WOTD

51CTO

全球软件开发技术峰会

World Of Tech 2017

2017年12月1日-2日 • 深圳中洲万豪酒店

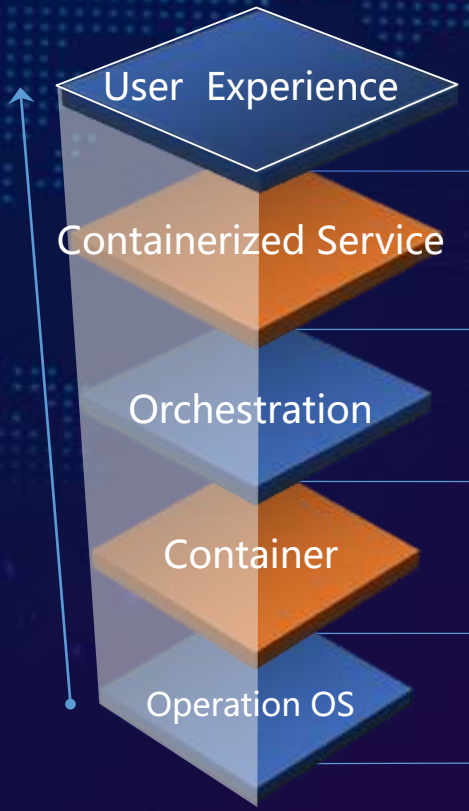
DEVELOPMENT

OCI容器标准的社区演进和OCI方案的实战

梁辰晔
华为开源能力中心主任工程师

目录

容器生态



- **用户体验** ：面向终端用户，提供相应的应用服务。
- **容器化服务** ：面向个人、企业开发者，提供容器的相关服务。
- **编排调度**：面向应用，提供便利的容器管理功能。
- **容器产品** ：面向应用，提供轻量的容器功能。
- **操作系统**：面向容器产品，提供基础稳定的内核。

