中 国 开 源 云 联 盟 标 准

COSCL 0002-2017

企业级容器云平台技术要求

Technical Requirements of Enterprise Level Container Cloud Platform

2017-12-26 发布　　2018-03-01 实施

中国开源云联盟　发布

目　　次

前　言

本标准按照GB/T 1.1-2009给出的规则起草。

本标准起草单位：中国电子技术标准化研究院、北京数人科技有限公司、英特尔（中国）有限公司、央视国际网络有限公司、阿里云计算有限公司、北京凌云雀科技有限公司、北京轻元科技有限公司、中国电子科技集团公司第二十八研究所、华为技术有限公司、联想(北京)有限公司、中兴通讯股份有限公司、杭州质云科技有限公司（网易云）、腾讯云计算（北京）有限责任公司、烽火通信科技股份有限公司、Hyper、海南易建科技股份有限公司、浪潮电子信息产业股份有限公司、北京华胜天成科技股份有限公司。

本标准主要起草人：陈志峰、杨丽蕴、肖德时、吴涛、杜永丰、王雷、闫长海、王建利、王昕、汪敏、赵华、汤文军、李伦、黄庆兵、邹辉、谭峤、王旭、赵杰、朱青林、张航源、王瑞、吴振、丁乙、张亚辉、梁钢。

引　言

云计算和DevOps都是“敏捷 IT”理念下的技术组合，旨在快速开发并交付业务，而且大规模稳定运行，挑战来自“高速度”和“低风险”两个方面。传统 IT 业务建设要经历规划、采购、开发建设、部署上线的冗长流程。虚拟化将硬件准备缩短到了分钟，但是应用构建、测试、部署和监控等应用运维依然以“月”记。面对互联网海量用户和海量数据的挑战，未来速度和风险的矛盾将愈演愈烈。其次互联网和大数据对于传统IT从架构到运维都是“从零到一”的过程，相关经验一直由互联网企业和开源社区掌握。传统 IT 厂商从产品到实践无法提供能落地的支持，企业往往陷入难以起步、一试就错的困境，难以进入通过快速迭代来培养队伍的正向循环。我们看到，“敏捷”的挑战具体为下面几个方向：

- 资源方面，解决引入开放平台（X86+Linux）带来的复杂度，统一纳管物理机、虚拟机和云主机等异构硬件
- 架构方面，支持应用向微服务架构转型，提供相应的运维和运行环境
- 中间件方面，为传统中间件在“云”环境下找到对应产品，并且针对大数据和高并发场景，开发出新的功能组件
- 应用交付方面，解决各个环境隔绝，交付物不统一，手工操作效率低且风险大

总之，针对云时代的软硬件环境建立标准化，有了标准化才有自动化，自动化成体系就形成了平台化，也就是常说的DevOps落地。基于上述场景，企业级容器云平台包括了三个层面的功能，既针对异构资源纳管的容器运行环境，针对微服务和分布式架构支撑的基础架构PaaS（iPaaS—infrastructure PaaS）和帮助用户快速搭建分布式应用的应用 PaaS（aPaaS—Application PaaS）。

本标准在2017年11月16日发布标准草案，2017年12月8日-12月22日期间在开源云联盟和行业中征求意见，并于2017年12月26日正式发布。

企业级容器云平台技术要求

1 范围

本标准规定了企业级容器云平台的功能和非功能要求。

本标准适用于企业级容器云平台的开发、测试和评估。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 32399-2015　　信息技术 云计算 参考架构

GB/T 32400-2015　　信息技术 云计算 概览与词汇

3 术语

GB/T 32399-2015与GB/T 32400-2015 中所有术语适用于本文件。

3.1

云平台

由云服务商提供的云基础设施及其之上的服务软件集合。

3.2

容器云平台

基于容器技术的云平台。

4 缩略语

PaaS	平台及服务（Platform-as-a-Service）
OCI	开放容器组织（Open Container Initiative）
CNI	容器网络接口（Conteinre Network Interface）
CNM	容器网络模型（Container Network Model）
Devops	开发运维自动化（Development and Operations）
LDAP	轻量目录访问协议（Lightweight Directory Access Protocol）
SLA	服务等级协议（Service-Level Agreement）
CI/CD	持续集成/持续交付（Continuous Integration/Continuous Delivery）
QoS	服务质量（Quality of Service）
DDoS	分布式拒绝服务(Distributed Denial of Service)

