

企业级云原生平台 驱动数字化转型

基于微服务、容器化和软件合规生产供应链
加速企业业务云化

易观分析
2022年7月

本报告联合发布方



目录

前言 03

数字化转型进入深水区,企业的IT架构面临系统性挑战 03

云原生平台全面升级IT架构,有效应对客户和场景规模激增带来的挑战 04

第一章 云原生平台对数字化转型的意义 05

数字经济成为中国经济新引擎,行业数字化转型进程加速 06

如何应对指数级增加的客户和场景,是决定企业转型效果的关键议题 07

转型成功的着力点是高效构建数字化应用,以支撑战略、业务、技术升级 08

当前IT架构难以满足新需求,企业需要构建起面向数字化时代的全新架构 09

在云原生架构下,云原生应用的构建效能显著提升,有效加速业务云化 10

CIO视角:云原生保障数字化转型的连续性、安全性和可审计性 11

第二章 企业级云原生平台的构成和价值 12

企业级云原生平台架构总览 13

支撑云原生架构特性的关键技术 14

云原生技术的演进趋势:持续收窄基础设施能力和开发者诉求之间的鸿沟 15

企业级云原生开发 - 立足微服务架构的业务应用开发和治理模式 16

企业级云原生运维 - 以容器化为中心的主动式应用运维和托管 18

企业级云原生安全 - 与数字化业务管理和流程耦合的软件合规生产供应链 21

云原生底座能力之计算 - 实现算力需求与算力资源的最优匹配 23

云原生底座能力之存储 - 运用符合云原生特性的存储方案和数据库 25

云原生底座能力之网络 - 保障企业级云原生架构的高可用性 27

企业级云原生架构顶层设计:搭建登录区,完成云原生架构体系落地 29

第三章 企业级云原生平台案例 30

大型股份制银行 - 零售业务基于云原生平台全面加速数字化进程 31

领先智能制造企业 - 利用Azure云原生平台为海外业务保驾护航 32

跨国保险金融企业 - 利用云原生平台在中国市场打响互联网战役 33

领先机器人流程自动化企业 - 基于云原生平台加速智能化和全球化战略 34

易观分析:企业级用户应将云原生架构和技术作为数字化转型的关键能力 35

前言：数字化转型进入深水区，企业的IT架构面临系统性挑战

数字化发展成为大中型企业必修课

今天，随着数字化基础设施和服务的普及，企业的产品和服务方式已经被根本性重塑。中国经济社会活动已经全面数字化，形成全球最庞大的数字社会。孵化出支撑经济运营的食、住、行、游、娱、购等各类数字化应用场景。

Covid-19进一步加速了数字化进程，面对外部环境的剧烈变化，那些处于数字化转型犹豫期的传统企业意识到，基于线下物理生产力和生产资料维系的生产经营活动存在很强的脆弱性。

为了提高应对危机的韧性和敏捷性，数字化转型成为企业发展的必选项。

数字化转型深化，业务规模化升级带来的复杂性冲击企业IT架构

企业推动数字化转型的切入点很多，比如技术的创新、场景的革新、流程的再造。但随着数字化进程的逐渐加深，企业的关注点不应只是单点突破，更需要全面开花。企业的数字化转型能否成功，取决于能否规模化实现企业经营能力的数字化改造和升级，并将更高效的数字经营能力体现到全方位的企业经营。无论转型的切入点是什么，单点的降本增效是不能系统驱动企业数字化转型目标的实现，规模化创新才是达成数字化转型战略的必选项。

规模化创新，意味着企业的研发、生产、供应链、营销、运营等各个业务环节都将进行数字化升级。然而，企业在云端开展数字化业务面对着很多新情况，比如电商促销时将面对海量、实时的数据存储、传输和计算需求激增的挑战。同时，业务边界和资源重构使得业务架构被重塑，业务系统也趋于复杂化，传统基于物理时空所构建的IT架构、流程和运维方法将面临剧烈冲击。

当前企业的IT基础设施，难以应对深度数字化业务下的规模化、高并发、数据智能等对数字底座带来的新挑战，严重掣肘数字化转型价值的充分实现。

为了让IT架构更具敏捷性、弹性和韧性，需要以管理和技术结合的全局视角、以及新一代的平台架构技术来升级基于当前的企业IT架构。打造基于云架构的云原生应用，让企业业务加速实现云端迁移，是企业转型的关键成功要素。



云原生平台全面升级IT架构,有效应对客户和场景规模激增带来的挑战

云原生平台全面升级IT架构,是充分发挥云效能的最短路径

云原生平台是数字经济时代企业转型的底层支撑,是推动企业上云的发动机。

应用规模上云带来了系统的复杂性,而云原生架构和技术最大程度降低业务系统的复杂性,实现了应用与运行环境的解耦、资源编排调度与底层基础设施解耦,打造了“新的计算形态”、“新的应用负载”和“新的物理边界”,以更先进的方式构建数字应用并赋能数字化应用全生命周期管理。技术用户可以利用云原生基础设施的微服务架构体系和开源标准,提升持续交付和智能自运维的能力,做到比传统DevOps更高的服务质量、更低的开发运维成本,让研发更专注于业务的快速迭代。

云原生架构将过去企业IT系统的支撑体系进行了虚拟化改造,大幅提升云计算效能。云原生架构包括技术功能,如容器管理、基础设施即代码、持续集成/持续交付(CI/CD)、服务网格和无服务器功能,发挥云计算基础设施的优势;同时提供应用程序架构、基础设施和运营(I&O)转型(文化、人员配备、工具和流程)方面的功能和连接,满足了企业发展对敏捷、弹性、持续、安全的需要。

构建企业级云原生架构,成为大型企业数字化转型的制胜关键

云原生技术和架构的价值得到越来越多的印证,越来越多企业愿意拥抱云原生进行数字化转型,市场上也涌现出大量的云原生技术厂商推动云原生市场生态的繁荣发展。

而对于企业管理者而言,应当思考如何选择并管理好各种云原生技术,保证在云端替代过程交付企业级标准的平台和产品来支撑业务稳健开展。这就要求云原生平台厂商从企业管理者视角出发,基于安全合规软件供应链,提供具备更高的稳定性、安全性、敏捷性、可监测、可解释能力的技术和产品。微软正在加大投入这些能力,满足客户的升级需求。基于Azure的云原生架构,企业的研、产、供、销、服等整个价值链路数据的打通和统一调度成为可能,加速企业的数字化进程。



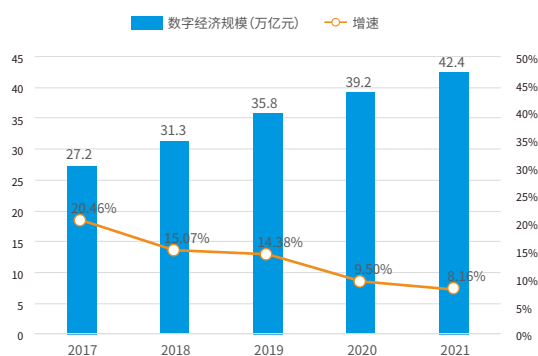
第一章 云原生平台对数字化转型的意义

数字经济成为中国经济新引擎，行业数字化转型进程加速

数字经济正在成为中国经济高质量发展的关键引擎

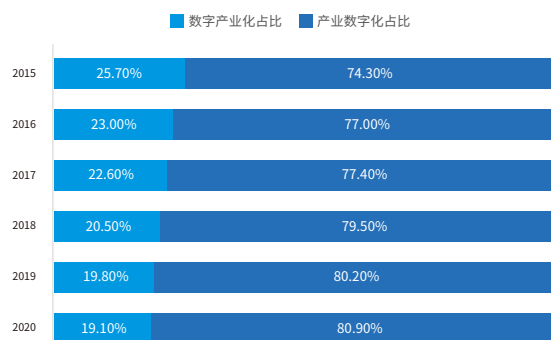
2022年1月，国务院印发《“十四五”数字经济发展规划》，是未来数年中国数字经济发展的总体纲领。易观分析测算，2022~2025年四年时间，中国数字经济核心产业总增加值约为5万亿元，成为支撑中国经济社会下一个阶段高质量发展的核心动力。¹

图：2017-2021中国数字经济规模及增速



© 易观分析 · 数据来源：信通院

图：中国数字经济内部结构



www.analysys.cn

传统发展模式难以维系，产业的数字化转型是关键增量

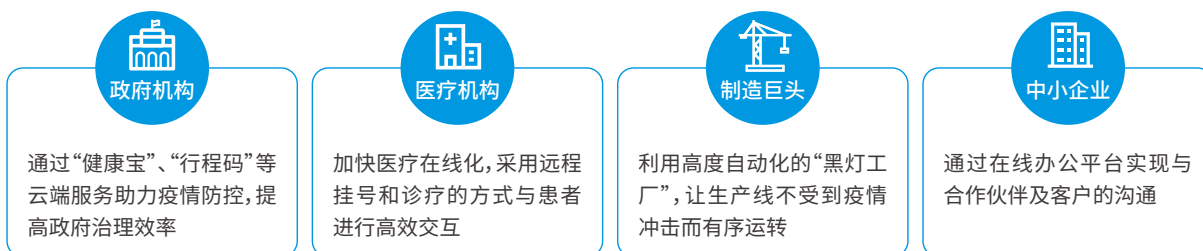
2012年易观分析首次提出“互联网+”之后，近10年中国经济生活各个方面已经完成了互联网化的改造，数字经济的核心产业得以形成。但随着内外部环境的显著变化，中国GDP增速持续放缓，迫切需要打造新型数字基建，实现新旧动能转换。在此趋势下，企业纷纷投入数字化、智能化转型的浪潮中，随之带来了产业互联网、信息科技、企业服务等一系列机会。

2022年，中国技术用户的CIO将持续增加对数字技术的投入，预计全年IT支出相较2021年同比增幅21%，保持了快速增长。75%的企业领导者表示，他们已经在立足数字化发展创造新的产品和服务。²

疫情的爆发和反复，凸显了掌握数字化能力的企业所具备的更强业务韧性

Covid-19以来，数字化转型在许多组织的推进速度明显加快。在这段快速变革的时期，企业内部为了保持业务连续性，需要采用不同的方式开展工作，最显著的变化是工作流向线上迁移的速度比以往任何时候都要快，掌握先进数字技术应用能力的企业已抢占先机。同时，社会生活、消费方式也越来越依赖在线服务，互联网人口数量的激增催生更多的数字化新业态、新模式、新应用，从而推动平台和技术厂商加快创新速度以满足日益增长的客户需求。

图：从落地情况来看，各行各业都在利用数字应用创新渠道、产品和服务，满足客户需求



¹ 数据来源：《2022年企业数字化技术应用十大趋势》，易观分析

² 数据来源：《100位CIO调研问卷，前瞻看2022年中国数字化转型趋势与IT投资机遇》，天风证券

如何应对指数级增加的客户和场景,是决定企业转型效果的关键议题

数字化转型是指重新构思如何将人员、数据和流程汇集起来,帮助企业在这个以“数字为先”的世界中保持竞争优势。

随着各个行业拥抱数字技术产业互联网化的趋势蓬勃加速,客户和场景数量和规模指数级增加,对于企业来说机遇和风险并存,而企业正在更加重视对于数字化转型的推进力度和速率,以更快形成应对数字化客户和场景的能力,抢先找到业务在数字经济时代的第二曲线。

在企业数字化转型的目标中,既要满足外部激增的数字化客户需求,又要构建自身业务的精益管理和运营。为了在激烈的竞争压力和不断变化的业务及市场需求中获得更强的韧性,企业在推进数字化转型工作时应重新审视其战略定位、组织文化、平台工具、治理方法和评估体系,在实现自身业务数字化后,利用数据智能优化客户旅程,获得业务增长。

图:企业的数字化转型面对指数级增长的数字化业务场景和客户需求

面对指数级增长的业务场景



业务管理 - 需要降本增效

企业为了获得持续性的发展与盈利,必须要聚焦内部业务场景优化。数字化转型符合企业降本增效的内在发展需求,可以支撑企业业务应用的标准化及快速定制化,实现以数字技术为核心要素、以开放平台为基础支撑、以数据驱动的精细化运营,改善以人治为核心的低效管控治理方式,降低管理成本,提升企业生产效率。



业务扩张 - 需要弹性支撑

企业需要不断扩张业务规模来立足或进入新市场,并不断稳固市场地位。数字化手段将有效帮助企业瞄准用户需求革新产品供给、拓展增值服务内容。将数字化的资源、知识、能力等进行模块化封装并转化为产品及服务,形成数据驱动的信息生产、信息服务新业态,覆盖更广的用户群体。

面对指数级增长的客户需求



市场竞争 - 需要迭代客户体验

现如今,企业或多或少面临行业高度同质化竞争问题。通过数字化手段挖掘客户价值并深度应用,有助于企业提高对客户需求预知的准确度、驱动科学投资决策、辅助判断产品/服务研发方向,革新运营及管理模式。数字化转型加速了企业积聚核心数据,缩短了交付周期,让企业获取差异化的竞争优势。



生态合作 - 需要构建数字底座

在数字化进程中,企业需要构建生态环境,通过群体竞争体系境来争夺行业话语权。通过打造开放的技术平台,实现数据、客户、业务资产的联通共享,做好行业协同与生态构建,充分发挥生态合作伙伴连接带来的“价值网络外部性”,快速扩大价值空间边界,才能横向扩张市场份额、纵向打造第二增长曲线。

易观分析认为,转型成功的着力点是高效构建数字化应用,以支撑战略、业务、技术升级

企业上云汇聚了大量的数据资源。企业逐渐拥有了收集、存储和分析海量数据的计算和存储能力,这使得企业利用数据要素来驱动规模化的数字化转型,打造战略、业务和数字技术应用的全方位升级成为可能。