争夺瓶颈——2015年全栈Docker，怕不怕？

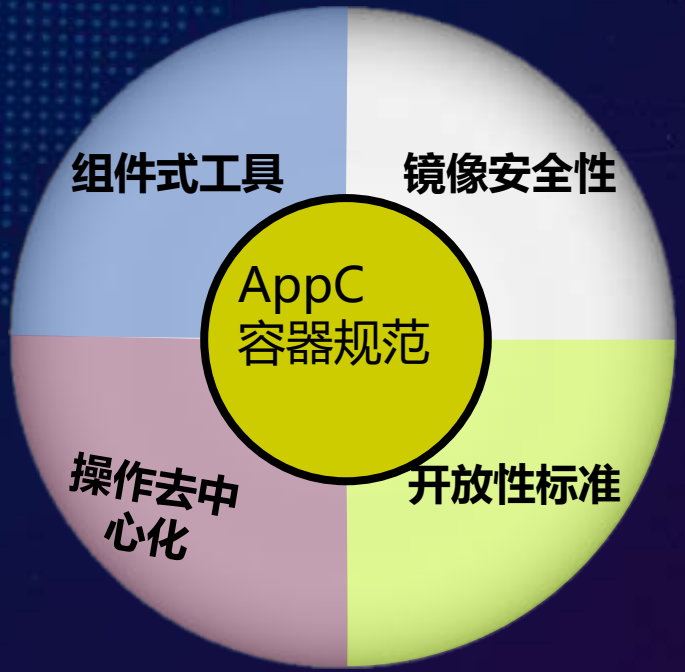
斗争的结果——OCI出现的必然性

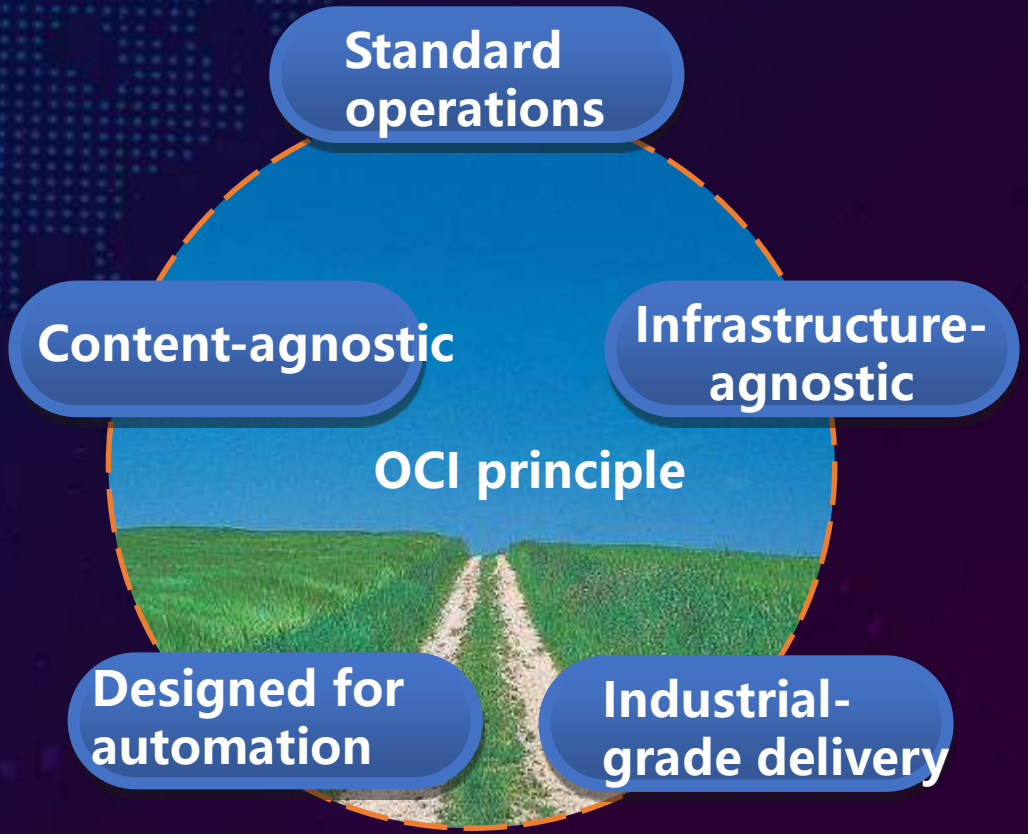
1. CoreOS 从AppC + Rocket 开始
2. Linux Foundation OCI成立

2015年大家怕Docker的什么，是否反应过度？

2017年的Google，怕不怕？

政治的技术外衣——行业的价值





社区建设的典范—— 组织+活动+规则

组织

- OCI成员
  企业，会员费加入
- TDC
  个人，通过代码贡献获取maintainer(TDC)
- TOB
  个人（9人），TDC推荐选举，参与重大决策
- Trademark Board

活动

- 邮件列表
  重大问题讨论和新方向讨论
- Github
  代码提交和问题讨论
- 在线会议
  集中处理快速决策
- 线下活动
  利用oss/cncf大会

