应用和其他资源的访问。Active Directory 域控制器将 Kerberos 密钥发行中心 (KDC) 与 LDAP 服务器相结合以提供身份验证和授权。为了验证用户的身份，AD 使用高度安全的 Kerberos 协议或旧版 NT LAN Manager (NTLM)。为了授予对资源的访问权限，AD 通常使用特权属性证书 (PAC)，PAC 是 Kerberos 请求单的数据结构，而 Kerberos 请求单包含以下信息：组成员资格、安全标识符以及有关用户概要的其他信息。请参见 LDAP。

## Admiral

Admiral 是一个容器管理门户，这是 VMware 的一个开源项目。可以为 DevOps 团队和其他人提供一个用户界面来调配和管理容器。借助 Admiral，您可以利用 Docker compose 将不同容器组合为一个应用。您也可以使用 Admiral 在 vSphere Integrated Containers 上管理容器主机并监管其使用情况，包括容量配额。Admiral 还可以显示容器实例的指标和其他信息。请参见 vSphere Integrated Containers。

## AKS

Azure 容器服务 (AKS) 是 Microsoft 提供的在 Azure 中运行的托管 Kubernetes 服务。

## API 服务器 (API server)

在 Kubernetes 中，API 服务器提供一种前端来处理 API “对象”（例如单元、服务和复制控制器）的 REST 请求和数据。



# B

## API 优先 (API-first)

一种开发软件的指导原则，将 API 作为应用的基石。从软件开发项目的一开始，就强调构建 API 以供客户端应用和服务使用，每一项功能要求都是通过 API 实现的。

## Azure 容器注册中心 (Azure Container Registry)

ACR 是 Microsoft 提供的专有映像注册中心，具有异地复制功能。

## BOSH

一种开源系统，可统一处理大型分布式系统的版本工程设计、部署和生命周期管理。BOSH 可执行监控、故障恢复和软件更新，且没有或仅有极短的停机时间。正如 Kubernetes 保持容器化应用的理想状态一样，BOSH 维持应用运行所在的底层基础架构（包括 Kubernetes 本身）的理想状态。

## 编排 (orchestration)

因为 Kubernetes 可以自动部署、管理和扩展容器化应用，所以通常被称为编排框架或编排引擎。它可以编排资源利用、故障处理、可用性、配置、理想状态和可扩展性。

## 标记 (tag)

在 Docker 中，标记是用户应用于 Docker 映像以将其与存储库中的其他映像区分开来的标签。



## Calico

正式名称为“Project Calico”，这一开源项目为容器提供由软件定义的第 3 层网络结构。它与 Kubernetes、OpenStack 和其他云平台相集成。

## Cassandra

Apache Cassandra 是 NoSQL 数据库，可管理跨商用硬件分布的结构化数据。常见使用情形包括建议和个性化引擎、产品目录、播放列表、欺诈检测和消息分析。

## CI/CD

指持续集成和持续交付管道，或持续集成和持续部署管道。往往可以通过上下文来判断缩写代表什么，但这种方法并不是每一次都有用。请参见持续集成、持续交付和持续部署。

## Clarity

VMware 提供了一种开源框架，正式名称为“Project Clarity”，这种设计系统可在 Web 应用中精巧地引入有粘性的交互式用户体验。Clarity 包括一个 HTML 框架和一个 CSS 框架、用户体验指导原则和 Angular 组件。

## Cloud Foundry 容器运行时 (CFCR) (Cloud Foundry Container Runtime)

之前称为 Kubo，Kubernetes on BOSH 的简称，这是一个开源项目，可使用 BOSH 部署和管理 Kubernetes。有关 CFCR 的更多信息，请参见 [CloudFoundry.org](https://CloudFoundry.org)。另请参见：BOSH。

## Cloud Spanner

一种分布在全球且高度一致的数据库服务，融合了关系数据库结构与非关系横向扩展的优势。

## CNCF

云原生计算基金会。CNCF 是隶属于 Linux 基金会的一个开源项目基金会，托管 Kubernetes 和其他关键开源项目，包括 Prometheus、OpenTracing、Fluentd 和 Linkerd。VMware 是 Linux 基金会和云原生计算基金会的成员。

## CNI

容器网络接口。它是由 CNCF 托管的开源项目，提供在 Linux 容器中配置网络接口所需要的规范和库。

## Concourse

Concourse 是旨在实现持续集成和持续交付的系统，可与 Pivotal Cloud Foundry 和其他平台配合使用，帮助企业开发团队尽早并频繁地发布软件。请注意，在 Concourse 语境中，CI/CD 中的 D 表示交付而不是部署。Concourse 会自动对频繁提交的代码进行测试和打包。请参见 CI/CD。

## CoreDNS

CoreDNS 是开源项目，可以与 Kubernetes、etcd、Prometheus 和其他软件集成，利用插件提供 DNS 和服务发现。CoreDNS 由 CNCF 托管。

## CredHub

Cloud Foundry 的一个组件，用于管理并安全地存储凭证，如密码、证书、证书颁发机构和密钥。CredHub 是通过 BOSH 部署的。请参见 Pivotal Cloud Foundry 和 BOSH。