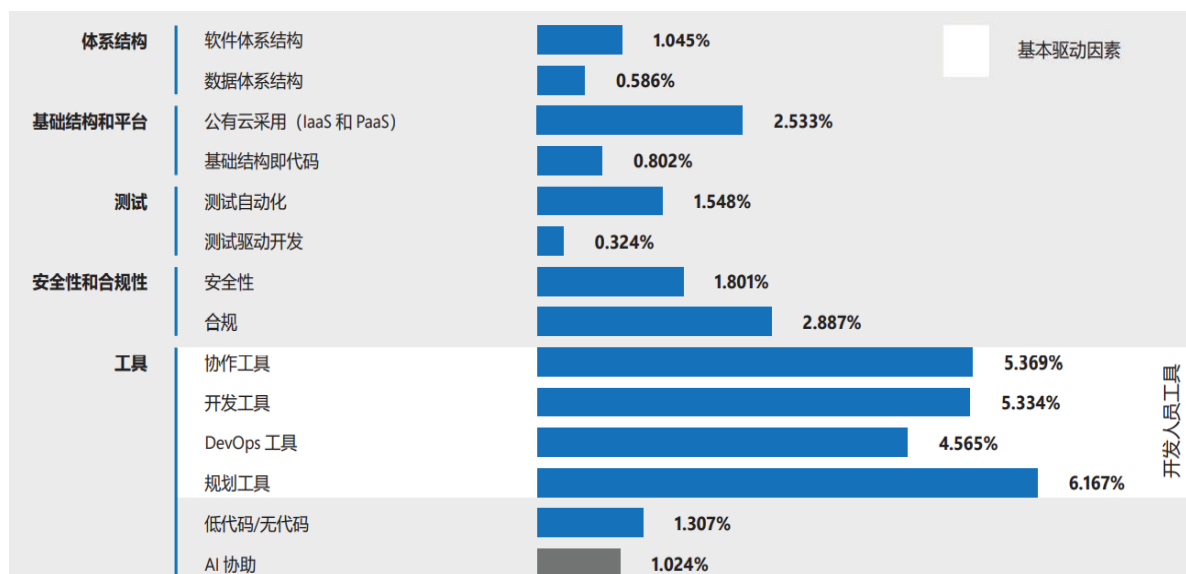
图:数字化转型需要战略、业务、数字技术应用全方位升级



为了追求上述转型价值,数字化转型先锋们应致力于利用数字化应用满足客户和伙伴需求。这是因为数字化应用可以端到端获得客户的反馈并持续优化,最终塑造出更全面和优质的客户体验。此时,云对业务的价值不再是简单的资源供应,而是以应用为中心赋能业务。

易观分析认为,在海量数据支撑下,企业的数字化转型需要更加强调**高效、规模化构建数字化应用**,以承载企业不同层级的数字化转型升级需求。

图:数字化转型需要战略、业务、数字技术应用全方位升级



数据来源:《2020-2021 年企业 DevOps 报告》

当前IT架构难以满足新需求,企业需要构建起面向数字化时代的全新架构

企业的转型方向:打造以数字应用驱动业务的数智化企业

为了高效构建数字化应用,企业需要重构面向数字化时代的架构,利用数字应用,链接战略、业务、技术和IT基础设施,形成驱动业务价值持续增长的良性闭环,进而驱动企业转型成功。

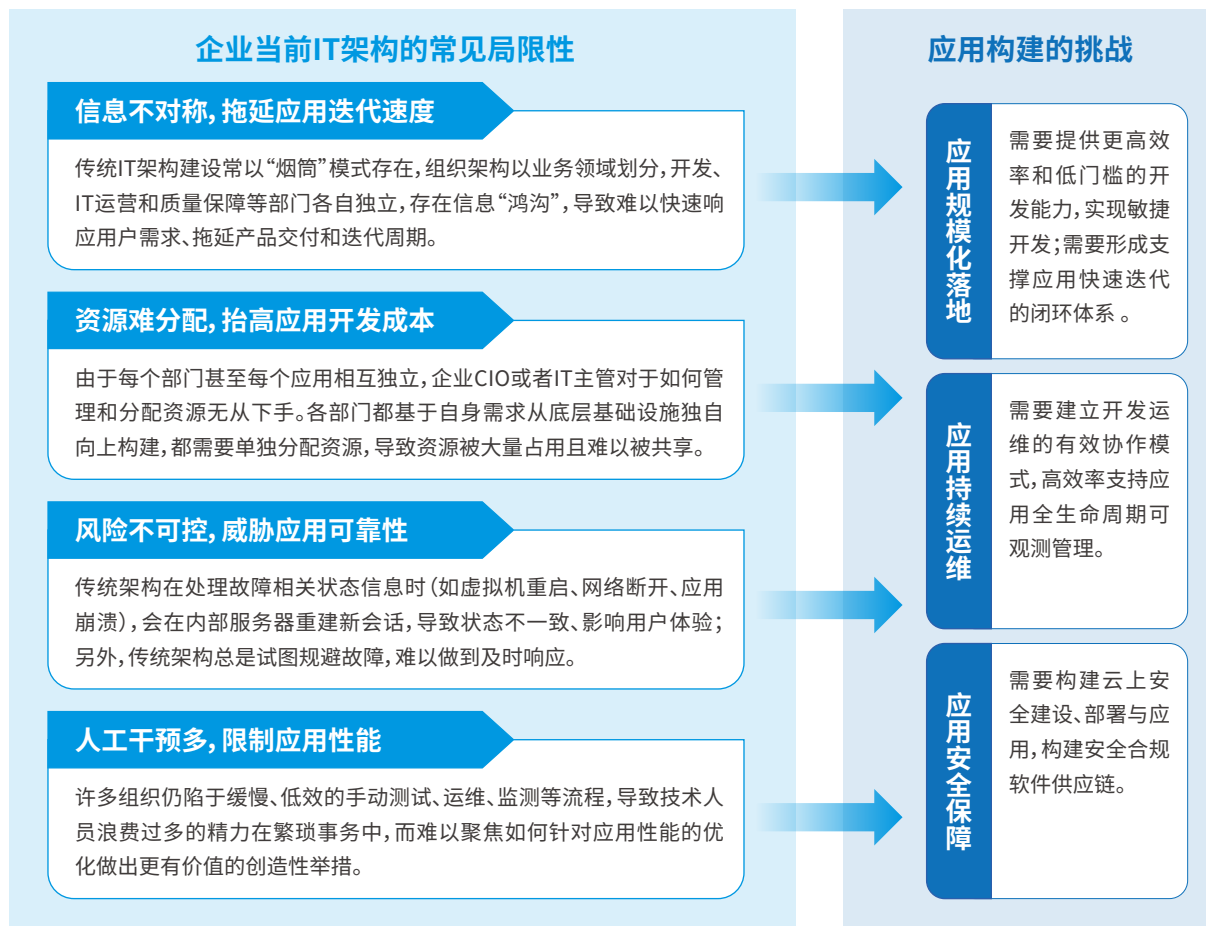
图:以数字应用为中心的智能化企业



当前市场主流的云计算和IT架构限制企业转型速度

当前云计算及基础架构技术难以支撑企业高效构建数字化应用的需求,导致企业难以应对外部不断变化的客户需求及各方挑战,以及难以满足保障业务连续性、提升运营效率的内生需求。传统IT架构制约了应用构建效率,导致了这三方面挑战:应用的规模化落地,应用的持续运维,应用的安全保障。这也是云原生时代到来的重要驱动力。

图:传统IT架构限制应用构建效率



在云原生架构下,云原生应用的构建效能显著提升,有效加速业务云化

云原生是基于以应用为中心的设计理念,提供了更高效的应用开发能力,可以有效支撑企业数字化转型进程中战略、业务、技术和IT基础设施的升级。

云原生可以帮助企业构建弹性可靠、松耦合、易管理可观测的应用系统,提升交付效率,降低运维复杂度。云原生的引入可以帮助企业屏蔽底层设施的异构,专注于应用的开发与部署,放大云的弹性、敏捷性等特征,向灵活性、轻量级架构演进。

云原生架构的价值主张保证了云设施的效能释放

云原生代表着速度与敏捷性,有效应对快速变化、大规模和弹性的复杂业务系统。云原生架构涉及以下若干的设计哲学,遵循这些原则可以帮助架构师或开发人员选择合适的技术路径,使得高效、规模化开发云原生应用成为可能。

图:云原生架构的设计哲学



弹性

- 弹性计算是云计算的核心特征,这也是云原生重要特征之一。弹性是系统对故障作出反应并仍然保持功能的能力,强调的是在系统发生故障时可以快速构建云原生服务进行及时应对,从而尽快恢复到完全正常运作的状态。
- 好的弹性能力可以保障系统的部署规模可以随着业务量的变化自动伸缩,无需受到固定的硬件及软件资源限制,保证了软件持续提供业务服务的能力,更好支撑业务的突发性扩张。



可观测性

- 云原生提供了评判应用运行状态及健康值的外部指标,保证了系统的可观测性。该特性宛如云原生架构的心跳监视器,能及时反映多维度的数据指标,帮助技术用户实时掌握软件的运行情况,并获得关联分析的能力。
- 可观测性可以解放技术用户的劳动性,脱身于收集数据等繁琐事务,,更加聚焦在对业务健康度和用户体验的优化。



基于自动化的韧性交付

- 云原生实现的开发运维一体化特性,实现了开发的自动化,使得软件的生产交付过程变成可复制、可批量化的生产流水线。同时持续集成/持续交付的自动化执行也执行了企业的流程和安全检查,提升了交付效率。
- 该特性一方面标准化了企业内部的软件交付过程,另一方面实现了软件交付和运维的自动化,允许技术人员付出极少的工作量就可以做频繁的、可预测的、高影响力的更改。

此外,云原生还包括其他重要设计理念,如服务化、零信任、架构持续演进、安全管控、高可用性等。企业通过采用基于云原生的技术和管理方法而生成的应用,就能够持续地享受到云计算提供的源源不断的高效服务。

对于企业CIO来说,云原生是业务转型连续性、安全性、可审计性的保障

云原生的价值在企业上云的浪潮中显得愈发明显,越来越多的企业和开发者开始把业务和技术向云原生推进。在云原生市场生态中,也涌现出越来越多的玩家,包括IT咨询和实施服务商、独立软件开发商、集成商、解决方案等供应商,积极推进云原生市场的发展。

从企业的管理者视角,企业的CIO在落地云原生技术时,需要在业务角度考虑如何最大化利用云原生技术实现业务价值,以充分获得业务数字化带来的红利。因此,企业管理者应该从是否满足降本增效的经济回报、是否能承载规模化的业务需求、是否能驱动技术和业务创新等角度,来科学地搭建云原生架构。

为了让企业更好地管理好云原生平台,让新技术安全可靠地走向产业,需要额外关注以下要点:

图:企业管理者视角的云原生架构



综上所述,企业级云原生架构需要更好满足企业管理者的诉求,让客户在享用新技术红利的同时,能保障云端的业务可以安全可靠地规模化开展。

第二章 企业级云原生平台的构成和价值

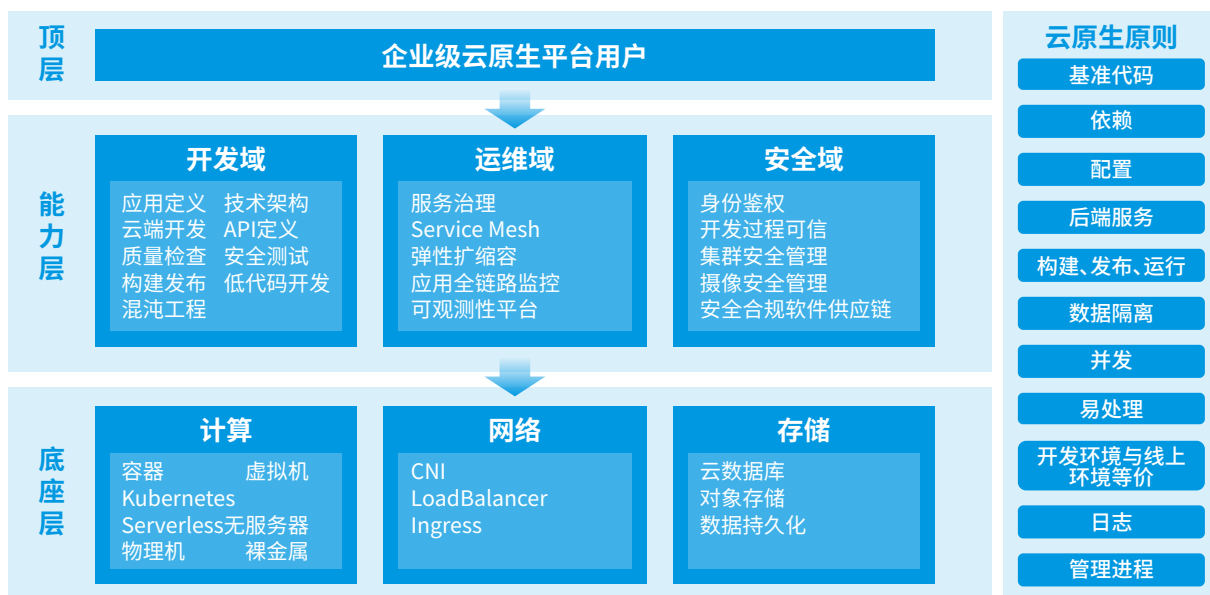
企业级云原生平台架构总览

当前云计算架构无法满足业务规模化上云的架构需求

在线用户规模和时长增加带来对互联网应用服务的更多诉求。传统的技术架构(如单体架构)无法频繁的更新迭代、基础设施模式(如自建数据中心等)使用与维护成本高、开发模式(如瀑布式)偏稳态,传统的开发运维模式已经无法支撑爆炸式的业务增长。

云原生将过去企业IT系统进行了虚拟化改造,企业级云原生平台技术架构具体应包含能力层(开发服务、运维监控、安全治理)、和底座层(计算、存储、网络)两大维度的功能实现,同时将云原生架构一系列原则和特性内嵌到平台架构、技术组件和流程方法中,大幅提升云计算效能。