规则

- 官方网站
  规定使命、介绍组织规则
- 代码提交规范
  共同模板
- 议事规则
  投票机制

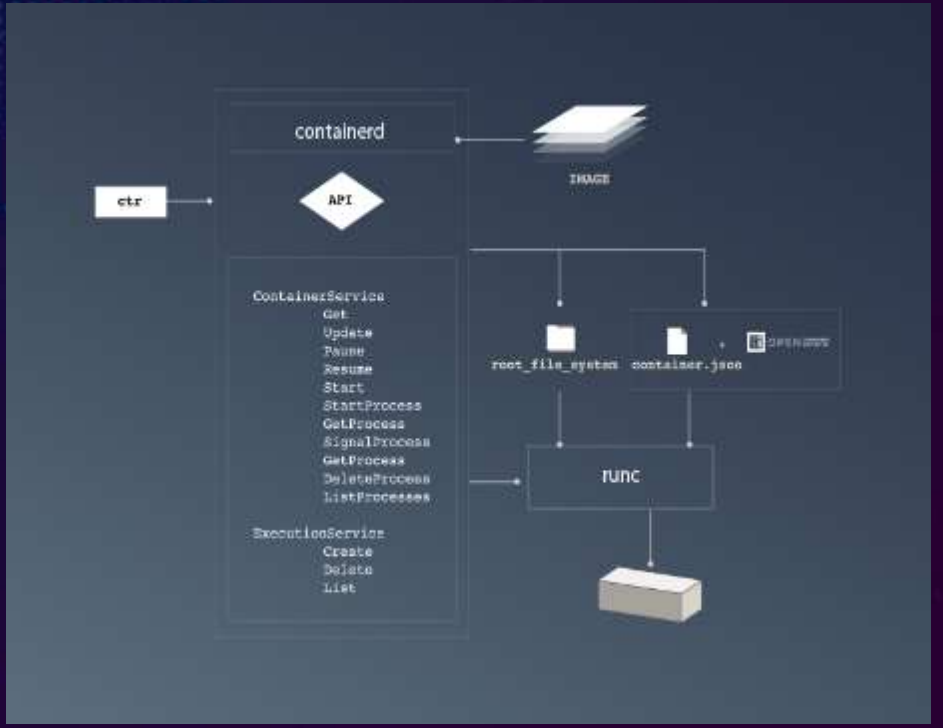
Runtime实现、标准和工具 —— 先有事实再有标准

Runc ——来自libcontainer, 标准引擎	
1. Create bundle 2. runc run myid	可以直接运行，简单轻便

Runtime spec —— OCI的核心标准	
config文件格式、参数含义	面向镜像开发者和runtime开发者

Runtime tool —— runtime标准的认证	
bundle,runtime是否符合OCI标准	遵照RFC2119规格严格验证

Containerd+runc的现有方案——runc is NOT belong to docker!



Runtime 标准举例和验证思路

```
"process": {
        "terminal": false,
        "user": {
                "uid": 0,
                "gid": 0
        },
        "args": [
                "sleep", "5"
        ],
        "env": [
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:
                "TERM=xterm"
        ],
        "cwd": "/",
        "capabilities": {
                "bounding": [
                        "CAP_AUDIT_WRITE",
                        "CAP_KILL",
                        "CAP_NET_BIND_SERVICE"
                ],
                "effective": [
                        "CAP_AUDIT_WRITE",
                        "CAP_KILL",
                        "CAP_NET_BIND_SERVICE"
```

https://github.com/opencontainers/runtime-tools

190个认证项 —— rfc2119自动更新

各种测试参数 —— generator自动生成镜像配置文件

自动化测试 —— Go 原生测试框架

Runtimetest —— 运行参数校验

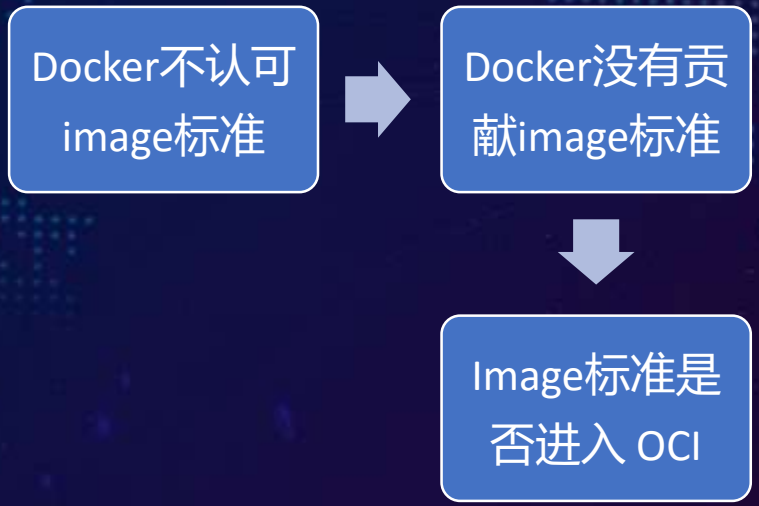
Lifecycle Test —— 运行周期命令校验

```
func validateLinuxProcess(spec *rspec.Spec) error {
        validateGeneralProcess(spec)

        uid := os.Getuid()
        if uint32(uid) != spec.Process.User.UID {
                return fmt.Errorf("UID expected: %v,
        }
        gid := os.Getgid()
        if uint32(gid) != spec.Process.User.GID {
                return fmt.Errorf("GID expected: %v,
        }
```