## 超融合基础架构 (hyper-converged infrastructure)

与融合基础架构一样，超融合基础架构也会集成关键类型的 IT 组件，区别在于超融合基础架构将组件集成在一个可以简化管理、提高性能和提升弹性扩展能力的可扩展机架或设备中。请参见融合基础架构。

## 持续部署 (continuous deployment)

自动将应用或应用的一部分部署到生产环境。

## 持续集成 (continuous integration)

持续将来自不同开发人员或团队的源代码整合到应用中，然后进行测试。

## 持续交付 (continuous delivery)

通过打包和验证，为应用或应用的一部分做生产准备。

## 初始 (day one)

指部署。

# C

## 传统应用 (traditional application)

传统应用在设计方面比较单一，采用 n 层应用体系结构（通常包括数据库、应用和 Web 服务器）。这些组件通常与基础架构紧密关联，并依赖基础架构实现高可用性。

## 重构 (refactoring)

重新架构应用或修改应用的代码以改善应用。例如，可以通过将应用分解为微服务来重构应用。

## 重新打包 (repackaging)

采用容器格式安置传统应用。

## 存储库 (repository)

对于容器而言，存储库是一组容器映像。存储库可以通过注册中心服务器与其他用户共享，而“存储库”中的映像则可以采用标签进行标记。

# D

## DevOps

以方便、可靠且可持续的方式交付软件需要 IT 团队和开发人员进行协作。当开发人员和 IT 团队为了精简并自动执行开发和部署而共同致力于运维时，即是 DevOps。DevOps 是推动开发和部署云原生应用的关键实践。

## Diego

Pivotal Cloud Foundry 的容器管理系统。

## Dispatch

Dispatch 是 VMware 提供的开源框架，用于管理使用函数构建的无服务器应用和服务。作为无服务器解决方案，Dispatch 可使用灵活的驱动程序接口与一项或多项“函数即服务”实施进行集成。Dispatch 可管理构建于 Photon OS 之上的函数运行时和容器。请参见函数即服务。

## Docker

Docker 是一种广泛应用的容器格式。Docker 定义一种用于打包和移植软件的标准格式，就像标准集装箱定义货运标准一样。容器是 Docker 映像的运行时实例，包含以下三个部分：

- 一个 Docker 映像
- 一个执行映像的环境
- 一套用于运行映像的指令

## Docker Swarm

Docker 的独立原生集群工具。Docker Swarm 将多台 Docker 主机组合在一起，并作为单台虚拟 Docker 主机呈现。它提供标准的 Docker API，因此已经与 Docker 配合使用的任何工具都可以透明地纵向扩展为多台主机。

## 单元 (pod)

在 Kubernetes 上，单元是可在其中管理一个或多个容器的最小可部署单位，换言之，您在单元中运行容器映像。一组单元通常将一个容器、其存储资源、IP 地址和其他选项封装到将在 Kubernetes 上运行的应用实例中。Docker 通常是单元中使用的容器运行时。Kubernetes 管理员或应用开发人员使用 YAML 文件指定单元。单元通常通过部署进行管理，请参见相应术语。

## 底层网络 (underlay network)

底层网络可使用传统的基于硬件的网络连接方式，或组合使用硬件和软件，将虚拟机或物理机连接起来。另请参见叠加网络、服务网格和 NSX。

## 叠加网络 (overlay network)

大多数编排系统都会包含一个软件定义的网络连接组件，称为叠加网络。它叠加于底层之上，在容器和主机的生命周期中提供网络连接能力，例如 IP 地址和端口。叠加还可以在使用同一物理网络的应用之间隔离通信。叠加技术包括 Flannel、Calico 和 VMware NSX。另请参见 NSX、底层网络和编排。

## 多云 (Multicloud)

这种云计算方式可将多个云服务提供商、平台或服务结合于一个环境或体系结构中。多云战略可降低对单一供应商的依赖性，防止云计算服务出现故障，是您能够根据需要定制体系结构，还可在需要改变时灵活地切换解决方案。但使用多云平台和服务会让确保安全性、监管和合规性变得更加复杂。容器的移动性会促进多云战略的发展。

# D

## 调度程序 (scheduler)

系统模块或软件组件，可以调度并运行对容器、作业、任务或其他类型工作负载的部署。大部分公有云服务（例如 Microsoft Azure）都包括让您可以在云中创建作业的调度程序。这些作业进而可以调用服务或任务，例如备份数据或清除日志。

# E

## ELK Stack

Elasticsearch、Logstash 和 Kibana 相结合构成了 ELK Stack。这三个开源项目共同提供一个用于收集、搜索、分析和直观显示数据的平台。Elasticsearch 是一种分布式搜索和分析引擎，数据工程师可以通过此引擎查询非结构化数据、结构化数据和时间序列数据。您可以通过 Logstash 收集非结构化数据、进行补充并将其路由到其他应用，例如 Elasticsearch。Kibana 是一种可视化引擎，用于在仪表盘中以图形和地图的形式显示数据。