5 容器云平台功能架构

企业级容器云平台功能架构如图1所示，主要包括管理容器运行引擎、容器网络，容器编排等；非功能要求包括可用性、兼容性、安全、易用性、负载优化等内容。企业级容器云平台主要是实现在云平台上运行云应用时取得最优化的效果。

图1 企业级容器云平台功能架构

6 功能组件

6.1 基础设施要求

6.1.1 容器主机

a) 应支持物理机；
b) 应支持虚拟机。

6.2 运行时环境要求

6.2.1 容器运行引擎

a) 应支持运行引擎热升级；
b) 应支持运行引擎出现故障时不影响用户应用；
c) 应支持OCI规范；
d) 应支持容器生命周期管理，包括但不限于创建、启动、停止、删除等；
e) 应支持性能数据采集，即具备容器级别的性能数据收集能力；
f) 应支持基本的镜像管理操作例如pull、push等，并能够通过镜像构建容器文件系统；
g) 应支持提供日志数据传输通道；
h) 应支持卷管理，包括本地和网络卷；

i） 应支持容器安全配置，包括但不限于selinux，apparmor，seccomp；
j） 应至少支持一种主流的写时复制文件系统，如aufs，Overlayfs等；
k） 应该支持对接kubernetes，swarm等的容器编排工具；
l） 可支持网络插件。

6.2.2 容器网络

a） 应支持CNI或CNM其中一种接口规范；
b） 可支持多网络平面。

6.2.3 容器存储

6.2.3.1 镜像存储

a） 应支持分层的联合文件系统；
b） 应支持写时复制，以减少新建容器时复件文件的代价；
c） 容器平台应该至少支持一种文件系统：aufs、devicemapper、overlay或overlay2。

6.2.3.2 数据卷

应支持数据卷的操作，至少包括创建、删除、挂载等操作。

6.2.4 安全

6.3 容器编排和管理要求

6.3.1 容器调度要求

容器调度是指基于抽象的底层资源对容器进行管理，从而保证资源的合理分配和容器的正常运行。容器调度满足以下要求：

a） 应提供资源分配与调度功能，包括统计资源利用率，并能够按照策略动态分配资源；
b） 应提供容器动态迁移功能，包括伸缩迁移、故障迁移等功能；
c） 应提供空间（租户间、租户内）逻辑隔离（安全）功能，包括：在不同的空间内，部署同样的应用，应用部署过程相同，运行环境通过空间逻辑隔离。常见的场景有测试、联调、开发环境的部署；
d） 应提供服务注册发现，包括：服务创建后，调度器可以主动发现运行的服务，其他的服务可以通过服务的名称和端口来使用服务；
e） 应提供服务有状态功能，如通过挂载数据盘，绑定公网IP，实现有状态服务；
f） 应提供灰度升级功能，实现应用新老版本之间的平滑过渡；
g） 应提供负载均衡，服务提供负载均衡能力，请求随机分配给副本处理。外部进入的流量可负载均衡进行分流；
h） 应提供错误恢复，在副本运行出现错误时，可以自动重启或者快速迁移，以修复错误状态。
i） 可支持自定义调度器。

6.3.2 容器编排要求

容器编排满足以下要求：

a） 应支持通过编排模板文件，自动化地部署和管理一个容器应用，例如 docker compose、kubernetes yaml 等；

b) 应支持配置容器的各项属性，如容器镜像、环境变量、端口、启动命令、资源规格、容器录挂载等；
c) 宜支持容器应用的可视化编排；
d) 宜支持配置容器间的拓扑关系；
e) 宜支持配置容器的虚拟网卡，如网口 ip 地址、网络上下行限速等，可支持多网卡；
f) 宜支持配置容器需要使用的中间件；
g) 宜支持配置容器的存储，如存储类型、存储大小等；
h) 宜支持配置容器的服务注册信息，如服务名、域名、协议类型、负载均衡策略等；
i) 应支持配置容器的副本数；
j) 可支持容器编排过程图形化方式；
k) 可支持健康检查图形化设置；
l) 可支持服务依赖关系设置；
m) 可支持定义路由；
n) 可支持定义服务，设置全局服务。
o) 可支持配置容器的启动顺序；
p) 可支持配置容器间的亲和性/反亲和性关系；
q) 可支持配置容器的弹性伸缩策略；
r) 可支持配置容器的滚动升级策略；
s) 可支持配置容器的健康检查策略；
t) 可支持配置容器是否使用特权模式。