Image标准 —— 没有事实的标准

不应该有标准：除了分层，还可能有更优的创新方案
应该有标准： 分层是目前的事实标准，一致性、完备性都很成熟



CoreOS CTO 和 Docker CTO的骂战

Image标准 —— 规则说话

Docker不认可image标准

Docker没有贡献image标准

CoreOS基于docker2.2发起，投票通过！

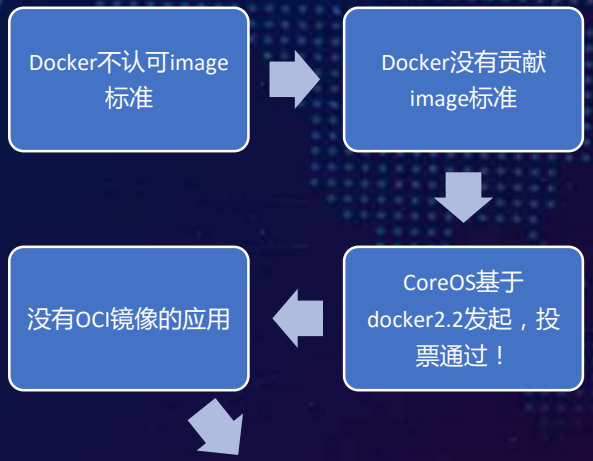
https://www.opencontainers.org/about/oci-scope-table

Bundle Format	In scope for OCI base layer	How bundle is expressed in the file system before running
Hashing for Content Integrity	In Scope for OCI base layer	Provide a standardized cryptographic method for ensuring container content has not been altered
Use of Hash as Content Addressable name for immutable containers	In scope for OCI base layer	Provide a mechanism for using the hash as a name
Archival Format	In scope for OCI base layer	Description of the serialized format and optimizations of a filesystem bundle.

Image标准 —— 没有事实的标准

发现和分发的重要性

OUT OF SCOPE!



{ "schemaVersion": 2,
"config": { "mediaType":
"**application/vnd.oci.image.config.v1+json**", "size": 7023,
"digest":
"sha256:b5b2b2c507a0944348e0303114d8d93aaaa081732b86451d9bce1f432a537bc7" },
"layers": [
{ "mediaType": "
application/vnd.oci.image.layer.v1.tar+gzip", "size": 32654,
"digest":
"sha256:9834876dcfb05cb167a5c24953eba58c4ac89b1adf57f28f2f9d09af107ee8f0" },
{ "mediaType":
"application/vnd.oci.image.layer.v1.tar+gzip", "size": 16724, "digest":
"sha256:3c3a4604a545cdc127456d94e421cd355bca5b528f4a9c1905b15da2eb4a4c6b" }
], "annotations": { "com.example.key1":
"value1", "com.example.key2": "value2" }
}

发现、分发机制的讨论

不应该有标准：有不同的网络方案，不应该限制创新
应该有标准：　没有标准，OCI完全无法商业落地

OCI Well Known URI Ref-Engine Discovery. There is a Go implementation in tools/refenginediscovery/wellknownuri. There is a Python 3 implementation in oci_discovery.ref_engine_discovery.well_known_uri.
OCI XDG Ref-Engine Discovery. There is a Go implementation in tools/refenginediscovery/xdg. There is a Python 3 implementation in oci_discovery.ref_engine_discovery.xdg.