## etcd

分布式键-值对存储，Kubernetes 使用它存储数据及其相关状态和配置。

# F

## Flannel

Flannel 为 Kubernetes 中运行的容器提供叠加网络。请参见叠加。

## Fluentd

用于实现统一日志记录的数据收集器。Fluentd 由 CNCF 托管，可与云原生应用搭配使用。

## 服务 (service)

服务的定义因语境而异。在 Kubernetes 中，服务是 API 对象，描述如何使用端口或负载均衡器等方法来访问应用（例如一组单元）。

对于一些大型应用而言，服务也可能是微服务。例如，HTTP 服务器就是一项服务。

## 服务发现 (service discovery)

在运行使用微服务构建的云原生应用的环境中，服务发现会自动检测为服务动态分配的网络连接信息，或运行服务的设备。

## 服务网格 (service mesh)

如果容器化应用作为一组服务或微服务构建，即形成服务的网格。服务网格在 IP 地址和端口之上创建一个层次，用来连接服务并管理其交互。例如，服务网格可以提供负载均衡、监控和服务之间的身份验证。可提供服务网格的技术包括 Istio 和 linkerd。

## 服务质量 (quality of service)

通常缩写为 QoS。

## 副本集 (replica set)

在 Kubernetes 中，副本集是管理单元生命周期的控制器。请参见控制器。

# G

## **GCP 开放服务代理 (GCP open service broker)**

借助该代理，应用可从任意位置访问 Google Cloud API。

## **Gemfire**

Pivotal Gemfire 是分布式数据管理平台，它压缩运维数据并将其保存在内存中，以实现数据密集型 NoSQL 应用的实时、一致且可扩展的访问。

## **Go**

Go 是一种编程语言，在构建云原生技术的开发人员中很受欢迎，由 Google 创造。Docker 和 Kubernetes 都是用 Go 编写的，该语言有时也称为 golang。

## **Google Cloud Platform**

GCP。

## **Google Kubernetes Engine**

一种代管环境，用于部署和扩展由 Kubernetes 编排的容器化应用。

## **Greenplum 数据库 (Greenplum Database)**

符合 ACID 的事务性数据库。Pivotal Greenplum 采用无共享、大规模并行处理的体系结构，符合 SQL 标准。它能与行业标准商业智能和 ETL 工具以及 Hadoop 进行互操作。借助分析功能库和用于构建自定义功能的框架，Greenplum 能够处理大数据的数据仓库使用情形。

## **GRPC**

GRPC 是 CNCF 的项目，是适用于分布式系统的开源通用远程过程调用 (RPC) 框架。您可以使用它来定义服务，方法是使用 Protocol Buffers（一种二进制序列化语言）。您还可以通过 GRPC 自动为服务生成各种语言版本的客户端和服务端存根。



## 工作负载 (workload)

工作负载是与应用相关的一组计算、网络连接和存储任务的计算或事务性负荷。在不同情况下或不同时间内，采用相同技术和工具的分类应用的工作负载可能截然不同。工作负载通常根据 CPU 或内存使用量、网络流量、请求数量、数据库查询量、事务数量等进行衡量。非常简单地概括来说，应用是一种处理事情的手段；工作负载是进行的处理；使用情形是您进行处理的原因。对于云计算和 Kubernetes 集群而言，可以将工作负载视为应用的实例或应用的一部分在特定时间段执行的工作量。

## 构建 (build)

对于 Docker，“构建”是指使用 Dockerfile 构建 Docker 映像的过程。对于 CI/CD 管道，“构建”是生成构件（例如一组包含应用的二进制文件）的过程。

## 固执平台 (opinionated platform)

请参见规范平台。

## 规范平台 (prescriptive platform)

对于应用平台而言，规范平台规定开发人员构建和运行应用时所使用的系统格式、管道和方法，从而使开发人员能够规避平台的复杂性。例如，规范容器平台可能规定调度程序、运行时引擎、与底层基础架构的集成、持续交付，以及平台的其他方面。规范平台也称为“固执”平台。

# H

## Hadoop

Hadoop 包含 Hadoop 分布式文件系统 (HDFS) 和 MapReduce。HDFS 是一个专为 Hadoop 和大数据构建的可扩展存储系统。MapReduce 是一个处理框架，用于对存储在 Hadoop 分布式文件系统中的文件进行数据密集型计算分析。Apache Hadoop 是 Hadoop 的免费、开源版本，由 Apache 软件基金会管理。该开源版本为多个商用发行版奠定基础，包括 Hortonworks、IBM Open Platform 和 Cloudera，还有 Hadoop 平台即服务。Microsoft 提供 HDInsight 作为其公有云 Azure 的一部分。Amazon Elastic MapReduce (EMR) 通过 AWS 以 Web 服务的形式交付 Hadoop。

## Harbor

VMware 开发的开源项目，正式名称是 Project Harbor，它是一个安全注册中心，用于托管容器映像的存储库。

## Hatchway

VMware 的一个开源存储项目，正式名称为 Project Hatchway，它为 vSphere 环境中的容器提供存储基础架构选项，包括采用 VMware vSAN 的超融合基础架构 (HCI)。Hatchway 与 Kubernetes 相集成，您可以将存储基础架构作为代码使用。快照、克隆、加密、重复数据消除和压缩等数据服务可在容器卷的级别提供。请参见 <https://vmware.github.io/hatchway/>。另请参见 vSphere Cloud Provider。