图:企业级云原生平台架构

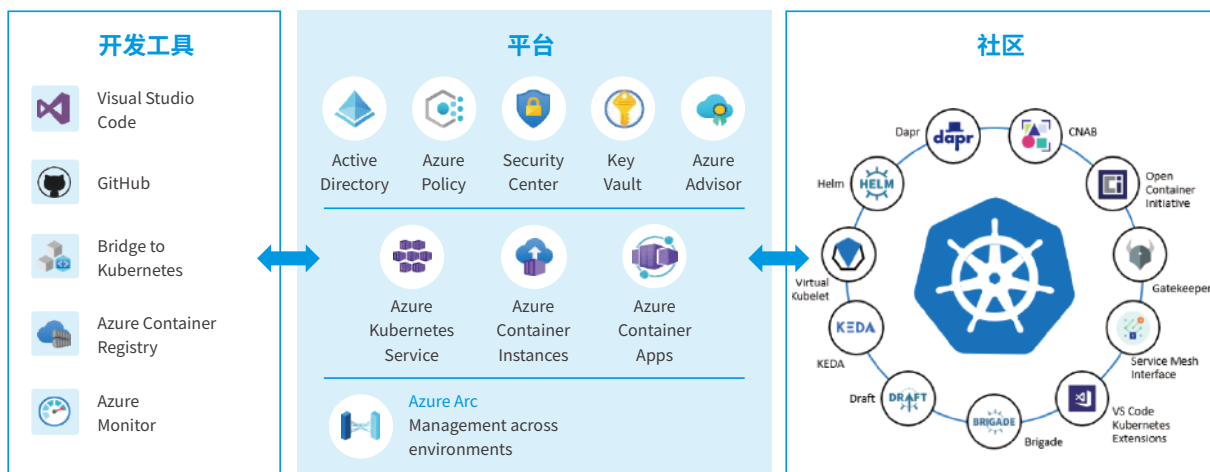


基于企业级云原生架构和能力,企业可以更敏捷、高效、安全应对业务变化,显著降低企业IT开发和运维的成本,提升云计算效能和业务创新速度。

Azure云原生平台 - 立足于可持续软件工程原则链接云原生技术生态

Azure基于面向云时代的原生平台,覆盖从开发到上线运维各个技术领域的工具能力,以及覆盖社区、开发者服务等云原生价值链环节的解决方案。整个生态立足于可持续软件工程原则,来构建和运行可持续软件应用程序所需的一系列核心能力。

图:Azure云原生平台全栈能力



支撑云原生架构特性的关键技术

云原生核心技术赋予了云原生架构新的价值特质

云原生应用通常使用一个或多个源自云的技术架构理念，在Azure的云原生技术框架下，关键技术 in 开发域包括微服务架构、声明式API、分布式运行时等，在运维域包括容器技术、不可变基础设施、GitOps等。

云原生开发域

微服务

微服务架构是为了满足复杂业务架构而设计的面向服务的架构体系，其中应用程序按业务功能被分解为足够小型的、相互间耦合度很低的各种服务，每个服务都可以被视为独立的应用，有独立的开发、测试、数据、部署等。小型、低耦合的微服务架构特性，能提高应用测试迭代的敏捷度、快速响应市场需求的持续创新、不影响整体应用的故障隔离、满足多元扩容和资源利用需求等，极大提高了应用的发布频率，让企业快速为客户提供更可靠的服务。

声明式API

声明式API描述的是目标状态，声明系统要执行的操作，系统将不断向该状态驱动，直到与期望状态相一致。声明式API使系统更加健壮，在分布式系统中，任何组件都可能随时出现故障。当组件恢复时，需要弄清楚要做什么，使用声明式API，组件只需查看API服务器的当前状态，即可确定它需要执行的操作。这种设计理念能减少开发人员工作量，极大提高开发效率。

分布式运行时

分布式运行时 (Distributed Application Runtime) 是微软开源的一个用于构建分布式应用的框架。它提供了分布式应用所依赖的执行环境，将分布式能力进行封装下沉作为运行时以简化分布式应用开发的技术复杂度。分布式运行时包括服务通讯、数据持久化、资源绑定、pub-sub、Actors、密钥等，让用户可以使用任何语言轻松构建与平台无关的微服务应用。

云原生运维域

容器技术

容器被认为是云原生应用的基石。容器通常被定义为“标准化的软件单元”，指运行在容器中的服务和应用可以能被完全地、私有地访问操作系统视图。每个容器可以共享基础主机操作系统，内存和处理器的部分，但彼此隔离。容器的移植性保证了跨云应用的一致性。此外，由于容器占用内存和磁盘空间非常轻量级，因此具备快速启动的特性，适合需要快速扩容的云原生应用。

不可变基础设施

不可变基础设施指服务器在部署后永远不会被修改，只能创建新的进行整体替换。可变基础设施导致配置成本高昂，容易产生常见的系统升级、配置漂移等副作用，在灾难发生时也难以重新构建服务；不可变基础设施保证了基础设施的一致性和可靠性，提升发布效率，方便快速扩展。

GitOps

GitOps是一种持续交付的方式。它的核心思想是将应用系统的声明性基础架构和应用程序存放在Git版本库中。将Git作为交付流水线的核心，每个开发人员都可以提交拉取请求并使用Git来加速和简化Kubernetes的应用程序部署和运维任务。GitOps具有提高生产力和软件质量的巨大潜力，非常适合基于容器化和微服务的开发云原生应用的组织。

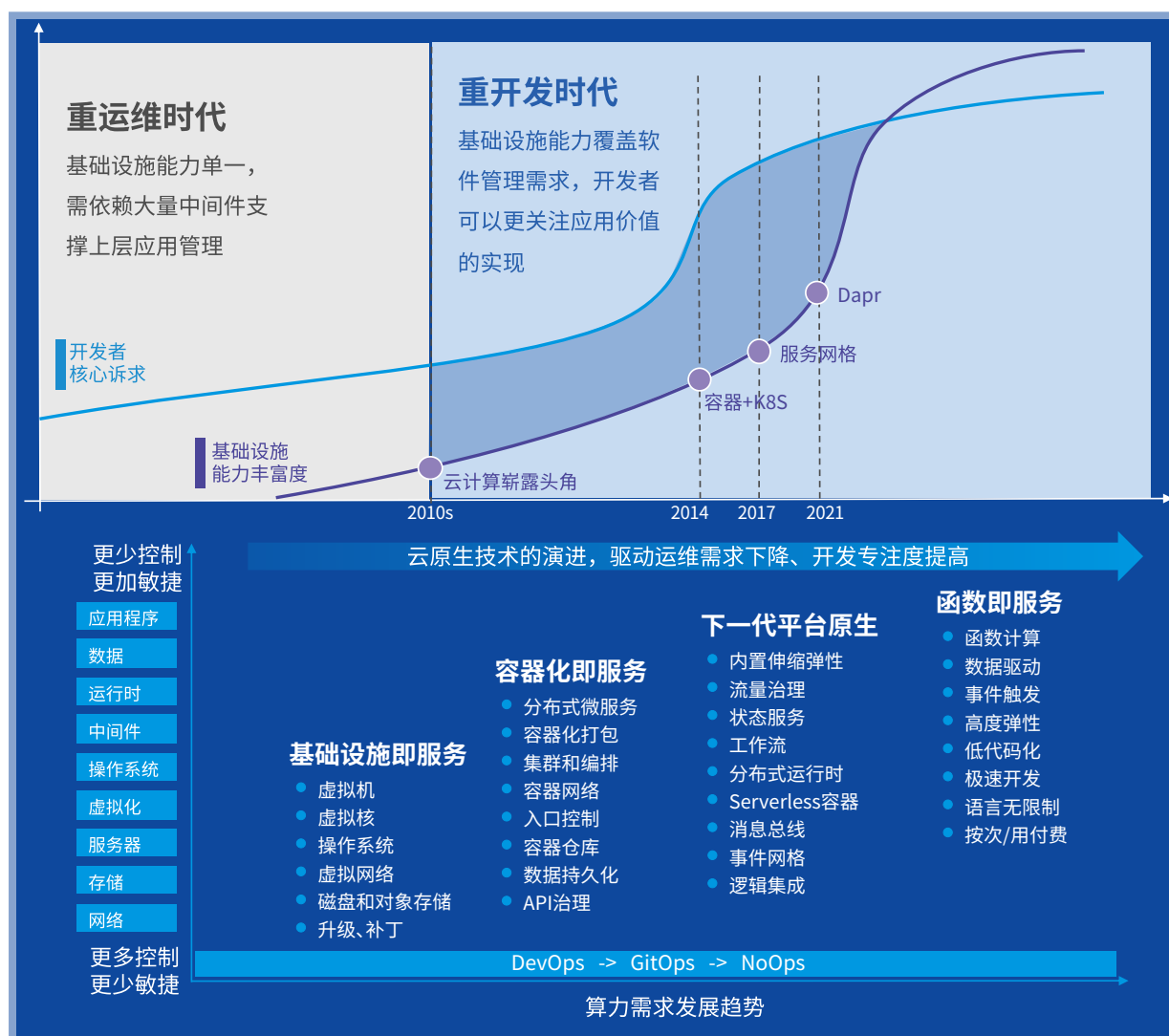
云原生技术的演进趋势：持续收窄基础设施能力和开发者诉求之间的鸿沟

云原生是一套自我演进的理论体系、技术体系和最佳实践的组合，目的是让软件或业务应用能在云上最大化地实现业务价值。云原生技术演进的趋势出现了很多不同阶段，但本质上基础设施能力丰富度与开发者的核心诉求之间的鸿沟趋窄的趋势使然：

- 过去，由于基础设施的能力匮乏，开发者需要依赖传统中间件、微服务框架或者PaaS去调度底层基础设施能力以协助管理软件。
- 而当基础设施的能力不断加强，云的能力已经几乎超出了今天软件架构的大部分需求，具备管理任何一种应用的能力，中间层已不再成为必选项。随着开发者的核心诉求与基础设施能力的差距越来越窄，各种云原生技术涌现，无论是屏蔽底层基础设施能力的容器技术、让应用更平滑进入云存储和网络的Kubernetes技术、还是以云流量控制的方式进行微服务治理的Service Mesh技术，其终极目的都是以更简单、更高效的方式帮助用户的数字化应用更好地使用云服务。

因此云原生技术的演进趋势是持续释放云计算的基础设施能力，让开发者充分享受云时代的红利，推动软件和应用的现代化，让高效开发的数字化应用加速激发业务价值升级。

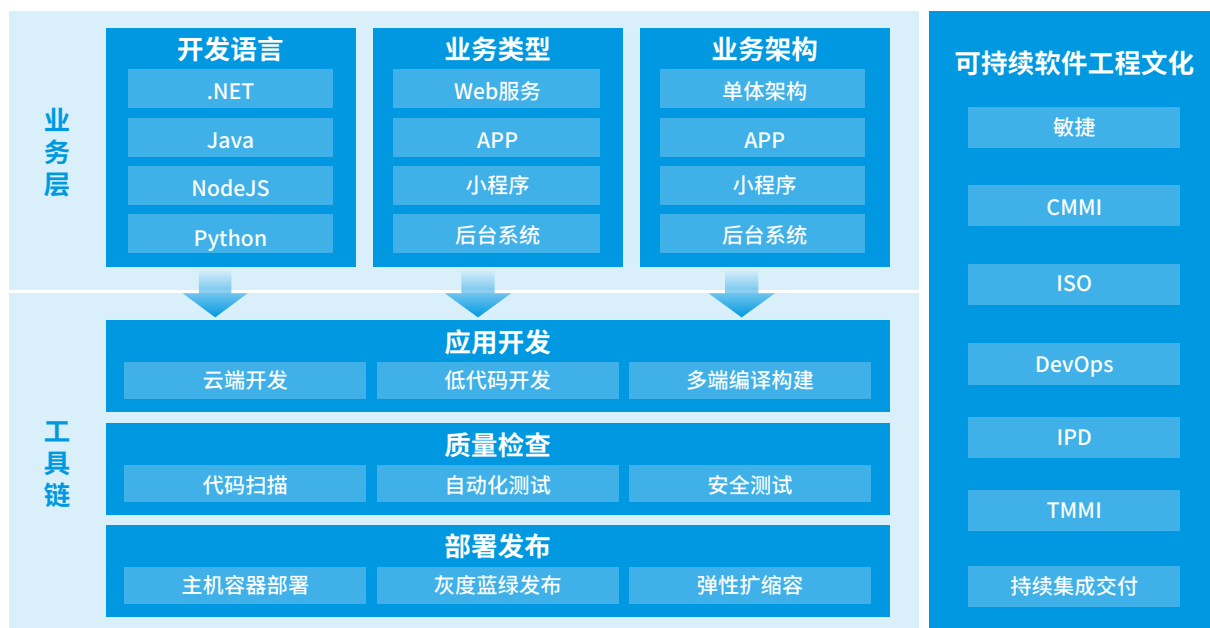
图：云原生技术的演化趋势和底层逻辑



企业级云原生开发 – 立足微服务架构的业务应用开发和治理模式

越来越多的企业以私有云、公有云、混合云进行应用的托管，云正在改变应用程序的设计、开发、运维运营方式。而应用程序的技术架构、开发工具、代码编写、测试、发布、弹性伸缩密切相关，如何设计开发可缩放、安全、可复原、高可用、高扩展的应用程序成为了企业的最大挑战。

图：云端应用开发全流程图



常见的七种应用架构风格

应用程序始于应用架构设计，常见的应用架构风格有七种：单体架构、N层架构、微服务架构、Web-队列-辅助模式、事件驱动的架构、大数据架构、大计算架构。

表：常见的应用架构

Description	架构处理方式		适用场景
	单体架构	把应用程序划分为表现层、业务逻辑层、数据访问层，按层进行依赖关系管理和数据调用	不够灵活、无法适应应用频繁更新，适用于已分层的传统领域
	N层架构	把业务的表现层、业务逻辑层、数据访问层打包在一个工程中，并部署在一台服务器	适合体量较小的服务
	微服务架构	把整个应用程序分为多个逻辑独立的服务，由各个开发团队独立维护	适合业务逻辑复杂的应用程序
	Web-队列-辅助模式	程序包括处理 HTTP 请求的 Web 前端、执行 CPU 密集型任务的后端工作程序，前端通过异步消息队列与后端通信	适合资源密集型任务
	事件驱动架构	使用发布订阅模式，生产者发布事件、消费者订阅事件，互相独立	适合极低延迟处理大量数据的应用程序场景
	大数据架构	把巨型数据集分配多个微数据集进行处理	适合大数据计算任务
	大计算架构	把数据分配到多个计算中心进行处理	适合计算密集型服务

易观分析认为，云原生应用的最佳开发架构是微服务架构。在微服务架构下，业务划分为多个独立的应用，开发者独立进行该应用的开发、测试、部署、发布，每个服务存储自己的数据，并且不需要重新生成和部署整个业务程序，服务与服务之间通过API进行通信。