Image-tools

- 获取　从dockerhub, 并转换
- 制作　manifest/config + 单层
- 校验　是否符合OCI标准
- 解压　变成 rootfs + config.json

OCI社区的美好告一段落

缺失了什么

	OCI	Docker
Builder	无	docker build (Dockerfile)
镜像分发	无	docker pull/push
Hub	无	docker hub
镜像本地存储	无	docker images
镜像运行	runC	docker run

内因+外因 —— 利、权

- Runtime　docker贡献
- Image　docker没有主动贡献
- Discovery　没有进入oci 1.0 scope
- Distribution　没有进入oci 1.0 scope
- CNCF　主战场转移

发现、分发机制的实现 —— OCI hub, based on beego

- Routers 发现、分发机制的API
- Storage 分层镜像的存储，对接不同存储方式
- Models 现数据库的管理
- Controllers router和后端的对接
- Logger 不同日志系统的封装

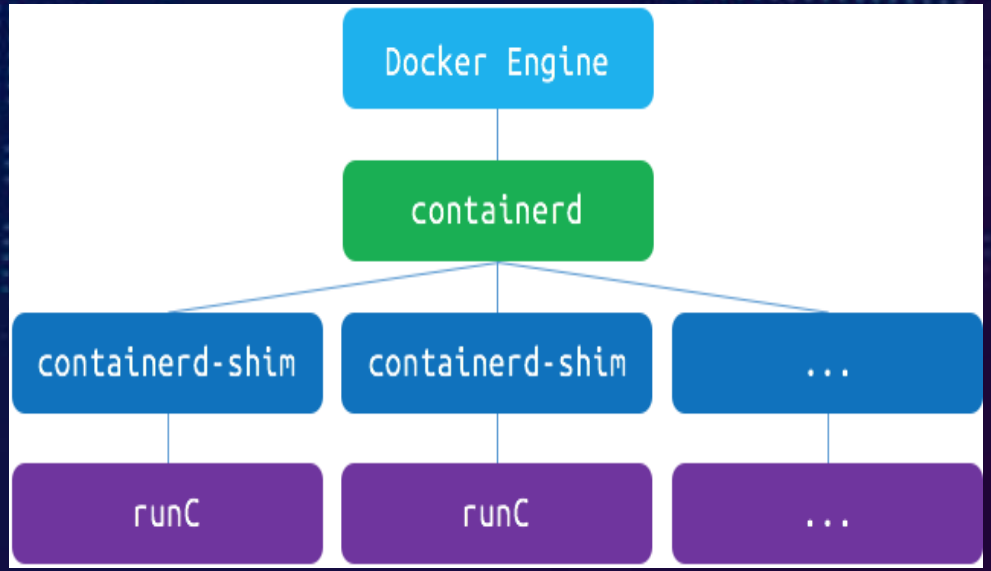
github.com/xiekeyang/oci-discovery

github.com/isula/ihub

镜像构建和本地存储

- 构建 https://github.com/containers/build
- 存储 https://github.com/containers/storage

OCI社区方案



OCI Hub

OCI Build

OCI Storage

Roadmap 2018

- 提供build 系统
- 提供Distribution API
- 支持本地存储
- 集成k8s

OCI F2F at KubeCon/CloudNativeCon 2017
Date: Dec 7th / Time: 1230pm - 2pm (Lunch will be served)
Room: Meeting Room 1, Level 1
https://kccncna17.sched.com/event/CxNK/oci-community-f2f

Agenda
OCI Organization Health/Update (via ChrisA)
V1.1+ planning?
    What’ s next for OCI?
    Distribution API (vbatts)
    Project to build standard OCI images (vbatts)
Open Mic / Q&A

initlove

Thank you!