## Helm Chart

Helm Chart 是 Kubernetes 资源包，这些资源是预先配置、自定义且可重现的；您可以使用 Helm 工具管理 Chart。这些 Chart 有助于改善 Kubernetes 应用的移动性。单个 Chart 可能包含整个 Web 应用，包括数据库、缓存、HTTP 服务器和其他资源。

## Horizontal Pod Autoscaler

在 Kubernetes 中，Horizontal Pod Autoscaler 是一个控制器，在对服务的请求超过管理员设置的阈值时，此控制器可以增加资源来应对需求增长。

## 函数即服务 (FaaS) (function as a service)

FaaS 是一种云计算模型，让您能够运行并管理应用函数，而无需管理传统的服务器作为应用的后端，“无服务器”也因此得名。函数通常会针对事件做出响应，这样 FaaS 就成为了物联网环境中处理事件的有用方法。我们来看一个示例：您在清晨醒来时，您的智能手表会监控您的血糖。如果血糖达到某一数值，手表就会向云中由供应商管理的远程服务器发送一个事件，该服务器通过 FaaS 框架使用函数处理事件，并向您家中联网的微波炉发送一个请求，触发它为您准备燕麦粥。AWS Lambda 是商用无服务器平台的一个示例。OpenFaaS 是 FaaS 框架的示例，用于使用 Docker 和 Kubernetes 构建无服务器函数。请参见 Dispatch。

## 后续 (day two)

指部署后运维。

## 混合云 (hybrid cloud)

任何涉及两个或更多个交付模式（例如私有云和公有云资源）且经过现代化改造的基础架构。



## Ingress

在 Kubernetes 中，Ingress 是指针对 Kubernetes 集群中服务（例如 HTTP 和 HTTPS）控制外部访问的 API 对象。Ingress 可以执行负载均衡操作。

## Istio

用于部署服务网格，以便连接、管理并保护 Kubernetes 上的微服务的平台。Istio 会拦截组成容器化应用的微服务（在 Kubernetes 上部署）之间的网络通信，从而管理微服务的交互。另请参见微服务、Sidecar 和服务网格。



## Jaeger

Jaeger 是 Uber Technologies 作为开源软件发布的分布式跟踪系统，可以监控基于微服务的体系结构。使用情形包括分布式事务监控、根本原因分析、服务依赖关系分析和性能优化。Jaeger 由 CNCF 托管。

## JSON

JSON 的全称为 JavaScript 对象表示法，是一种极简的数据互换格式，常用于注释数据（例如 API 输出）。

## 基础架构即服务 (IaaS) (Infrastructure as a Service (IaaS))

基础架构即服务 (IaaS) 提供对底层 IT 基础架构（包括用于存储、网络连接和计算的资源）的按需访问权限。用户可以在需要时利用 IaaS 调配 IT 服务，从而部署并运行任意软件。用户通常只需为所使用的资源付费。但用户无法管理或控制底层云计算基础架构。请参见云计算。

# K

## 集群 (cluster)

实际构成单个系统的三台或更多台互连虚拟机或物理计算机。集群中的计算机称为节点。在集群中运行的应用通常是分布式应用，因为它运行在多个节点上。集群本身便具备高可用性、容错能力和可扩展性，因此是云计算的关键部分。

## 卷 (volume)

在 Docker 中，卷（数据卷）是一个或多个容器中的指定目录，它可以绕开联合文件系统。卷旨在使数据保持独立于容器生命周期。

## K8s

Kubernetes 的缩写，8 是指“ubernete”这 8 个字符。

## KaaS

Kubernetes 即服务。

## Kafka

Apache Kafka 可对数据流进行分区，并将它们分散在分布式集群中的各台计算机上，以协调摄取大量数据进行分析的过程。更正式地讲，Kafka 是一种分布式发布订阅消息传递系统。Kafka 的主要用途是帮助 Spark 或类似应用处理数据流。在这样的使用情形中，Kafka 将数据流（例如来自不同服务器的日志文件）聚合成“主题”，并提供给可以实时分析数据的 Spark Streaming。

# K

## **Kops**

这个术语是 Kubernetes Operations 的缩写，它是可以帮助您安装、维护和升级 Kubernetes 集群的命令行工具。

## **Kubectl**

一种安装在计算机上的命令行界面，可用于运行那些控制和管理 Kubernetes 集群的命令。

## **Kubelet**

在 Kubernetes 集群中每个节点上运行以管理单元的代理。PodSpec 指定 kubelet 的工作方式。PodSpec 是描述单元的 YAML 或 JSON 对象。Kubelet 获取一组通过各种机制（主要通过 API 服务器）提供的 PodSpec，并确保这些 PodSpec 中描述的容器正在运行且状况良好。

## **Kubernetes**

一种用于自动部署和管理容器化应用的编排系统。Kubernetes 应用及其服务在分布式虚拟机或物理机集群上的容器中运行，它能够编排所有移动块，以优化计算资源的使用、维持理想状态并实现按需扩展。Kubernetes 又称为编排框架或编排引擎。请参见理想状态和编排。