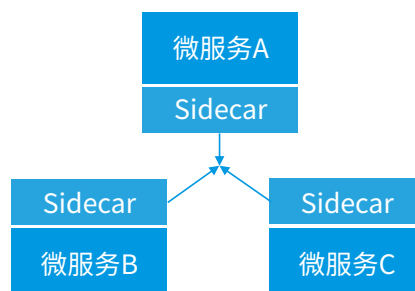
企业可利用Service Mesh和Sidecar来构建微服务治理体系

微服务架构最大化发挥了云原生技术优势,但是也显著提升了系统复杂性,如何有效进行微服务治理成为难点。对此,Service Mesh (服务网格)的价值被凸显,它将复杂性和应用程序分离,通过一个服务代理为应用提供一系列功能,如服务发现、监控、流量控制、日志记录、服务限流和服务熔断等。在服务网格架构中,用户只需要关联业务逻辑,而无需关联服务治理等非业务功能,让开发人员更加专注于业务价值的增长。

实现Service Mesh的最佳模式是 Sidecar。Sidecar是一种分布式架构的设计模式,在不改变主应用的情况下,通过给应用服务加装一个“边车”来辅助主应用做一些基础性、甚至额外工作,达到控制应用逻辑分离目的。

Sidecar模式下,微服务的应用层和基础解耦,提升了应用的可拓展性,降低架构复杂度,优化了微服务治理。

图:利用Sidecar实现微服务之间通信



Dapr提供了Sidecar模式的最佳实践

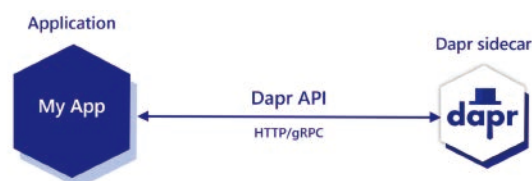
Dapr (Distributed Application Runtime, 分布式应用运行时) 是一个可移植的、事件驱动的运行。Dapr支持多云、边缘、本地等多种环境,在远程开发中,开发可以者利用Dapr在本地模拟出云端开发环境,在本地完成开发和调试,之后将代码移植到云端做进一步调试,极大降低了开发过程本身对环境的依赖性,提升了远程开发云原生应用的效率。

图:Dapr解决方案 - 用于构建跨云和边缘的分布式应用程序的便携式、事件驱动的运行



Dapr以 Sidecar架构的方式公开其HTTP和gRPC API,无论是作为容器还是作为进程,不需要应用代码包含任何Dapr运行时代码。这使得Dapr与其他运行时的集成变得容易,在应用逻辑层面做了隔离处理,提高了可扩展性。

图:Dapr Sidecar架构示意



企业级云原生运维 - 以容器化为中心的主动式应用运维和托管

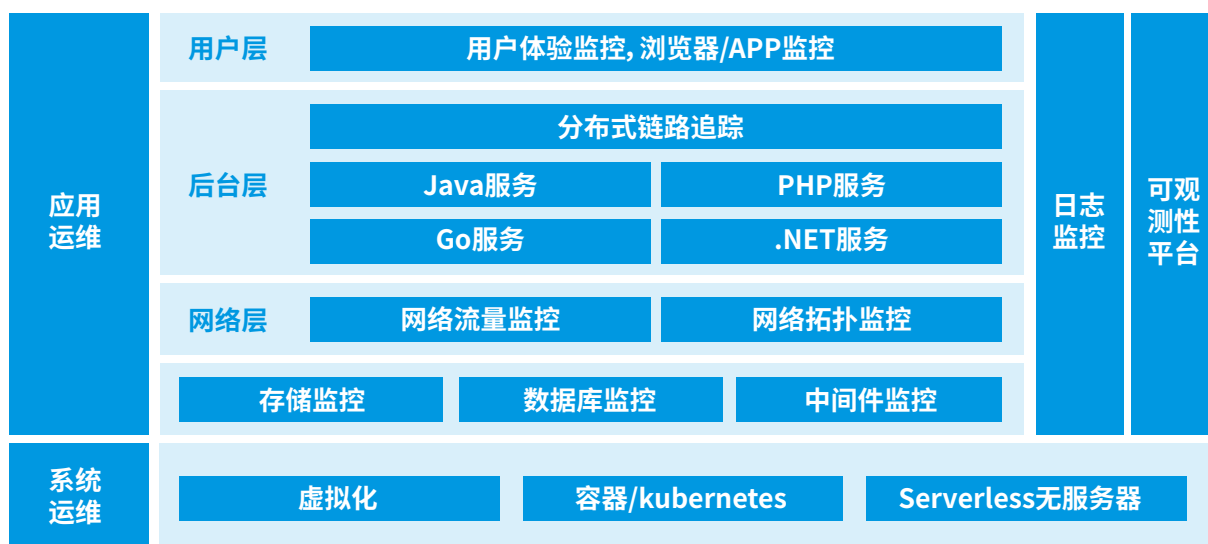
容器正在成为企业对云应用程序执行打包、部署和管理操作的首选方式。借助容器，可以使应用程序可以在几乎任何地方运行，在运行环境中，应用及其相关的依赖被隔离在容器里，互不干扰。而以容器化为中心的主动式运维体系正在成为利用云原生架构和技术的企业所必不可少的能力。

企业的IT运维体系总览

对于大多数企业来说，IT运维体系主要应用运维、系统运维、硬件运维三个层级构成。

- **应用运维层：**从监控、可观测性角度进行应用故障检测、告警、故障自愈，包括用户端、后台、网络、中间件、存储等；
- **系统运维层：**管理平台层，如负责虚拟化、支持开发者进行应用Kubernetes托管、ServiceMesh通信、测试、部署、管理和更新、弹性扩缩容等；
- **硬件运维层：**是运维管理底层，如IDC的物理设备，比如服务器、交换机、路由器、存储设备等。

图：应用运维体系基本结构



云原生时代, 基于Kubernetes的运维存在新挑战

Kubernetes 是一个快速发展的平台,用于管理基于容器的应用程序及其相关网络和存储组件。Kubernetes 重点关注应用程序工作负载,而不是底层基础结构组件。Kubernetes 提供了一种声明性的部署方法,由一组针对管理操作的强大 API 提供支持。

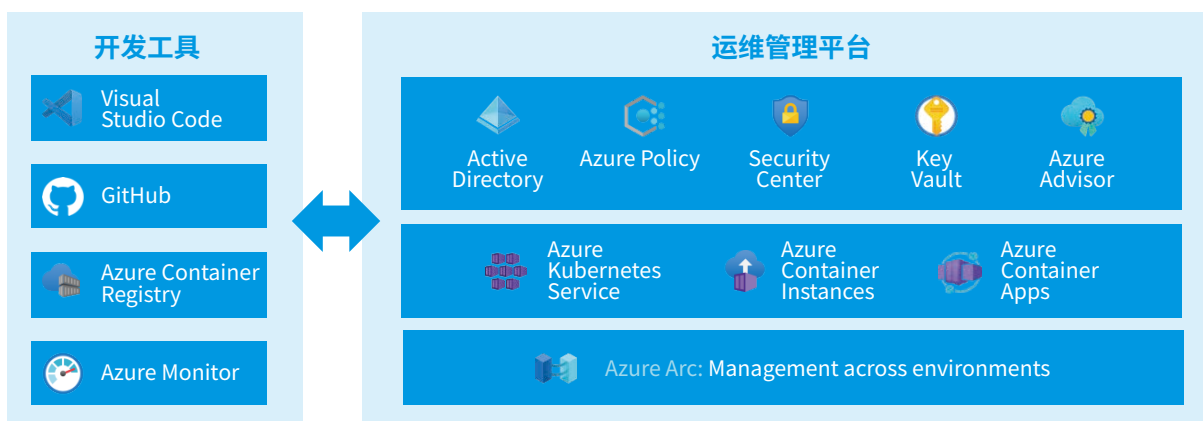
在云原生时代,利用容器化的基础设施构建的新式应用程序(如微服务)越来越多,以Kubernetes为代表用于大规模部署和管理这些容器的容器化基础设施被广泛使用,这让运维有了新的内涵。

但是,应用程序拆分成了微服务后,它们运行在跨多个服务器部署的多个容器中,对于容器的启动、运行、管理、调度、监控变得更加复杂。超过90%的云技术使用者认为Kubernetes导致的系统和组织的复杂性制约了其的云原生技术的使用。

利用云厂商的托管服务和可管理性功能应对Kubernetes运维

基于云厂商的托管服务,用户可以以更低的管理成本、人员投入和可控的技术风险来实现对Kubernetes集群的有效管理。Azure Kubernetes Service提供了完整容器业务可管理性的解决方案,包括跨多个容器的服务发现、自动缩放、协调式应用程序升级、简化部署托管,可观测性等。对于需要将现有应用程序迁移到容器的用户,也可以在 Azure Managed Kubernetes 服务中实现这一目的。

图:Azure应用开发和运维管理平台

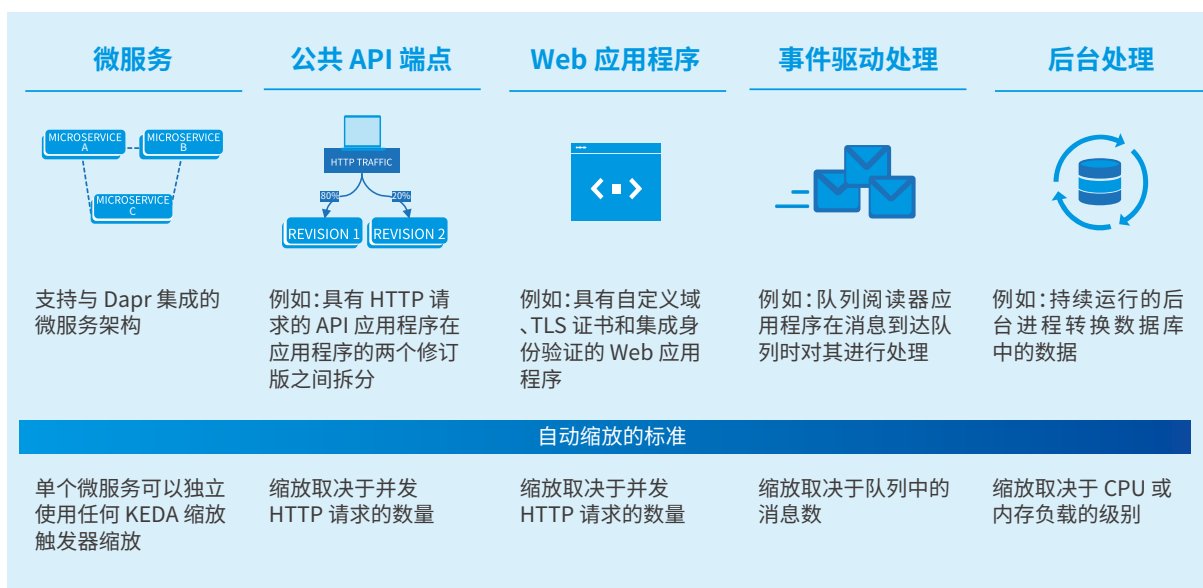


但是,虽然Kubernetes 具备强大的能力,但是它的学习曲线也是令人生畏的。当前业界技术演进的趋势是寻找更接近开发者的认知水平,降低云原生应用开发门槛的方法。

无服务器容器化应用程序托管,让开发者更加专注于应用本身

无服务器技术成为进一步降低运维复杂度的新方式。Azure Container Apps服务可以让用户使用无服务器容器化技术构建完全支持分布式应用程序运行时 (Dapr) 的微服务应用程序。它将中间件能力下沉到了平台,让用户无需负责基础架构,摆脱对Node和Pod运维的局限性,专注于开发应用程序以及特定解决方案,以实现独特的业务价值。

图:Azure Container Apps的各类解决方案



可观测性是构建主动式云原生运维能力体系的不可或缺环节

在云原生时代,基础设施与应用的部署构建都发生了极大变化,传统的监控方式已经无法适应云原生的场景。可观测性成为云原生平台必不可少的一环,能否构建可观测性能力关系到云原生技术实践能否在生产环境中落地。

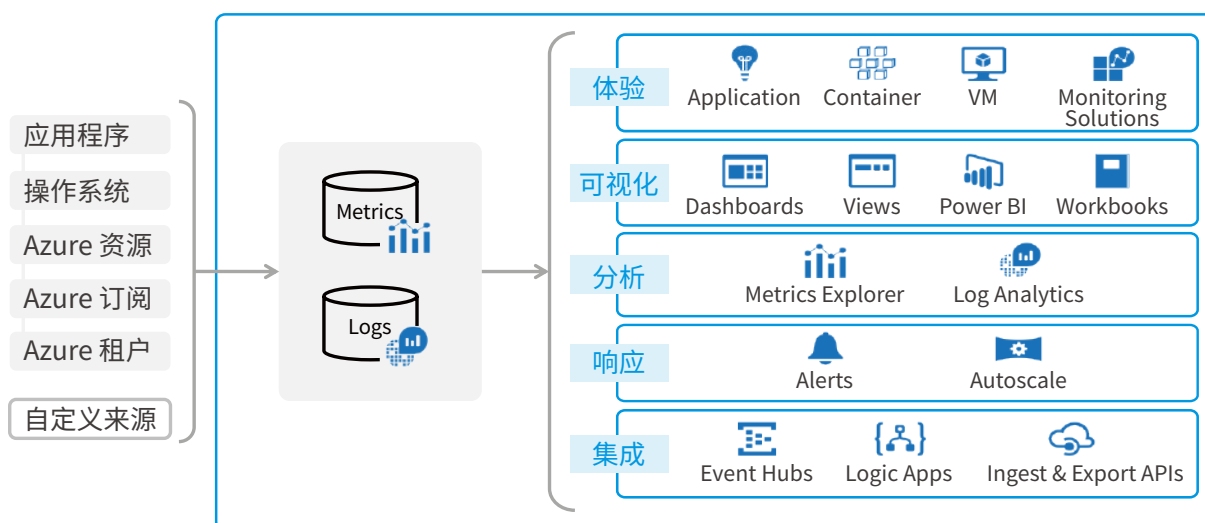
图:传统监控方式和可观测性的差异

对比维度	传统监控方式	可观测性
监控对象	监控资源,如基础设施	监控全局,包括基础设施、中间件和调用链条等
监控重点	问题发现	问题发现和归因
监控视角	黑盒化,在外部看系统状态	白盒化,由内而外看系统状态
数据类型	指标数据	指标、日志、链路
面向用户	运维人员为主	全生命周期开发人员和运维人员

Azure Monitor – 提供云原生时代的多层可监控性能力

Azure Monitor是Azure统一的监控服务,它提供了一个全面的解决方案,用于从云和本地环境收集、分析和处理遥测数据。此信息可以帮助用户以可视化的方式了解应用程序的性能,并主动识别影响应用程序及其所依赖资源的问题,找到解决办法。