6.3.3 服务（内部）管理要求

服务（内部）管理满足以下要求：

a) 应支持通过模板方式进行服务创建；
b) 应支持服务启动、停止、重启等服务状态控制能力；
c) 应支持服务部署配置和管理；
d) 应支持根据服务负载与资源使用情况，能够通过服务弹性扩缩容，达到服务正常使用；
e) 应支持根据服务负载与资源使用情况，能够将负载均衡至每个服务实例；
f) 应支持服务升级和回滚，提供无间断服务升级回滚能力；
g) 应支持服务健康检查（可用性，对于容器生命周期的），针对服务健康状态进行探测；
h) 应支持服务容错机制，实现服务故障自动恢复能力；
i) 应支持通过命令行或者图形界面方式进行服务注册，服务注册信息包括：服务标识、服务名称、服务地址等；
j) 应支持通过命令行或者图形界面方式进行服务发现，并提供二次开发接口；
k) 应支持容器应用的部署；
l) 应支持容器应用的查询；
m) 应支持容器应用的删除；
n) 应支持容器应用的扩容、缩容；
o) 宜支持服务一键部署能力；
p) 宜支持通过配置服务发现策略，满足不同业务需求，包括：随机、轮询等；
q) 宜支持容器应用的启动、停止；
r) 宜支持容器应用的滚动升级；

s) 宜支持容器应用的一种或多种发布方式，包括灰度发布、蓝绿发布等；
t) 宜支持容器应用的负载均衡；
u) 宜支持容器应用的服务发布；
v) 宜支持容器应用的自动化部署；
w) 可支持服务白名单和黑名单功能。
x) 可支持服务（有状态和无状态）热迁移，提供服务容器运行时状态保持，达到容器迅速复制并快速恢复运行；
y) 可支持容器应用的容错机制；
z) 可支持容器应用的健康检查；
aa) 可支持容器应用性能监控（性能/事件/日志/告警　至少包括：CPU、内存、存储、网络）（放在监控中）；
bb) 可支持容器应用的重启。

6.4 中间件及开发运维自动化要求

6.4.1 中间件

6.4.1.1 基本要求

a) 应提供代理服务器镜像模板，如 Haproxy、Nginx 等；
b) 应提供应用中间件镜像模板，如 Tomcat、Websphere 等；
c) 宜提供语言类镜像模板，如 Java、Python、Golang 等。

6.4.1.2 数据库服务/缓存服务

a) 应提供数据库服务镜像模板，如 Mysql、Postgres 等；
b) 应提供缓存服务镜像模板，如 Memceche、Redis 等。

6.4.1.3 流式服务/消息

a) 应提供流式/消息服务镜像模板，如 Kafka、RabbitMQ 等。

6.4.1.4 代码版本服务

a) 应提供代码版本服务镜像模板，如 GitLab 等。

6.4.1.5 镜像管理服务

a) 应提供镜像管理服务模板，如 Harbor 等。

6.4.2 开发运维自动化（Devops）

6.4.2.1 容器化持续集成

a) 应对接代码仓库，并支持对接主流代码仓库，如 Github、Gitlab、Subversion 等
b) 应提供代码扫描工具（平台提供与客户自定义），平台支持和客户代码扫描平台（如 SonarQube）对接，支持对客户代码的扫描功能；
c) 应提供容器化编译工具（平台提供与客户自定义），提供 java、GO 等语言的容器编译工具，或与客户容器编译工具做对接，支持多语言的编译打包功能；