## **Kubo**

请参见 Cloud Foundry 容器运行时。

## **开发人员就绪型基础架构 (developer-ready infrastructure)**

VMware vSphere、VMware NSX、VMware vSAN 和 VMware vRealize Operations 为 Software-Defined Data Center (SDDC) 奠定基础。例如，在 VMware SDDC 上运行 VMware Pivotal Container Service 或 Pivotal Cloud Foundry 会形成开发人员就绪型基础架构，即可用于构建和运行云原生应用的敏捷、自助式基础架构。

## **控制器 (controllers)**

在 Kubernetes 中，控制器是由 Kubernetes Controller Manager 启动的进程，用于执行与管理集群相关的日常任务。



## LDAP

轻型目录访问协议。它是用于存储和访问目录服务信息（尤其是用户名和密码）的标准协议。应用可以连接到 LDAP 服务器来验证用户和组。

## Lightwave

Project Lightwave 是 VMware 发布的开源安全平台，它通过提供目录服务、Active Directory 互操作性、Kerberos 身份验证以及证书服务来保护云计算平台的安全。借助 Lightwave，IT 安全经理能够在其云计算环境中实施本地计算系统的经验证的安全策略和最佳实践。具体来讲，Lightwave 包括以下服务：

- 目录服务和身份管理，具备 LDAP 和 Active Directory 互操作性
- 身份验证服务，支持的协议有 Kerberos、SRP、WS-Trust (SOAP)、SAML WebSSO（基于浏览器的 SSO）、OAuth/OpenID Connect (REST API) 和其他协议
- 证书服务，提供证书颁发机构和证书存储

## Linkerd

向云原生应用添加服务发现、路由、故障处理和可见性的服务网格。Linkerd 由 CNCF 托管。

## 理想状态 (desired state)

Kubernetes 的关键优势是它能自动维持理想状态，即管理员或平台操作员指定的应用应有的状态。

# M

## MANO

在电信行业和其他行业，MANO 是指对电信网络连接资源的管理和编排。在此语境中，MANO 包括 vRealize Suite: vRealize Operations、vRealize Automation、vRealize Network Insight 和 vRealize Log Insight。

## Memcached

作为将数据缓存在计算机集群的分布式内存中的系统，Memcached 通过将最近数据库调用的结果保存在随机访问存储 (RAM) 中来提升 Web 应用的性能。

## Minikube

一种工具，让您可以在虚拟机内部或个人计算机本地运行单节点 Kubernetes 集群。

## MongoDB

MongoDB 是一种分布式 NoSQL 文档数据库，它使用能够适应变化的、灵活且无架构的数据模型来存储数据。MongoDB 提供二级索引、地理空间搜索和文本搜索。常见使用情形包括为移动应用提供数据以及执行实时分析。

## MySQL

它是一种开源关系数据库管理系统 (RDMS)，常用于各种类型的应用，尤其是 Web 应用。它还广泛嵌入在由独立软件供应商 (ISV) 和原始设备制造商 (OEM) 分发的解决方案中。名字中的 SQL 是 Structured Query Language 的缩写，意为结构化查询语言。

## 敏捷软件开发 (agile software development)

强调迭代、递进、协作的开发方式，跨职能开发团队经过优化，可快速响应由于反馈、学习和新信息产生的不断变化的要求。

## 命名空间隔离 (namespace isolation)

限制可与容器连接或交互的资源。



## N

## 命名空间 (namespace)

对于 Linux 计算机而言，命名空间是内核的一种特性，用于隔离并虚拟化系统资源。仅限于在一个命名空间中运行的进程只能与同一命名空间中的其他资源和进程交互。

在 Docker 中，命名空间可以隔离网路连接和存储等系统资源。

在 Kubernetes 中，如果多个虚拟集群由同一个底层物理集群提供支持，则这些虚拟集群称为命名空间。

## Node.js

一种事件驱动型 JavaScript 运行时环境，在服务器上执行 JavaScript 代码，以生成 Web 应用中的动态内容。

## NodePort

在 Kubernetes 中，NodePort 表示 Kubernetes 集群中节点的端口上的一种用于进行外部访问的服务（例如 Web 服务器）。

## NoSQL

NoSQL 数据库存储那些采用传统关系数据库的表格形式以外的形式构建的数据。NoSQL 也称为非 SQL 数据库、非关系数据库和非仅 SQL 数据库。NoSQL 数据库常用于大数据和实时数据处理。常见的 NoSQL 数据库示例包括 MongoDB、Cassandra 和 Pivotal Gemfire。

## NSX

VMware NSX 是一款提供软件定义的网络虚拟化的产品。



## OCI

OCI 是 Open Container Initiative 的缩写，意为开放容器计划，是一个致力于制定整个行业的容器标准的组织。OCI 是在 Linux 基金会的赞助下成立的，目的是为容器格式和运行时创建开放行业标准。OCI 包含两种规范：运行时规范 (runtime-spec) 和映像规范 (image-spec)。VMware 是 OCI 的成员。请参见 <https://www.opencontainers.org/>。