图:Azure Monitor 多层可观测性概要



在上述 Azure Monitor 的全景中,

- **中心是用于 Metrics (指标) 和 Logs (日志) 的数据存储。** Metrics 是数字值,用于描述系统某些方面在特定时间点的情况。Logs 包含不同类型的已经整理成记录的数据,每种类型都有不同的属性集。
- **左侧是填充这些数据存储的监视数据源。**数据源涉及应用程序、操作系统和它所依赖的服务等。同时,还可以使用数据收集器 API 从任何 REST 客户端收集日志数据。
- **右侧是 Azure Monitor 针对这些收集的数据执行的不同功能。**其中包括分析、警报和流式传输到外部系统等操作。

同时,Azure Monitor 的多层可检测性能力内嵌在了数据可视化平台 Azure Managed Grafana 中,可它支持用户实时查看和分析应用程序和基础结构遥测数据。

企业级云原生安全 - 与数字化业务管理和流程耦合的软件合规生产供应链

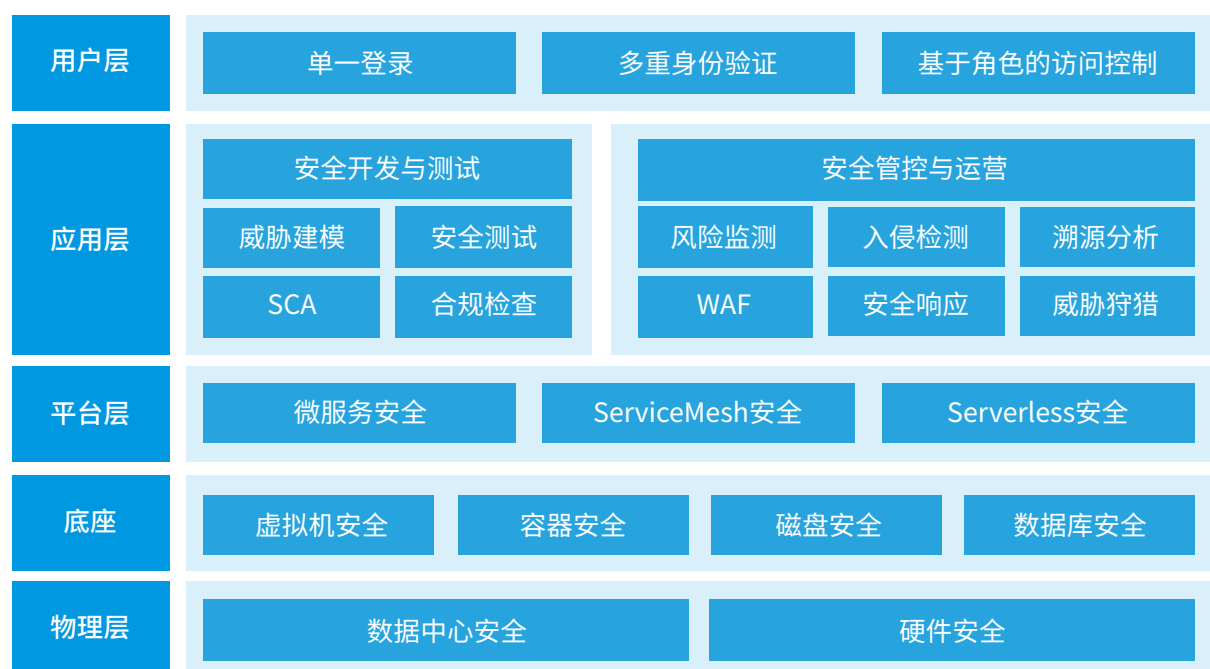
世界对技术的依赖正在快速增长,但技术带来的安全风险也在极具增加,尤其对于金融、零售、制造等行业的企业级用户,安全的重要性正在显著提升。对于企业级云原生平台来说,必须为平台上的开发人员配备企业级工具,以便在问题导致安全漏洞之前及时识别和修复问题,保障企业云端业务的安全运转。

云原生安全包括应用的使用过程安全、开发过程安全、运维过程安全等,即应用软件生产供应链的每一个节点都做到安全可控,尽可能的减少安全攻击,当安全攻击发生时及时发现和响应,减少有价值的信息暴露。

保障云原生软件安全,应着眼于构筑软件合规生产供应链

完整的企业级云原生安全架构覆盖了应用全生命周期,包括用户使用层、应用开发与运维层、底座层、物理层的维护,用户的使用过程安全包括单点登录、多重身份验证、基于角色控制权限等;应用开发过程的安全性包括进行安全性扫描、漏洞敏感信息检测、合规性检查;应用运维过程安全包括通信安全、虚拟机安全、容器安全、存储安全、网络安全、数据中心安全等。

图:云原生安全架构



技术在不断的发展更新,随着网络攻防两端的手段变化,对应的应对策略也会变化,但不变的是软件安全合规供应链的核心原则。在云原生架构下,企业业务上云也扩大了企业IT架构的可攻击面、同时也对整个软件生产流程的规范性、合规性带来了巨大挑战。

易观分析认为,保障云原生软件安全,应当从应用全生命周期视角出发,着眼于构筑与数字化业务管理和流程耦合的、涵盖软件供应链安全管理、技术、工具等各个方面的软件合规生产供应链。

对云原生安全架构分层加固,是构筑企业级软件合规生产供应链的关键

企业可以在用户层、应用层、底层分别实施加固,实现对整个链条安全性的加强,通过在各个层级阻止可疑入侵、及时发现和修复安全漏洞来提升整体研发环境的安全水平。

图:企业级云原生平台的分层安全加固方案



Azure利用多层加固的方式,提供软件合规生产供应链的解决方案

Azure云原生平台内嵌了软件合规生产供应链的内在逻辑,通过多层加固的方式来对链条上每一个环节进行有效的管理和控制。

图:Azure基于企业级云原生安全架构的分层安全加固



云原生底座能力之计算 – 实现算力需求与算力资源的最优匹配

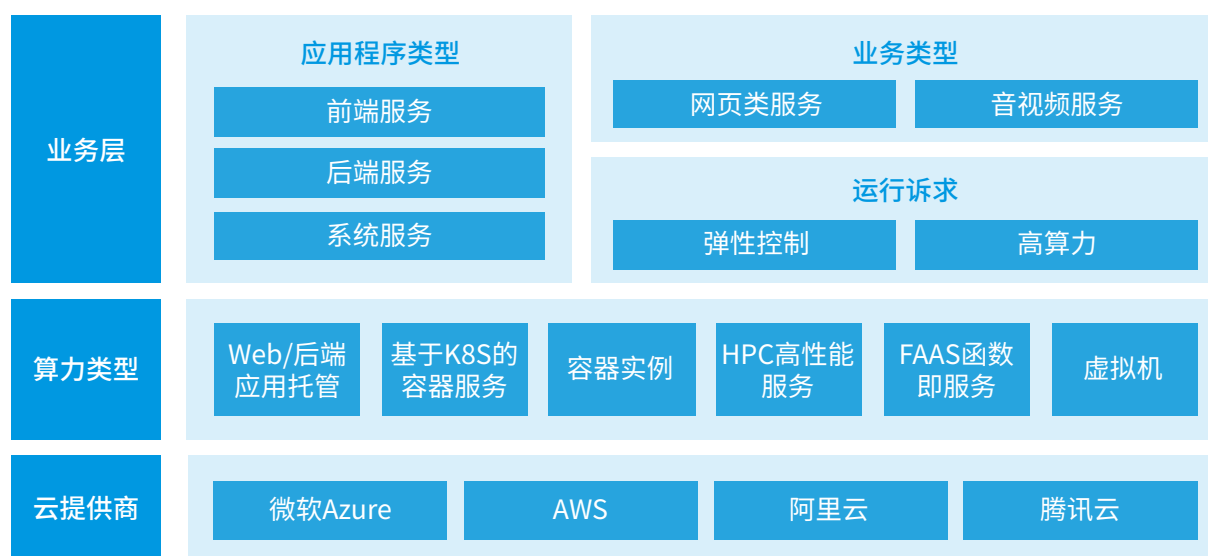
企业数字化转型过程中，以公有云、私有云、混合云等方式逐步上云，算力模型从以服务器为核心已变为以资源和应用为核心。

这对算力提出了更高要求，体现在如何让弹性伸缩的资源对各类应用运行提供良好支持、如何对多云、混合云提供良好支持、如何对应用开发全流程的可靠性、安全性进行有效管理。因此算力的概念不再是狭义的物理机、虚拟机等服务资源的概念，而是广义的不同应用运行起来所需计算资源的托管模型。

企业业务云化的复杂性提升了算力架构和管理调度的难度

- 不同的企业应用上云方式对算力的要求差异。一般来说可分为三类情况，分别是无需改造架构和代码上云、改造应用和技术架构上云、直接基于云进行服务开发。
- 不同的业务价值意味着对算力需求的差异，包括特定应用程序类型（不同的开发语言、技术框架等）、业务类型（业务特性、技术处理流程等）、运行诉求（高频、低频、实时等）都决定了算力的选择。

图：云原生算力能力服务全景



云原生时代的算力管理目标是实现算力需求与算力资源的最优匹配

在云原生平台的支持下，灵活的算力获取可以满足不同类型企业的业务上云、应用弹性扩缩容、按需付费等个性化业务诉求，最大化降低了企业云端业务使用算力的成本。具体来说，企业可以从技术架构和运营管理两个维度实现算力需求与算力资源的最优匹配。

- 在技术架构方面，充分依托弹性容器实例、智能监控、Service Mesh等云平台所提供的云原生技术产品，实现算力供需的智能管理，达到算力需求与资源更好的匹配。
- 在运营管理方面，企业可以建立算力治理专项组织，综合业务和技术视角开展成本和速度相关的优化、运作业务项目预算审批流程，在全局维度实现高效的算力取用。

Azure云原生基础设施生态 – 服务不同对象、语言和生态的虚拟化基础设施支撑

Azure支持用户按需使用、灵活通用的云计算能力。在资源调度方面,用户可以按需缩放资源使用,仅为使用的资源付费。在资源通用性层面,Azure的产品生态也提供从容器、微服务到Serverless、事件驱动计算的不同基础设施抽象程度的计算方案。

云原生计算能力被深入内嵌在Azure云原生产品生态中。对于开发人员和运维人员来说,无论更加看重基础设施的调用的灵活性,还是更看重应用开发的易用性,都可以根据自身开发和运维工作的特点和需求在Azure生态产品的使用中获得更优的计算解决方案,即在开发工作中实现多语言开发、容器化部署和集成,在运维工作中更好地实现工作负载管理、灵活缩放、灵活托管等。

图:Azure云原生产品生态对开发运维人员提供更灵活的使用体验

专注于基础设施 更高的灵活性					基础设施抽象 专注于应用			
	 Azure Container Instances	 Azure Kubernetes Service	 Azure Red Hat OpenShift	 Azure Container Apps	 Azure Spring Cloud	 Azure App Service	 Azure Static Web Apps	 Azure Functions
优化对象	单一容器	容器化的应用、API、微服务			Spring Boot 应用	基于Windows and Linux的三层结构应用	Web前端应用程序	Event-driven code
支持语言	任何语言和框架				Java	.NET, Java, Python, Node.js, PHP, Ruby	Angular, React, Svelte, Vue, Blazor	.NET, C#, Node.js, TypeScript, Java, Python, PowerShell
支持生态		CNCF	Red Hat		Spring, VMware	Microsoft		Microsoft
开发人员体验	部署容器后的开发人员体验最少。不支持服务发现、多个微服务等。	- 使用 K8s YAML 文件部署容器；使用部署中心与 GitHub Actions 集成。 - 使用开源项目和内置托管项目，例如 KEDA、Dapr 和 ML。	- 使用 Red Hat 通用基础镜像部署容器化应用。 - 使用 S2I 引入应用程序代码。 - 使用内置 CI/CD 利用受支持的 Red Hat 组件	- 部署容器化应用程序或与 GitHub Actions 集成。 - 使用内置的 Dapr 进行微服务通信、发布/订阅等。 - 依靠内置的应用程序生命周期管理功能，例如修订和流量转移。	- 使用 Maven 构建 Spring Boot 应用程序，并部署 JAR 工件。 - 与 GitHub Actions 集成，让CI/CD 流程自动化。	- 将 Web 应用程序代码、与 Git 集成或部署容器映像。 - 与Visual Studio 和 GitHub 的紧密集成。	- 一键构建一个包含所有流行 JavaScript 框架的网站。 - 使用 Azure Functions 或 Blazor 创建后端应用程序，在发布之前生成暂存版本以快速预览。	- 使用 Azure Functions SDK 或 Azure Functions 基础运行时映像构建事件驱动函数。 - 支持来自不同事件源的触发函数。
运维人员体验	使用 Azure API/门户 部署 单个容器。不支持容器缩放。	- 使用 K8s原生 API/Azure 门户 创建和管理工作负载。 - 使用集群自动扩缩器配置扩展，使用虚拟节点到 ACI 进行突增。 - 使用 Azure Policy 应用治理策略并与Azure 服务集成。	- 使用OpenShift 原生API和控制台创建和管理工作负载。 - 使用 Operator Hub 启用其他功能；使用机器集缩放器配置缩放。	- 使用 Azure APIs/portal 创建托管多个应用程序的容器应用程序环境。 - 使用支持归零的 KEDA 定义缩放参数。	- 使用 Azure Log Analytics 和 Application Insights (或第 3 方 APM) 集成日志记录和监视。 - 可基于计划和应用程序指标自动缩放；支持绿/蓝部署策略。	- 简单的运营模式和完全托管的体验。 - 通过使用部署槽 减少停机时间并最大限度地降低风险。	- 自定义域管理和免费 SSL 证书自动续订以获得最佳安全覆盖。 - 与 GitHub、AAD 和 Twitter 帐户的身份验证集成。	如果需要，可以使用 Azure Arc 在 Kubernetes 上部署 Azure Functions。

云原生底座能力之存储 - 运用符合云原生特性的存储方案和数据库

在云原生时代，云存储能力是重要的基础设施底座。云计算供应商将海量的存储资源池化，企业或个人用户按需购买存储资源，快速、低成本搭建云端存储解决方案。

实现云原生应用的特性，要求更先进的数据存储能力支持

云原生应用特点是容器化、服务网格化、运行时弹性、DevOps、故障自愈等。想要实现这些特性，对存储能力也提出了多方面的新挑战，云原生数据库的核心是要设计一种可弹性扩缩容、充分利用云平台池化资源、适配云平台已有的基础设置的数据库架构，这也对存储能力提出了诸多新要求。