d) 应提供容器环境自动化测试，容器环境自动化测试工具或与客户测试工具做对接，支持自动化单元测试功能；
e) 应支持获取编译得到的产出物，如编译生成的代码包/可执行文件，代码扫描报告，单元测试报告等；
f) 支持获取 dockerfile 后能进行自动化 docker build 镜像，并且自动将镜像推送到指定镜像仓库；
g) 应支持镜像扫描工具（如 Clair）或和客户镜像扫描工具做对接，对推送镜像进行分层扫描，并生成扫描报告；
h) 应支持持续集成日志功能，即自动生成并在平台实时显示构建状态，存储每次构建的日志信息；
i) 可支持容器编排对象编排模板的生成，镜像新版本生成时，容器编排对象（蓝图）自动生成新版本；
j) 可支持容器编排对象编排模板自动化测试，容器编排对象有新版本生成时，自动部署容器编排对象进行测试；
k) 可支持并行执行测试；
l) 可支持多用户同时提交持续集成和发布的流程。

6.4.2.1 容器化持续发布

容器化持续发布功能包括：

a) 应支持自定义持续发布流程，支持客户在平台上自定义满足自身要求的持续发布流程步骤；
b) 应支持容器自动化部署，支持读取镜像并将单个或多个微服务自动化部署在指定的集群环境中；
c) 应支持容器自动化测试和测试报告获取，支持启动客户自定义测试镜像，对被测服务进行接口测试；支持测试报告的存储和获取；
d) 应支持部署流程支持人工控制，持续部署流程的人工控制，暂停执行流水线，进行人工测试或人工审核等；
e) 应支持发布策略（不间断升级、蓝绿发布等）功能，支持更新已有业务，发布时支持不间断升级、蓝绿发布、灰度发布等方案；
f) 应支持部署流程执行过程/结果通知，将构建流程/结果通知到相关人员；
g) 应支持持续发布日志，自动生成并在平台显示实时流水线步骤及状态，存储每次持续发布的日志信息。

6.5 云管理平台要求

6.5.1 组织与用户管理

6.5.1.1 用户管理

用户管理功能包括：

a) 应提供用户管理的功能或集成第三方用户管理系统的能力。
b) 应提供用户管理的功能，应包括管理员浏览、添加、更改和删除用户的功能。
c) 应提供用户管理的功能，应包括用户自注册的功能。

6.5.1.2 组织管理

组织管理功能包括：

a) 应提供组织管理的功能，包括用户组织的浏览、创建、更改和删除。
b) 应提供指定组织管理员的功能。
c) 组织管理员应能通过管理平台为组织添加和删除用户成员。

6.5.2 用户认证与授权

6.5.2.1 认证

认证功能包括：

a) 应支持云平台与镜像仓库的单点登录的功能，通过统一的用户名，授权访问云平台的资源的镜像仓库；
b) 应支持一种或多种用户认证的方式，其中至少包含一种认证方式，如 OpenID Connect；
c) 应提供用户登录网络端点应得到传输层安全加密的保护。

6.5.2.2 授权

授权功能包括：

a) 应支持不同工作空间范围的角色授权；
b) 应支持整个平台范围的角色授权；
c) 应提供默认的角色，至少包括但不限于只读角色、读写角色和管理员角色；
d) 应提供多种角色权限管理功能，只读角色的用户只能浏览和读取云平台中相应范围的资源属性；读写角色的用户可以浏览、读取、创建、更改和删除云平台中相应范围的资源；管理员角色的用户可以进行云平台中相应范围的一切操作，包括读写管理资源的操作和管理授权规则的操作。
e) 应提供本地账户密码强度检查；
f) 应提供防止暴力账号破解保护功能，如综合使用双因子认证、验证码、客户端证书验证、密码尝试次数限制等技术；
g) 应支持和三方的用户管理模块对接实现单点登录，如 LDAP 域，三方社交平台等；
h) 应支持按项目或者资源组分配给不同用户，同一用户在不同项目或者资源组中可以有不同权限；
i) 可支持多种方式管理用户信息；
j) 可支持为不同项目租户应用配置不同的安全策略。