## OpenStack

一种云计算开源平台，使您能够通过 API 和仪表盘控制计算、存储和网络连接资源池。请参见 VMware Integrated OpenStack。

## OpenTracing

不受供应商限制的分布式跟踪标准。由 CNCF 托管。



## PaaS

平台即服务。

## PAS

Pivotal Application Service。PAS 原名 Elastic Runtime，它在 Pivotal Cloud Foundry 上运行 Java、.NET 和 Node 应用。

## PCF

Pivotal Cloud Foundry，用于开发和部署云原生应用的专有平台即服务。

## Photon OS

VMware 发布的开源项目，Project Photon OS 是为运行容器而进行了优化的 Linux 操作系统。

## PKS

Pivotal Container Service，基于 Kubernetes 的容器服务。

## PostgreSQL

也称为 Postgres，它是一个可延展的对象关系数据库管理系统，可安全地存储面向 Internet 的大型应用或数据仓库的数据。Postgres 符合 ACID；请参见 ACID。

## Prometheus

Kubernetes 的开源监控系统。Prometheus 由 CNCF 托管。

## 平台 (platforms)

使用容器平台的首要业务目标是加速开发和部署易于修改、扩展、运维和维护的、可扩展的企业级软件。三种类型的平台为容器技术提供不同程度的支持：

- 用于运行各个容器实例的平台。用于运行容器实例的平台有助于开发人员构建和测试容器化应用。但是，此类平台不采用 Kubernetes 来编排容器化应用，也不提供服务代理，因此开发人员无法将工具、数据库和服务与应用相集成。VMware vSphere Integrated Containers 就是一个容器实例平台。
- 容器即服务。
- 平台即服务。

## 平台操作员 (platform operator)

负责管理 Kubernetes 等平台的工程师。

## 平台即服务 (platform as a service)

平台即服务 (PaaS) 是使用平台提供商支持或提供的编程语言、库、服务和工具，对应用进行开发、测试和运行的云端环境。平台即服务有时简称为应用平台。就这点而言，应用平台不仅可以帮助开发人员编写代码，还可以将工具和服务（例如数据库）作为微服务等与其应用相集成。Pivotal Cloud Foundry 就是一个也被称为应用平台的专有平台即服务。请参见容器即服务、基础架构即服务和云计算。

## 平台开发人员 (platform developer)

负责自定义 Kubernetes 平台（或其他现代平台）以满足项目或企业需求的工程师。

## 平台可靠性工程师 (platform reliability engineer)

负责 Pivotal Cloud Foundry 等平台运维的工程师。他们的工作可能包括管理灾难恢复、网络连接、存储、监控应用和服务代理。平台可靠性工程师的主要职责是建立并维护持续集成和持续交付管道。请参见 DevOps。

# R

## RabbitMQ

RabbitMQ 是开源消息传递代理，它实施高级消息队列协议来为应用提供一个通用中间平台，应用可以通过此平台建立连接和交换数据。

## RBAC

基于角色的访问控制。在 Kubernetes 上，RBAC 是按角色授予资源访问权限的模块。借助 RBAC，管理员能够通过 Kubernetes API 动态配置访问策略。

## Redis

Redis 是键-值对数据库，可以将数据集存储在联网的常驻内存的缓存中。因为 Redis 中的键可以包含字符串、哈希、列表、集、有序集、位图和 hyperlog，所以 Redis 通常被称为数据结构服务器。例如，数据科学家可以针对这些数据类型执行操作，以执行计算集合相交、联合和差异，以及排序等操作。

## Rkt

Rkt 的发音类似于 rocket，是 CoreOS 开发的基于标准的容器引擎。

## RunC

可启动容器的代码模块。它是 Containerd 的一部分，由 OCI 管理，OCI 表示开放容器计划。请参见 OCI。

## 容错 (fault tolerance)

容错是一种让系统能够在组件发生故障的情况下继续正常运行的属性。

## 容器 (container)

一种可移动的可执行格式，称为映像，用于打包应用及其所有依赖项和运行指令。执行容器映像时，容器映像计算机或虚拟机上以进程形式运行，并且具有自己的可自我描述的隔离式应用、文件系统和网络连接。容器的使用量与日俱增，这是因为它们提供可移动、灵活且可预测的方式来打包、分发、修改、测试和运行应用。容器可以加快软件的开发和部署速度。

## 容器化 (containerize)

指将应用打包到容器中。

## 容器化应用 (containerized application)

已打包的、在一个或多个容器中运行的应用。

# S

## 容器即服务 (containers as a service)

容器即服务平台有助于开发人员构建、部署和管理容器化应用，通常采用的方法是使用 Kubernetes 或其他编排框架（例如 Mesos 或 Docker Swarm）。

## 容器主机 (container host)

为运行容器而进行了优化的 Linux 操作系统。例如，CoreOS 以及 VMware 提供的 Project Photon OS。

## 容器注册中心 (container registry)

请参见注册中心。

## 融合基础架构 (converged infrastructure)

将为 IT 提供支持的不同基础架构元素（包括服务器、数据存储设备、网络连接功能、虚拟化、管理软件、编排和应用）融合在一起的技术。请参见超融合基础架构。

## 软件即服务 (SaaS) (software as a service (SaaS))

在通过网络（通常是 Internet）使用的云计算基础架构上运行的应用，而不是下载并安装在本地计算机上的应用。服务的使用者不管理或控制底层云计算基础架构或应用的功能。也称为 Web 应用。