图：云原生应用特性对存储能力的新要求

易用性	高性能	高可用
微服务架构、容器部署使得存储服务部署、运维复杂，与主流编排平台没有整合。	规模化增长的用户带来大量应用 IO 访问，低时延、性能成为应用运行效率瓶颈。	云原生存储需要高可靠/高可用，不能出现单点故障。
业务敏捷性	持久化存储支持	
DevOps成为主流研发流程，因此存储需要敏捷性，能快速应对业务变化。	在云原生存储中，对于有状态应用，所有数据需要持久化存储。	
多云API	兼容性	Azure Kubernetes Service
云服务通过 API 进行通信。不同的规则结构或不同的语言。这些差异需要应用程序自定义以实现跨服务通信。	集成到单一环境中存储服务需要跨云兼容。服务需要适应相同的数据结构。	多云混合使用，对于跨云服务和环境的可见性很难保障，需要集中监控和联合服务。

云原生存储方案的选择面大，大中型企业应关注存储端的企业级特性

云上存储所带来的便利性、低成本性、可扩展性也成为企业数字化转型的关键能力支撑，而伴随着云计算厂商对于IaaS层存储能力的竞争，云上存储的价格会越来越低，企业对于云厂商的存储解决方案的选择也有了更大的空间。

易观分析建议，大中型企业应当更关注那些提供企业级存储解决方案能力的厂商和产品，其典型企业级特性包括存储高度可用、安全、持久、可扩展且冗余等。同时，正在推动数字化转型的技术用户还应当关注云厂商提供的不同形态云架构的数据存储和管理能力，比如Kubernetes 以卷为首选机制来管理容器生成的需持久化的文件，用户可利用厂商提供的云盘、SSD等在容器中构建卷，卷亦可用于在同一个 Pod 中多个容器之间共享文件。

数据库是数据存储解决方案的主要载体

无论是在信息化时代还是云时代,数据库都是最常用的数据存储软件。数据库通常能提供完备的数据存储管理能力,以及一部分计算能力。常用的三种数据库为MySQL, Redis和MongoDB,他们之间的特性和用途各有差异。

图:三种常用数据库类型的对比

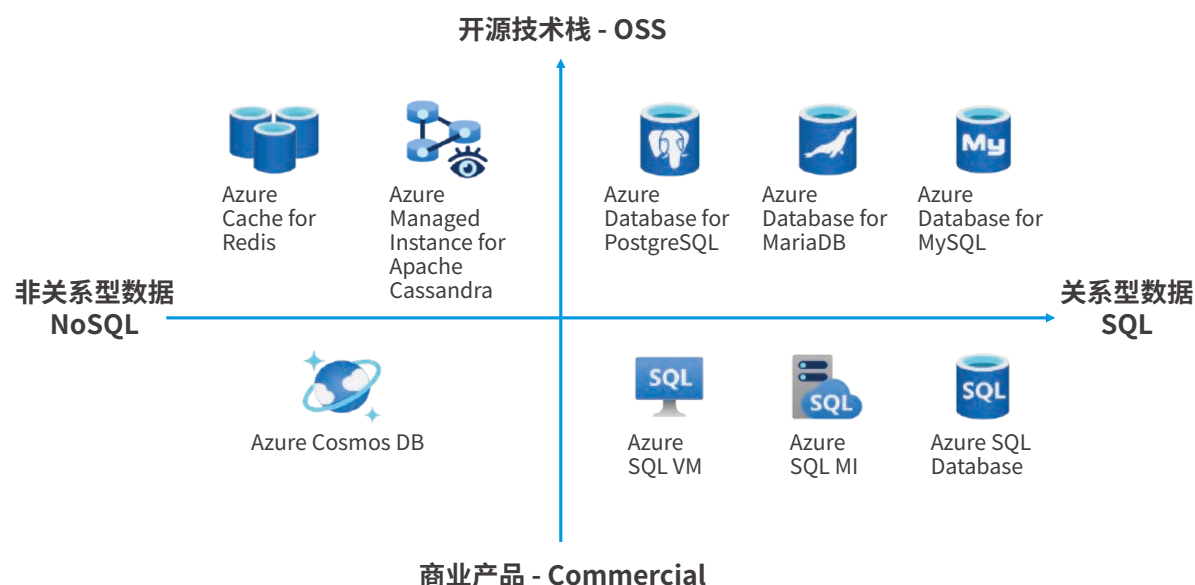
	MySQL	redis	mongoDB
基本描述	最流行的关系型数据库管理系统	key-value 存储系统,是跨平台的非关系型数据库	基于分布式文件存储的数据库。由C++语言编写
存储数据类型	持久化数据	读取频繁但无需长期保存的数据	持久化数据
存储位置	数据存储在磁盘	数据存储在磁盘	存储在硬盘上的,只不过需要经常读取的数据会被加载到内存中
数据库分类	关系型数据库	非关系型数据库	介于关系数据库和非关系数据库之间
读写速度	读写速度慢	读写速度快	读写速度快
缺陷	在海量数据处理的时候效率会显著变慢	短时间内大量增加数据,可能导致内存不够用	不支持事务操作,消耗内存大

Azure存储平台 - 适用于现代数据存储方案的云存储解决方案

Azure 存储平台是 Microsoft 提供的适用于现代数据存储方案的云存储解决方案,为云中的各种数据对象提供具有高可用性、可大规模缩放、持久且安全的存储。

Azure 提供了完全托管的关系数据库、NoSQL 数据库和内存中数据库,覆盖面向现代应用开发人员的专有引擎和开放源代码引擎。基础结构管理(包括可伸缩性、可用性和安全性)为自动执行,可节省时间和资金,专注于构建应用程序。用户可根据不同类型的数据存储需求选择对应的云数据库产品和服务。

图:基于Azure 的云数据库产品分类和技术应用特性



云原生底座能力之网络 - 保障企业级云原生架构的高可用性

在基于容器的微服务应用程序开发方法中,应用程序组件会利用网络协同工作以处理其任务。对于企业级云原生平台用户来说,除了已具备的网络资源共享、弹性伸缩、按需付费外,对于网络的高可用,网络与Kubernetes和容器生态的良好结合、以及网络安全等能力也存在更多的诉求。Kubernetes提供了各种网络资源来促成这种协作。

Azure Kubernetes两种网络模型比较

Azure Kubernetes 兼容Kubenet (基本网络) 和 Azure CNI (高级网络)。虽然Kubenet 和 Azure CNI 都为 AKS 群集提供网络连接,不过他们各有优缺点:

图:Kubernet与Azure CNI比较

	Kubenet	Azure CNI
在现有或新的虚拟网络中部署群集	支持 - 手动应用 UDR	支持
Pod-Pod 连接	支持	支持
Pod-VM 连接;VM 位于同一虚拟网络中	由 Pod 发起时可正常工作	采用两种工作方式
Pod-VM 连接;VM 位于对等互连的虚拟网络中	由 Pod 发起时可正常工作	采用两种工作方式
使用 VPN 或 Express Route 进行本地访问	由 Pod 发起时可正常工作	采用两种工作方式
访问服务终结点保护的资源	支持	支持
使用负载均衡器服务、应用程序网关或入口控制器公开 Kubernetes 服务	支持	支持
默认的 Azure DNS 和专用区域	支持	支持
对 Windows 节点池的支持	不支持	支持

云原生网络关键能力 - 负载均衡

“负载均衡”是指在一组后端资源或服务器之间均衡地分配负载,通过分配网络流量提高应用程序的可用性和可伸缩性,是云原生架构下高可用的网络基础架构的关键组件。

Azure 提供可用于跨多个计算资源分配工作负荷的负载均衡服务。按照功能,Azure负载均衡机制可以分为三种:Azure Load Balancer, Application Gateway , Traffic Manager

图:Azure的三种负载均衡机制对比

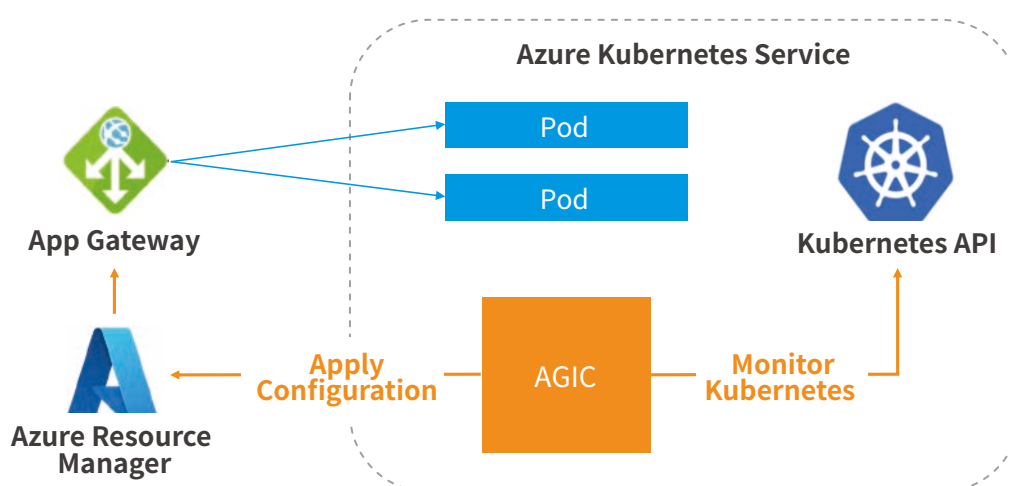
	Load Balance	Application Gateway	Traffic Manager
简介	提供OSI Layer 4的负载均衡器	一种应用程序层负载均衡	针对DNS的负载均衡
所在网络层	传输层 (L4)	应用层 (L7)	DNS
协议支持	任何协议	HTTP / HTTPS	任何协议
虚拟网络支持	同时支持公网和内网的负载均衡	同时支持公网和内网的负载均衡	只支持公网负载均衡
服务端点	Azure虚拟机和Cloud Service Role	任意的内网IP地址或公网IP地址	Azure虚拟机, Cloud Service, Azure Web App或其它外部服务端点

将负载均衡器和Kubernetes Service有效结合,可利用Azure应用程序网关入口控制器

Ingress 是对集群中服务的外部访问进行管理的 API 对象,用于实现用域名的方式访问k8s内部应用。Ingress 可以提供负载均衡、SSL 终结和基于名称的虚拟托管。

为了让 Ingress 资源工作,集群必须有一个正在运行的 Ingress 控制器。Azure应用程序网关入口控制器 (AGIC) 是一个是一个配置 Azure 应用程序网关 的 Ingress 控制器,让用户可以利用 Azure 的本机应用程序网关 L7 负载均衡器向网络公开云软件。AGIC 监视托管时所在的 K8s群集并持续更新应用程序网关,将前端负载均衡器和k8s有效做了结合。

图:Azure应用程序网关入口控制器 (AGIC)



在 Azure Kubernetes 服务中, 用户可使用 Network Policy 来管控 Pod 之间的流量

默认情况下,AKS 群集中的所有 Pod 可以无限制地发送和接收流量。为了提高安全性,用户在 Kubernetes 中运行最新的基于微服务的应用程序时,可定义用来控制流量流的规则以控制哪些组件可以相互通信。

Network Policy (网络策略) 是一种以应用为中心的结构,在 Kubernetes 中,使用 Network Policy 可在群集中定义用于 Pod 之间的入口和出口流量的规则。Azure Policy 通过内置策略与 AKS 集成,以集中、一致的方式在群集上应用大规模强制措施和保护措施。除了兼容自身的网络策略, Azure 还兼容 Calico 网络策略 - 由 Tigera 建立的开源网络和网络安全解决方案。甚至对于企业级用户来说,还可以按照公司特定的安全性和合规性要求创建自定义的 Policy。用户可以在创建 AKS 群集时选择一个网络策略选项。但创建群集后无法更改策略选项。

若要使用 Azure Network Policy,必须使用 Azure CNI 插件并定义自己的虚拟网络和子网。Calico 网络策略可与此 Azure CNI 插件配合使用,也可与 Kubelet CNI 插件配合使用。

企业级云原生架构顶层设计:搭建登录区,完成云原生架构体系落地

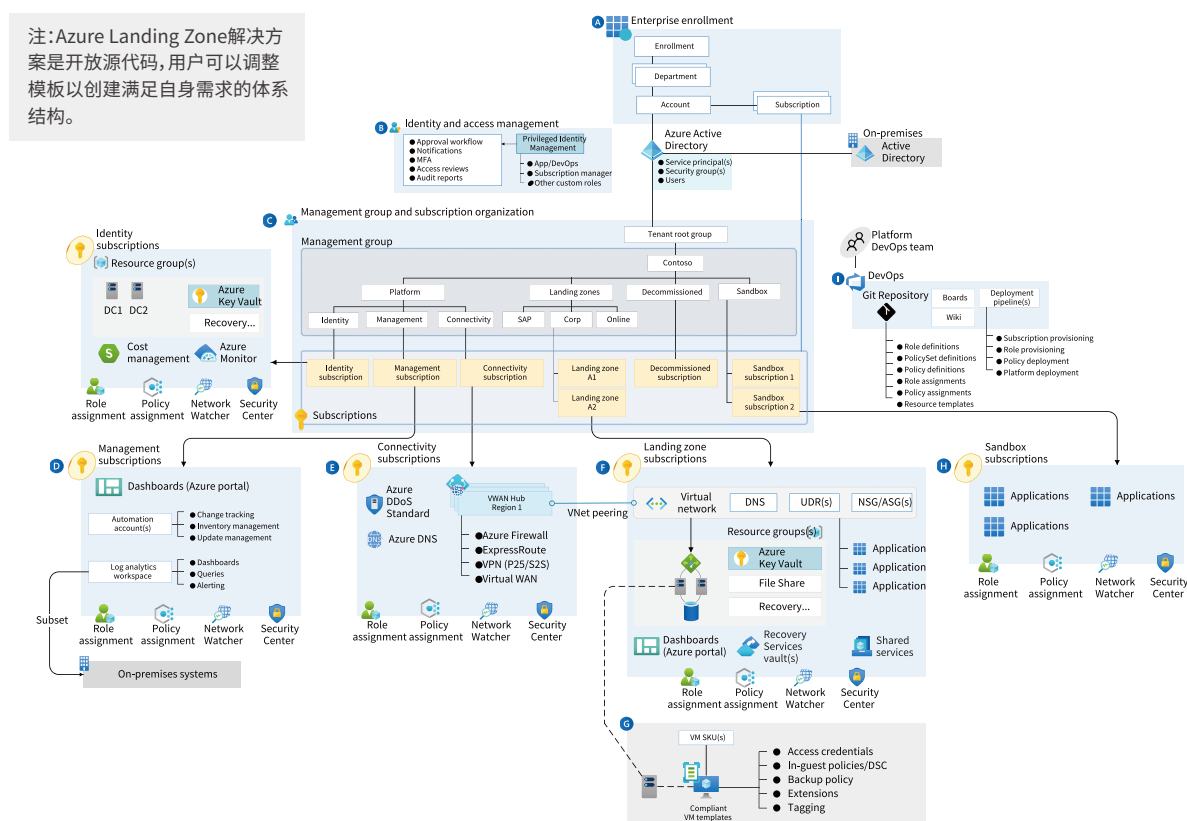
虽然云原生平台的能力和技术特性可以极大地帮助企业在数字化转型的过程中构建起自身的业务创新优势,但企业级用户在导入云原生平台仍是一个系统性的复杂工程,需要考虑诸多的因素,比如标识、管理访问、网络结构、资源的组织、管理性、自动化、可伸缩,成本的管理、安全合规等。