6.5.3 应用目录

6.5.3.1 服务目录基本功能

服务目录基本功能包括：

a) 应支持服务的实现者发布服务到云平台上；
b) 应支持服务的使用者能够发现服务并使用服务。

6.5.3.2 应用上架和应用使用

应用上架和应用使用功能包括：

a) 应支持在应用目录上发布服务的能力；

b) 应支持同一种服务的不同服务等级或服务规格；
c) 应支持应用列表功能，让服务的使用者能从集中的服务目录中浏览到所有自己可以使用的服务，了解每个服务的作用和服务等级。

6.5.3.1 应用实例管理

应提供管理服务实例的能力，包括浏览、创建、更新和销毁服务实例。

6.5.4 资源配额与计量计费

6.5.4.1 资源配额

资源配额功能包括：
a) 应支持为用户设定资源配额的能力；
b) 应支持设定用户能使用的最大总 CPU 数的能力；
c) 应支持设定用户应用单个容器实例能使用的最大 CPU 数的能力；
d) 应支持设定用户能使用的最大总内存数的能力；
e) 应支持设定用户应用单个容器实例能使用的最大内存数的能力；
f) 应支持设定用户能使用的最大总存储容量的能力；
g) 应支持设定用户应用单个容器实例能使用的最大存储容量的能力；
h) 应支持设定用户能够部属的最大容器实例个数的能力。

6.5.4.2 计量计费

可支持多种计费方式进行计费，包括按使用量和使用时长计费。

6.6 监控日志要求

6.6.1 监控

6.6.1.1 主机（物理机/虚拟机）监控

主机监控满足以下要求：
a) 应支持实时监控主机的 CPU 利用率、内存利用率、存储利用率、磁盘 IO 流量、网络 IO 流量等状态是否正常，并提供图形化界面展示。
b) 应支持对主机的 CPU、内存、磁盘和网络的历史监控记录进行统计分析，可提供统计图表。

6.6.1.2 容器监控

容器监控功能包括：
a) 应支持实时监控外挂主机存储、共享存储的容量信息、挂载情况和连接状态等，并提供图形化展示；
b) 宜支持查看容器平台的网络拓扑关系，如以容器为中心、虚拟机为中心、物理服务器为中心、数据存储为中心显示拓扑关系；
c) 可支持容器消耗主机资源的信息；
d) 可支持容器事件监控信息。

6.6.1.3 应用监控

应用监控功能包括：

a) 可支持实时监控容器中的进程的运行状态，并提供展示结果和报警策略；
b) 可支持管理容器中的进程，如重启、停止等操作以及控制重启次数等策略；
c) 可支持应用 SLA 的监控，实时监控服务是否正常提供服务。

6.6.1.4 中间件监控

中间件监控包括：

d) 应支持实时监控中间件的运行状态，CPU、内存等系统资源的使用情况，并提供图形化展示，宜支持以报表的形式导出历史监控数据；
e) 应支持查看以中间件为中心的服务访问拓扑关系；
f) 应支持对中间件的总处理数、传输速率、连接数、队列长度等性能指标进行实时监控和图形化展示。

6.6.2 日志

6.6.2.1 控制台日志

控制台日志功能包括：

a) 应支持保存和查看对容器编排平台的操作记录，记录内容可包括操作者、操作类型、操作内容、完成状态、完成时间、执行结果、失败原因等信息；
b) 宜支持按关键字检索相关操作记录，如操作者、操作日期、操作内容等关键字进行检索，并支持将检索结果导出为文件。

6.6.2.2 本地日志

本地日志功能包括：

a) 应支持访问和下载容器或主机上的多种类型的系统日志信息；
b) 可支持容器平台相关日志的自定义日志记录存储格式、字段含义等规范化需求；
c) 可支持长期存储，并确保它们的完整性、保密性等，不得随意修改、删除，对于较大的日志量宜提供压缩机制。