## Sidecar

一种云原生体系结构模式，可将应用的组件或服务置于独立但位于同一位置的容器中进行隔离，从而实现独立性和灵活性。

## Software-Defined Data Center (SDDC)

基础设施在实施虚拟化之后以服务形式交付的数据中心。SDDC 的基础架构包括虚拟化网络连接和软件定义的数据存储和管理。SDDC 支持应用的方法比传统方法更加灵活、敏捷、有效和经济高效。在 SDDC 中，对基础架构的所有组件（计算、网络连接、存储、安全性和可用性）进行抽象化处理并作为自动化的策略驱动型软件进行交付。SDDC 可从根本上减少手动流程、加快 IT 服务交付速度、降低成本并提高投资回报。

# S

## Spanner

请参见 Cloud Spanner。

## Spark

Apache Spark 是可以从 Python shell 交互使用的大规模数据处理引擎。Spark 为共存于同一应用中的一套工具提供支持，因而兼具流式传输、SQL 和复杂分析功能。Spark 可以访问多种数据源，不仅包括 Hadoop 文件系统 (HDFS)，还包括 Cassandra 和 MongoDB。数据科学家喜欢 Spark，因为他们可以访问 Python 强大的数字处理库。

## spec

在 Kubernetes 中，spec 表示规范。规范是对理想状态（包括用户提供的配置）的描述。

## Spring

Spring Framework 是 Pivotal 提供的开源 Java 框架，可处理开发 Java 应用的基础架构。Spring 包括 Spring Boot、Spring Cloud 和 Spring Cloud Data Flow。

## Spring Cloud Data Flow

用于构建数据集成和实时数据处理管道的工具包。Spring Cloud Data

Flow 服务器使用 Spring Cloud Deployer 将管道与 Pivotal Cloud Foundry、Mesos 或 Kubernetes 相集成。Spring Cloud Data Flow 通过提供可以统一执行摄取、实时分析、批处理和数据导出的分布式系统帮助工程师开发分析管道。

## Swarm

在 Docker 中，swarm 是在 swarm 模式下运行的一个或多个 Docker 引擎的集群。但是，Docker Swarm 与 Docker 引擎中的 swarm 模式的功能特性并不相同。请参见 Docker Swarm。

## 数字化转型 (digital transformation)

优化对计算资源、组织流程和软件开发实践的使用以改进企业的自适应性、工作效率、创新力、竞争优势和全球覆盖面。概括来说，数字化转型通常需要采用新技术，其中包括云计算、移动设备、社交媒体和大数据分析。具体来说，云原生技术和实践（例如容器、Kubernetes、微服务、容器平台、DevOps 和 CI/CD 管道）融合成了一种强有力的数字化转型方法。

# T

## 私有云 (private cloud)

一种完全虚拟化的数据中心，它包括两项能提高敏捷性且不同于虚拟化数据中心的主要功能：自助和自动化。

## Tern

VMware 提供的一种开源工具，用于标识 Docker 容器中安装的程序包的源、版本和许可证。生成的清单有助于确保容器中的程序包符合许可要求。请参见 <https://github.com/vmware/tern>。

## Terraform

HashiCorp 开发的软件，用于在云服务提供商的环境（例如 Google Cloud Platform）中创建“基础架构即代码”。

## 弹性 (elastic)

可以动态扩展或收缩以适应需求波动的资源或服务。

## 提取 (pull)

将容器映像从注册中心下载到本地缓存，以便可以基于映像启动容器。

# U

## UID

可表示用户标识符、用户 ID 或唯一标识符，具体取决于上下文或系统。例如，在 Kubernetes 中，UID 是可唯一标识某一对象的字符串。

# V

## Vagrant

HashiCorp 的 Vagrant 将计算机的配置转换为可分发模板，以便为应用生成可预测的开发环境。

## Virtual Infrastructure Manager

VIM 用于管理与网络功能虚拟化基础架构 (NFV-I) 相关联的资源。请参见网络功能虚拟化。

## VMware Integrated OpenStack

由 VMware 支持的 OpenStack 发行版，可在 VMware 基础架构上运行 OpenStack。请参见 OpenStack。

## vSphere Cloud Provider

这是 VMware 提供的持久性存储解决方案，可支持有状态数据密集型容器化应用，包括具有数据库的应用。vSphere Cloud Provider 是通过 Project Hatchway 提供的，属于 Kubernetes 的一部分，使 VMware Pivotal Container Service 能够支持 vSphere 存储中的以下存储基元：卷、持久性卷 (pv)、持久性卷声明 (PVC)、存储类和有状态集。vSphere Cloud Provider 还可提供企业级存储功能，如基于存储策略的管理功能。请参见 Pivotal Container Service 和 Hatchway。



# W

## **vSphere Integrated Containers**

这是 VMware 提供的产品，可在 VMware vSphere 中创建容器基础架构，这样您就可以在虚拟机上运行容器化应用。该解决方案中包含一个运行容器的引擎、一个管理容器的门户、一个用户安全存储容器映像的注册中心，以及与 Docker 兼容的接口。vSphere Integrated Containers 的一项关键优势是，它可以将容器的移动性与 vSphere 的安全性、可见性和管理功能相结合。