易观分析认为,对于企业级用户来说,如何基于一个完整的顶层设计和方法论将这一系列因素整合起来,为后续的业务上云扫清障碍,让企业真正实现云原生技术的落地,成为云原生厂商在服务客户数字化转型的过程中需要解决的关键命题。

Azure Landing Zone是一套为应用缩放、安全管理、网络访问和身份标识服务的基础架构资源框架。作为一种成熟的横向扩展目标体系结构,Azure Landing Zone帮助组织运营成功的云环境,在遵循安全和治理方面最佳做法的同时,推动企业数字化转型和业务发展。这一体系结构基于已采用 Azure 治理数字资产的客户所获得的大量经验及其反馈而形成,体现了引入云原生技术的企业级最佳实践。

下图为企业级Landing Zone解决方案架构及其八大关键设计领域,这个体系有助于将客户需求转化为Microsoft Azure 架构和功能,并解决本地和云基础设施之间的不匹配问题。

图:企业级Landing Zone解决方案架构及其八大关键设计领域



只有基于企业级Landing Zone体系,方能构建起企业级的云上环境,实现和业务发展曲线相同的基础架构支撑,将云上资源治理妥当。通过Azure Landing Zone,用户可在 Azure 中实现企业级的应用程序迁移、现代化和创新。

第三章 企业级云原生平台案例

大型股份制银行 - 零售业务基于云原生平台全面加速数字化进程

该企业名列全球银行前20强,传统银行业务是其增长支柱。但在金融业务增长趋缓、监管趋严的压力下,公司为了保持稳健增长,希望借助数字化手段加速零售业务,打开多元创收空间。

■ 转型背景

为了快速开展非银零售业务,公司组建了非银团队,并希望以此作为公司数字化转型的先驱。通过打造数字化的零售平台,吸引大量的零售商家入驻,同时激活更多的零售用户,以此盘活银行的零售业务。

公司有自主研发的IT系统,但在开展零售业务后,面临着业务规模参差不齐的问题,比如零售业务往往具备周期性特点,面临每年的双十一、黑五等大促活动、一天不同时间段业务高峰期分布不齐的问题,基于虚拟机按照集群要求进行扩充的传统方式难以满足应用的快速负载均衡需求。

同时,金融行业属性的企业在上云进程中,额外关注技术自主可控,因此在选择上云服务时更注重容器等与云厂商弱绑定的技术应用。

■ 转型思路及方案

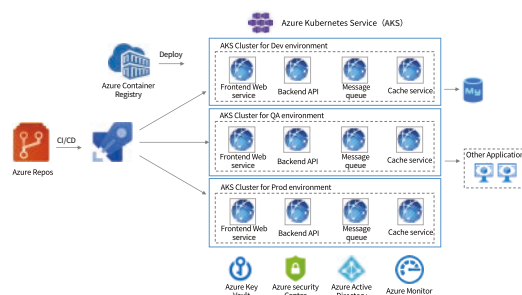
为了适应零售业务的特点,需要采用容器、K8s等更弹性的计算及存储技术对IT架构进行弹性扩充。

2018年开始,公司与微软进行深度合作来共同驱动数字化转型进程。当确定非银业务作为试点项目后,利用Azure云原生技术将其业务和技术架构进行改造升级,包括 Azure Kubernetes

托管服务、Redis缓存、MySQL数据库等核心技术的应用,有效满足其对于业务弹性扩容的需求。

为了让业务可持续,加强云原生技术能力的安全合规性,微软帮助公司建立和完善数据中心的容灾隔离、双活、热备冷备等机制,让业务的开展更安全稳定。

图:该银行客户基于Azure的数据系统架构



■ 转型价值及收益

- 1. 零售业务数字化大跃迁:**云原生技术的应用,历经四年帮助公司的零售业务实现了从畅想到落地,到2021年底,其线上零售平台注册用户数突破2000万,成交额近1亿元,开辟了互联网零售业务的增长曲线。
- 2. 反哺银行金融业务:**零售业务的数字化给公司积累了大量的用户数据,让用户画像更精准,通过数据的精细化运营,给主营的金融业务带来巨大的价值。
- 3. 驱动创新业务的深化发展:**公司与微软将进一步探索金融科技赋能的业务创新实践,会开展更多如AI、IoT、区块链等前沿技术及产品与云服务协作的验证及落地,赋能公司业务的研究。

为什么选择微软Azure云原生平台

四年的合作过程中,微软扮演着值得信赖的良师益友角色,基于积累的数字化转型经验,和该公司一起探索业务转型的路径,全程参与了从最开始探索确定非银业务是合适的试点项目,到业务架构的上线及深化改造,再到安全合规性的加强,帮助公司真正实现了业务模式的革新:

- 1. 中立性:**微软更多承担着咨询顾问与实施的角色,不会让客户产生核心数据或商业模式被窃取的顾虑,能从专业的云原生平台的角度给予客户值得信赖的业务及技术改造的指导及建议。
- 2. 开放性:**生态上,微软有大量的生态伙伴,一方面可以给客户引荐合适的技术厂商,通过Hackathon、技术创新大会等形式给客户打开微软的生态窗口,另一方面客户也可以加入生态成为生产力;开源上,微软提供企业级的GitHub开源社区,给客户公开开源产品间的依赖关系、版本更新、安全漏洞的透明信息,实现可靠开源;技术迁移上,微软会开放如DevOps平台的核心技术及功能,并帮助客户真正消纳为其自主可控的技术。
- 3. 满足业务扩张需求:**微软作为国际技术厂商,给客户的业务出海也提供了诸多便利性,中国零售业务的成功得到验证后,公司将考虑与微软进一步合作拓展海外零售业务。

领先智能制造企业 - 利用Azure云原生平台为海外业务保驾护航

该企业作为智能家电的领军企业，近年来业务扩张势头迅猛，从国内本土的3C消费电子业务起家，拓展到家用电器品类，紧随着又逐渐开展海外业务。出海战略让公司重新审视其数字化举措。

■ 转型背景

在开展海外业务初期，公司在管理业务数据自动化程度低，比如采用Excel输入数据并在本地存储。但随着业务规模的拓展、渠道及门店管理规模量愈发复杂，传统模式难以为继。但公司研发资源有限，希望利用云原生技术改进效率。

■ 转型思路及方案

公司2019年开始与微软的合作，随着海外业务规模的扩张，云原生解决方案经历了三个方面的深化

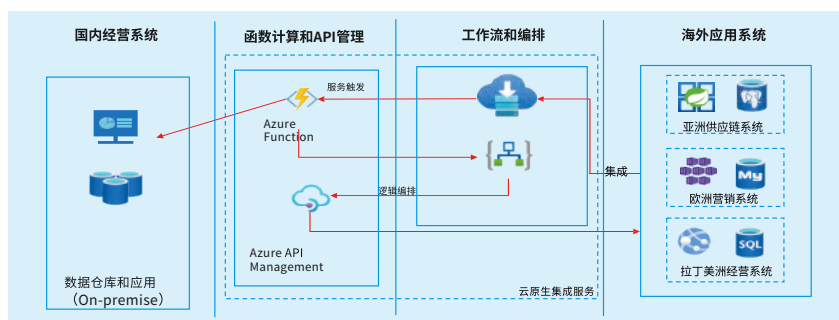
1. 系统切入，打通数据链条

在从0到1阶段，公司的海外业务希望有一个完备的业务集成系统，打通渠道商、分销商、门店等业务参与方一系列的进销存数据，升级海外业务的管理能力。微软首先提供了Dynamics CRM系统对基于客户的零售终端管理提供系统支持，并做定制化模块部署；同时，Power Platform作为低代码的应用开发平台，支持公司用有限的研发人员快速建立海外渠道的数字化管理平台，让海外零售的业务数据能够都在线进行管理。

2. Azure：提供云端的全生命周期数据集成和管理能力

随着海外各大洲、各国家的数据都迁移至云，需要系统性实现海外数据的集成和管理。Azure基于建立好的数字管理平台进一步为客户构建了数据全生命周期的管理和挖掘能力。具体来说，首先将海外业务系统的数据利用云原生集成服务集成到Service Bus（托管的消息队列）中进行工作流编排，进一步利用Azure Functions（托管的函数计算服务）让客户实现跨区域数据的协同处理，并将数据抽取传输至数据分析系统。整个方案基于Azure的云原生平台完成，客户不仅有效管理了海外零售系统的日常业务运营、还可利用Azure高级分析功能获得对海外市场运营的更深入洞察。

图：Azure为该制造业企业提供基于云原生平台的海外业务IT架构和解决方案



3. DevOps工具链：提高研发效率

随着应用规模不断提高，传统的研发体系的弊端逐渐显现，开发流程各个角色环节配合仍不流畅，许多云端应用的发布及更新甚至需要手动部署。DevOps是基于Azure的端到端工具链平台，提供了开发全生命周期的管理能力。帮助公司实现从项目的源码版本管理到测试、发布的流程自动化，从而整体上提高了公司的研发效率及质量。

■ 转型价值及收益

- 1. 团队研发管理效能的提升：**云原生现代化应用开发及项目管理流程自动化的理念，提升了研发团队不同角色协同办公的研发效率；同时低代码平台的应用从提高人效及节约人力成本的经济角度也显现了巨大价值。
- 2. 降低开发者门槛：**对企业管理者而言，一方面能以更少的这个工程师来满足业务扩张的需求，另一方面在市场上也可以招募到更多基于低代码开发的人才

为什么选择微软Azure云原生平台

- 1. 微软产品家族广而全：**对于客户的多样化需求，微软都有现成的产品及服务进行支持，能迅速满足客户的当下需求。
- 2. 微软的基础设施分布广泛：**客户在出海过程中，业务分散在各大洲；微软在客户涉及的业务区域都有相应布点，能快速提供资源帮助客户部署应用；同时也能兼顾不同区域对于数据隐私及保护的法律法规诉求。
- 3. 微软智能云无缝集成应用程序、数据和流程：**客户在拓展需求时，微软都能很顺利地集成更多的产品为客户提供服务，同时也很方便地集成第三方厂商的产品。

跨国保险金融企业 - 利用云原生平台在中国市场打响互联网战役

该企业是一家总部位于加拿大的传统保险公司。在进入中国市场并深耕本土化过程中，公司希望能根植中国的互联网肥沃，实现本土化线上业务创新。

■ 转型背景

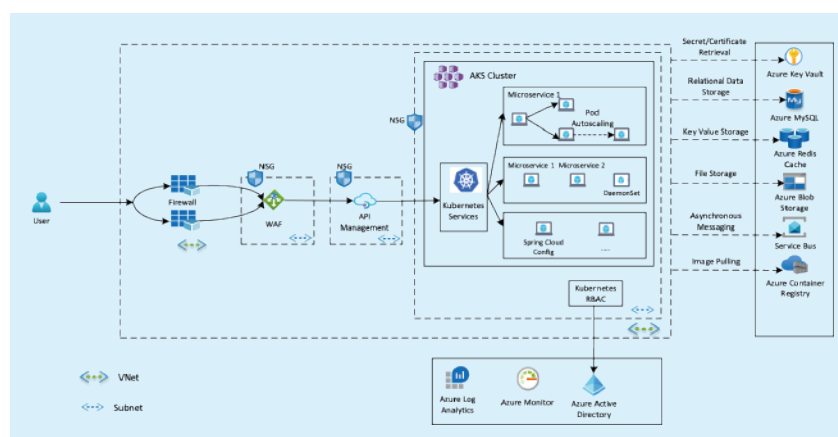
2019年，公司希望在中国开展互联网业务以更好适应中国市场需求。但此时入局，国内保险互联网业务已发展得如火如荼，公司将面临激烈的同业竞争；与此同时，公司也希望能将国内分部作为集团互联网业务的试验田。因此，公司希望通过更现代化的技术或开发流程来快速实现业务应用的改造甚至重建，以实现互联网业务转型。

■ 转型思路及方案

公司基于微软Azure平台的容器技术实现应用改造，把传统的业务模式迁移至云端微服务架构；随着现代化应用积累到一定规模，逐步建立起云原生平台技术的使用标准，实现从应用改造、安全运维、开发部署等端到端流程把控，让核心保险业务高效安全地朝着微服务方向进行分布式解耦。

- 1. 解耦和再造业务应用：**通过Azure云原生容器化、微服务架构改造互联网保险业务应用，如对微信平台售卖的意外险、车险等保险产品或者保险咨询等服务进行应用解耦，在打造互联网业务收入增长曲线的同时也获得更多客户触点。
- 2. 重塑技术管理体系及机制：**公司从传统业务架构转变成微服务架构是一个艰巨的过程，微软在早期介入后，基于对行业领域知识的理解，从企业管理角度帮助公司树立正确的云原生应用管理理念及标准，包括建立了从应用改造、发布、运维等标准化体系，并逐步完善其管理机制，并基于该架构协助公司进行产品和技术选型。
- 3. 满足业务弹性扩张需求：**公司的业务具有周期性，比如在春节前后“开门红”的促销活动业务规模量激增，让传统IT架构承压。利用Azure平台的弹性灵活的特点，一方面可以实现迅速扩容，另一方面也实现了有效的成本控制。