6.6.2.3 流式日志

流式日志功能包括：

a) 宜支持实时采集所有系统日志和业务日志，转化为流式数据，集中收集到日志中心，需支持断点续传，无数据丢失，短时间保留等；
b) 宜支持对原始日志进行过滤、切割、归并等操作，宜支持日志数据存储的持久化、高度容错、分布式存储、低延迟等需求，宜支持对日志某些字段提供报警策略，通过日志实时监控业务的访问情况；
c) 宜支持不同维度图形化展示界面，实时分析日志趋向，宜支持多维关联分析需求；
d) 宜支持对大量历史数据进行分析，并支持以报表的形式导出分析结果。

7 非功能要求

7.1 可用性

可用性要求包括：

a) 应支持运行时环境的高可用；
b) 应支持管理节点高可用；
c) 应支持进程/主机关机/断电自动恢复；
d) 应支持应用的高可用；
e) 应支持集群多个计算节点高可用；
f) 应支持自动弹性伸缩；
g) 应支持容器故障自动迁移；
h) 应支持备份恢复；
i) 应支持平台元数据备份、迁移及恢复能力；
j) 应支持应用及服务的数据备份、迁移及恢复能力；
k) 应支持容灾策略；
l) 应支持平台组件支持高可用部署；
m) 宜支持多可用区应用部署。
n) 宜支持同城灾备和二地三中心等容灾方案。

7.2 性能

性能指标包括：

a) 平台支持并发请求量；
b) 平台 CI/CD 并发量；
c) 平台资源利用率；
d) 支持主机、存储、网络的利用率；
e) 平台吞吐率。

7.3 兼容性

7.3.1 容器运行时

容器运行时兼容性要求包括：

a) 应满足多种运行时接口的规范标准（例如 RunC, RunV）；
b) 应提供运行时需要提供查询接口（创建，启停，查询接口）。

7.3.2 镜像管理

容器镜像管理功能的兼容性要求包括：

a) 应符合体系架构的镜像制作规范的镜像可以在相同体系环境中运行；
b) 应支持不同体系结构下镜像的输出/测试/验证；
c) 宜提供镜像根据不同体系结构(AMD64,ARM64,Window,Linux)进行分类功能，索引以及查询功能。

7.3.3 调度

调度功能的兼容性要求包括：

a) 应提供对不同体系结构各种资源进行监控/数据收集以及管理；

b) 宜提供过滤机制区分容器运行所需的体系结构，并根据调度策略分配到相应的物理/虚拟主机。

7.4 安全性

7.4.1 平台安全

平台安全要求包括：

a) 应支持平台组件和应用网络的隔离能力；
b) 应支持租户间对于计算、网络、存储资源的逻辑隔离能力；
c) 应支持加密的网络传输；
d) 宜为应用提供 DDOS 防御的能力管控链路支持 TLS 连接；
e) 宜支持设置容器守护进程最小权限访问；
f) 宜支持异常攻击和入侵行为的实时检测、告警和自动防御；及时阻断非法外联；
g) 宜确保镜像仓库内的容器镜像不被恶意篡改；
h) 宜支持限制容器守护进程的资源要求；
i) 宜支持容器守护进程的审计；
j) 宜配置项加密，支持内置的 Secret。Secret 要求加密保存；
k) 宜支持当镜像签名开启时，只有签名通过的镜像才可以运行。

7.4.2 内容（镜像）安全

内容（镜像）安全要求包括：

a) 应支持镜像安全扫描；
b) 应支持镜像认证；
c) 应支持追溯和认证基础镜像的来源；
d) 应支持镜像的创建、拉取、查询、上传等接口权限；
e) 应支持将私有镜像授权给特定的用户和团队；
f) 宜支持镜像不可被覆盖。

7.4.3 运行时安全

运行时安全要求包括：

a) 应支持操作系统权限细分，包括 ulimit、capability、指定 user 身份等；
b) 应支持操作系统资源隔离，包括 namespace 和 cgroup 等；
c) 应支持采用受信注册；
d) 宜支持网络 QoS 流控和管理，支持 Vlan、Vxlan 等；
e) 宜支持限制每个容器的资源上限（资源包括：CPU，应内存，网络，挂载卷，日志）；
f) 宜可支持磁盘容量 Quota；
g) 应支持操作系统安全集成，包括 Linux security modules(SELinux,AppArmor)、seccomp 等；
h) 宜提供运行引擎漏洞在线热修复。