## **网络功能虚拟化 (NFV) (network functions virtualization)**

NFV 这一概念是指将用于创建通信服务的网络节点功能虚拟化。在电信行业中，这是云原生设计战略的一部分。NFV-I 表示网络功能虚拟化基础架构。在使用 NFV 的电信环境中，虚拟网络功能 (VNF) 是一种特定功能，可与其他功能进行组合，以提供通信服务。VNF Manager 可控制一组 VNF 功能，包括将 VNF 与 NFV 基础架构进行集成，以及为其分配资源。

## **微分段 (microsegmentation)**

在 VMware NSX 中，微分段策略可以指定精细流量流模式，例如，运行容器化应用所在的 Kubernetes 命名空间中的精细流量流模式。借助微分段，您可以制定一些对工作负载强制实施安全要求以及在微服务级别隔离资源的规则。



### 微服务 (microservices)

用于构建应用的“现代”体系结构模式。微服务体系结构将应用的功能划分为一组小规模、独立、分散且以目标为导向的进程，对于其中每个进程，都可以单独进行开发、测试、部署、替换和扩展。请参见云原生应用。



### XML

可扩展标记语言。它是一种灵活且详细的格式，用于构建和交换数据。XML 通常用于旧版应用、Java 应用和 Web 应用以实现多种目的，例如构建配置文件或交换数据。尽管 XML 有时用于云原生应用，但 JSON 或 YAML（请参见相应术语）是首选数据格式。



## YAML

人工可读的数据序列化标准，通常在配置文件中用于构建信息和命令。在 Kubernetes 中，规范文件是采用 YAML 编写的。

## YARN

YARN 是 Apache Hadoop 的子项目，它将资源管理与计算处理分离开来，以便将交互性模式扩展到 MapReduce 之外，用于处理 HDFS 中存储的数据。YARN 在 Hadoop 应用（例如 MapReduce 和 Storm）执行计算时为其分配资源。事实上，YARN 通过为其他应用提供数据操作系统和可插拔体系结构，位于 Hadoop 环境的中心位置。

## 映像 (image)

在 Docker 中，映像是容器的基础。映像指定对根文件系统和要在容器运行时环境中使用的相应执行参数的更改。映像通常是相互堆叠的分层文件系统的组合。映像没有状态，而且永远不会改变。

## 有状态集 (StatefulSet)

在 Kubernetes 中，有状态集根据理想状态管理一组单元的部署和扩展。例如，有状态集可以管理有状态单元的持久性存储和其他资源。

## 云 (the cloud)

通过 Internet 提供的计算资源。请参见云计算。

## 云计算 (cloud computing)

云计算泛指通过 Internet 以服务形式按需交付的弹性共享计算资源和服务（例如计算能力、存储容量、数据库使用、分析和软件应用等），通常采用可计量定价方式。提供云计算的企业通常被称为云服务提供商。另请参见三种云计算服务模式的定义：IaaS、PaaS 和 SaaS。有关云计算的正式定义和分类，请参见 NIST 对云计算的定义，NIST 特别出版物 800-145，网址为 <https://csrc.nist.gov/publications/detail/sp/800-145/final>。



## 云计算基础架构 (cloud infrastructure)

包含服务器、虚拟机、存储系统、网络连接以及云计算和基础架构即服务所需的其他组件。云计算基础架构提供构造块或基元来创建可以交付云计算服务的混合云和私有云。

## 云原生应用 (cloud-native applications)

一般而言，云原生应用是指那些为了在云中作为分布式应用运行而开发和优化的应用。具体来讲，根据云原生计算基金会的定义，云原生应用（通常又称为“现代”应用）具有以下特征：

- 已经容器化，实现了可重复性、透明性和资源隔离。
- 已经过编排，以优化资源利用率。
- 细分为微服务，以简化修改、维护和可扩展性。

但是，不同的组织/企业有不同的定义。例如，Dell EMC 将云原生应用定义为高度可扩展的、遵循开源标准且具有动态性质的新一代分布式应用体系结构。

云原生应用通常在容器即服务平台 (CaaS) 或平台即服务 (PaaS) 上进行开发和部署。另请参见：12 要素应用。



## ZooKeeper

Apache ZooKeeper 可以像管理动物一样对分布式应用进行协调。它提供了一个注册中心以添加应用的名称。它可配置并同步应用。它可让应用远离混乱运行。

## 注册中心 (registry)

一种包含容器映像存储库的托管服务。例如，VMware 开发的开源项目 Harbor 就是一种注册中心。

## 转移平台 (replatforming)

将应用移动到另一个更高效的平台。如果正在迁移的应用是传统应用，并且新平台使用容器，则转移平台也涉及重新打包。



## 12 要素应用 (12-factor app)

一种开发软件即服务 (SaaS) 应用（即 Web 应用）的方法，通常将应用部署在平台即服务或容器即服务上。

- A
- B
- C
- D
- E
- F
- G
- H
- I
- J
- K
- L
- M
- N
- O
- P
- Q
- R
- S
- T
- U
- V
- W
- X
- Y
- Z