图：Azure为该保险金融企业提供的云原生平台解决方案



■ 转型价值及收益

- 1. 现代化业务体系的形成：**业务获得现代化能力，包括天然的高可用性、自愈性、滚动升级无宕机、快速弹性等，帮助公司更好地基于互联网开展业务。
- 2. 标准化IT管理体系的建立：**通过云原生容器镜像的交付模式，统一了不同应用的开发语言、依赖库等使用标准，实现了IT管理的标准化、自动化、敏捷性及资源的节省，降低了管理难度。
- 3. 新技术规范的建立：**形成基于现代化应用体系的云原生应用平台技术规范，适应现代化业务的研发、发布、监控和安全合规能力，也提高了公司在Global范围内的技术话语权。
- 4. 业务创新速度的提升：**云原生的微服务架构已成为了公司认可的理念，未来应用的改造及创新也会基于该架构推进，让实现新一代核心保险业务成为可能。

为什么选择微软Azure云原生平台

- 1. 微软提供高可用的服务协议：**作为国际企业，公司在选择技术厂商时倾向于有平台侧的SLA背书；微软Azure提供可用性极高的SLA协议，满足该公司的转型需求。
- 2. 微软具备成熟的云原生产品及方法论体系：**公司最开始尝试开源，但面临很多运维的困难，微软介入后，不仅利用Azure平台对公司进行容器化架构改造，还从安全合规的角度为其梳理标准及创新流程体系，从产品和方法论体系助力客户真正打造新一代保险业务模式。

领先机器人 流程自动化 企业 - 基于 云原生平台 加速智能化 和全球化战 略

该企业是一家主张人机协同的科技公司,提供国际领先的RPA+AI开发平台和解决方案来帮助企业客户整体布局和加速智能自动化转型。公司与微软达成深度战略合作,展开云原生RPA应用深度融合的探索。

■ 合作模式

随着公司业务的升级,与微软的合作不断加深:起初,公司为了更好地服务企业客户,将客户需要的资源在云上进行有效管理,并通过微软云提供对外的RPA SaaS服务;随着产品及技术的升级,公司加大了对AI相关技术的使用,需要利用微软云的GPU资源进行模型训练;2021年,公司选择微软云的集群帮助业务出海。

- **高效稳定的服务器,提升 AI 能力:**在持续优化 AI 效果的过程中,公司选择通过离线训练的方式不断优化数据模型。经过对市场上主流公有云服务提供商的全面审慎评估,并综合对比采购物理服务器的能效,最终选择 Microsoft Azure 智能云平台提供的高性能GPU系列服务器。
- **为全球用户提供可靠的安全支持和隐私保护:**微软Azure智能云作为国际品牌公有云服务,能够提供完善的技术特性,为客户提供国际化的解决方案和技术支持。不仅如此,在安全性、合规性和隐私保护方面, Azure 作为一个中立的云平台,提供了全面的合规性与隐私保护承诺,在服务条款中明确声明不会将客户的数据用于商业目的,满足了公司对数据隐私的保护要求。
- **提高客户本地化需求实施效率,实现快速交付:**公司作为一家 To B 的企业,在面对客户本地化的需求时,微软的实施团队会基于 Azure Kubernetes 服务以及丰富的 PaaS 中间件,实现快速交付。

在客户侧响应协同上,公司来主要独立服务客户,由微软云提供能力支撑,包括:① **基于微软云架构搭建海外社区环境,为海外客户免费提供产品及服务;**② **客户要求使用微软云时,公司以微软云多租户方式为客户定制化部署产品功能。**比如公司在服务一家大型快消客户时,用到微软云的AKS (Azure Kubernetes托管服务)、WAF (Web Application Firewall Web应用防护系统)、扩展循环。在AKS上部署对话式机器人产品集群,配置对应的API,客户可根据应用场景自定义开发,集成到客户内部的通讯软件,以实现智能对话式机器人服务的使用。

■ 合作价值

1. **基于云原生的RPA应用助力客户数字化转型:**云原生架构的应用,展示出满足客户随时随地调用资源需求的灵活性,保障异地备份、业务持续的云端稳定性,满足数据隐私需求及风险控制的安全合规性。
2. **品牌协同效应:**微软作为全球云计算领先企业,与作为RPA技术成长期公司联合,在客户界面具备更强的产品品牌效应。
3. **出海业务的有力推手:**公司将进一步与微软携手推广海外RPA云服务,将在海外建设更多的基于微软云数据中心。
4. **生态能力协同和建设:**企业可以使用微软云更多服务,如CV、小语种语言识别、虚拟桌面、基于SaaS的RPA服务。

选择与微软智能云合作的原因

1. 微软云的产品性能、安全性、服务能力都是行业顶尖水平,能够保障公司的业务的安全性和可用性。比如在AI模型训练的POC过程中,各大云厂商只有微软云的V1版本的GPU服务器能够在性能上满足需求,同时相比其他主流云厂商,微软覆盖全球的多数据中心,能够保障公司全球业务的稳定拓展。
2. 针对公司业务出海需求,微软云从云方面的架构、技术上及海外数据安全合规上给公司提供很多协作;同时,微软也提供了详细的技术指导,帮助公司更好使用如AKS等功能。

图:Azure为该保险金融企业提供的云原生平台解决方案

	Development	DevOps	Monitoring	Networking	Storage	Security
Take advantage of services and tools in the Kubernetes ecosystem	 	 	 	 	 	
Leverage 100+ turn-key Azure services	 	 		 	 	

**易观分析：企业应积极采用云原生架构和技术
作为驱动数字化转型底层能力**

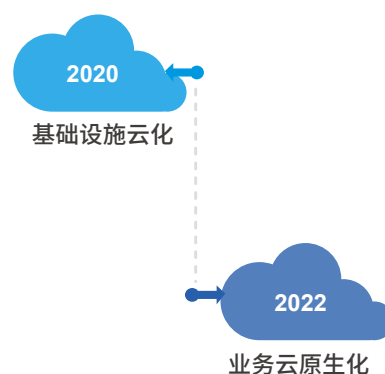
易观分析：企业应将云原生架构和技术作为驱动数字化转型的关键能力

趋势：

企业云原生战略正从“基础设施云化”走向“业务云原生化”

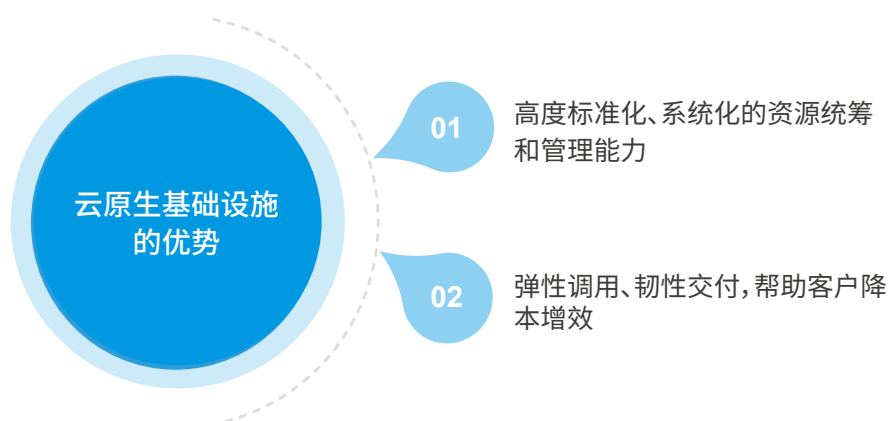
云化的趋势已被广泛共识，容器、基于serverless部署的应用以及service Mesh等云基建的不断推进使得企业上云渗透率持续提升。

当大厂供应商主导完成企业云原生第一阶段——基础设施云化，企业开始进入云原生第二阶段——业务云原生化。比如通过微服务架构构建可弹性扩展的云原生应用，实现应用和业务强耦合。容器是云原生的底层技术设施和技术基础。DevOps是贯穿在云原生应用云化中的应用开发平台，是云原生发展的基石和工具。



云原生基础设施云化的意义

- 1.云原生基础设施提供高度标准化、系统化的资源管理能力，提升基础设施使用效率。基于云原生架构，云资源得以统一供给，可以有效提升云资源的利用率，减少资源重复建设和使用。另外，资源与应用一体化的调用与运维能力，能够全景化的监视整个开发运维流程，提升整体的开发运维效率。
- 2.云原生基础设施具备弹性调用、韧性交付的能力，帮助客户降本增效。它能使资源的管控粒度更细、扩容效率更高。比如在电商业务的高峰期，无需提前储备过多的云资源来保证业务流畅运行，而是根据业务吞吐量灵活弹性扩容，这样可以增加资源的有效利用率，减少重复开发和资源浪费，实现资源利用的最大化。



业务云原生正在加速落地行业

进入2022年以来，外部环境不确定性加剧，Covid-19疫情防控、地缘政治冲突和局部战争的出现，使得金融、零售、制造等诸多行业的经营连续性和稳定性受到冲击。这让更具韧性的云原生基础设施的价值凸显，企业正在加速探索业务云化和云原生化。

对于业务人员来说，云原生技术能够提高业务部门系统对外部冲击时的稳定性、自愈性和可修复性。能够有效预防因为资源调配问题导致的业务中断，提升系统抗风险能力。利用云原生平台提供的云数据库，开发框架以及运维工具等，让IT开发人员的关注点可以从基础设施转向业务本身的功能和逻辑实现，也能够让企业更加着眼于业务的数字化升级。



A 银行业 更敏捷的需求响应	B 零售电商 更高效的应用部署	C 工业制造 可观测的安全管理
业务价值：DevOps可以帮助银行重构开发，测试运维的流水线，提升业务需求响应速度，有效提升资源利用率，实现新敏态业务形态。 典型案例：某银行为了降低现金管理中心的系统脆弱性，提高业务响应速度，从而实现业务的流畅运行和升级迭代。所以该银行采用了DevOps流程。DevOps 快速开发的设计架构不仅让IT开发品质提升，同时也减少很多运维工作。因为采用云原生的开发模式，在前期投入最小的成本，但是在资源利用上能够获得更大的弹性。更重要的是因为减少运维工作，让IT部门可以专注于开发、回应业务的需求，优化目标的达成，同时更好地提升银行客户体验。	业务价值：很多零售企业的业务模式繁多，不同的业务模式对交易系统的要求也差异很大，对云原生的解耦能力提出了非常高的要求。容器架构因为轻量，易移植交付的特点，可以适配多种操作系统，可以帮助零售企业更好地管理统筹整个开发运维系统，管理平台统一调度资源，可以节省成本，提高部署效率。 典型案例：某零售品牌在电商节参与大型促销，峰值是平时的好几倍，给服务器带来较大的压力。公司采用云原生架构，从入口到后端，做了一个端对端全链路的限流降级熔断的机制，通过整个全链路压测系统去演练，并且提前一个月演练成功。云原生架构成功支撑了该零售品牌在双 11 当天的电商交易系统。	业务价值：通过云原生平台较强的观察性，可以加强预警，帮助企业提前获知故障或故障概率高的节点，可以提前准备容灾应对方案。容器云还可以实现跨网络和设备的节点管理，并通过多个备份实现高可用性。云原生灵活多变的弹性架构和高效的容灾方案可以帮助公司业务稳定运行。 典型案例：某制造业公司希望支持更安全的代码构建，从而为容器镜像安全提供可见性，这就要求在软件开发生命周期的早期便发现安全问题。云原生架构可以提供全面的安全问题检测服务，比如漏洞扫描、动态威胁分析等能力，在前期完成镜像安全扫描和已知漏洞的修复等，保证镜像进入生产环境后的安全，使得在应用部署前能有效减少攻击面。

易观建议:与领先云技术伙伴合作,利用企业级平台支撑业务云原生化

易观分析认为,在云原生发展趋势之下,随着客户数字化转型需求的升级,一部分基于云原生架构的独立软件开发商 (ISV) 将会出现。这些ISV可以利用云原生PaaS平台更加敏捷高效完成云原生软件开发、部署和运维,交付客户数字化转型所需的业务应用。

与此同时,无论是软件购买者还是供应商,想要让云原生技术更加贴合业务场景以及更好地落地,一个开放、标准、开源的云原生生态必不可少。立足领先的云原生技术平台和基于可持续软件工程原则的应用开发生态,开发者可以加速技术革新和资源共享,提升开发、部署、运维全流程效率,实现对云端基础设施的灵活调用。

易观建议利用云原生能力驱动数字化转型的技术用户:



本报告联合发布方



关于易观分析

易观分析成立于2000年，是中国数智化领域专业的科技和 market 分析机构。旗下产品有“易观千帆”- 国内权威的数字经济行情信息和数据终端，“易观博阅”- 中国数字经济和数字技术应用领域覆盖面最全的分析师服务订阅平台。

经过22年同科技、金融、零售、制造等行业客户的深入互动和持续服务，易观博阅平台积累了超过8000份独有的分析成果，内容涵盖数字经济和数字技术应用全链条中的业务场景、技术厂商、产品数据库，行业案例库等，并拥有“易观实力矩阵 (Power Matrix)”，“应用成熟度曲线 (AMC)”等多个易观自主知识产权的分析模型。

同时，易观博阅聚集了易观最资深的行业分析师、技术分析师以及与易观长期合作的业界专家，通过提供分析报告、分析师问询、workshop等多种方式为客户输出价值，帮助客户在数字化战略、数字化业务以及数字技术应用等方面系统分析评估现状，明确转型方向，做出正确决策，导入适合的产品、技术、合作伙伴和最佳实践，最终帮助客户打造卓越的数字业务能力，实现技术ROI的跃升，完成面向数智化时代的转型。

订阅易观博阅，让科技有道，让创新可依。



网址: www.analysys.cn

微博: Analysys易观分析

公众号: 易观数字技术应用

客户热线: 4006-010-231

本报告包含的所有内容，除易观分析（以下称“本公司”）经授权引用“微软(中国)有限公司”的相关产品和技术素材外，其余部分知识产权归本公司所有，受中国及国际版权法的保护。本公司和微软(中国)有限公司均已对本报告享有排他性的使用权，并受中国及国际版权法的保护。其它任何公司和个人对本报告内容的任何使用，包括修改、发布、转发、再版、交易、演示等行为都应获得本公司授权。如需引用，需注明来源：“易观分析-《企业级云原生平台驱动数字化转型》”