

面向企业应用及混合云的Docker容器服务实践

杨博， IBM中国云平台服务部架构总监

主要内容



企业应用需求与混合云

基于**Docker**的容器服务实践

用于**Docker**的专家模式（**Pattern Engine**）

IBM 云平台**Bluemix/Pure** 中**Docker**服务介绍

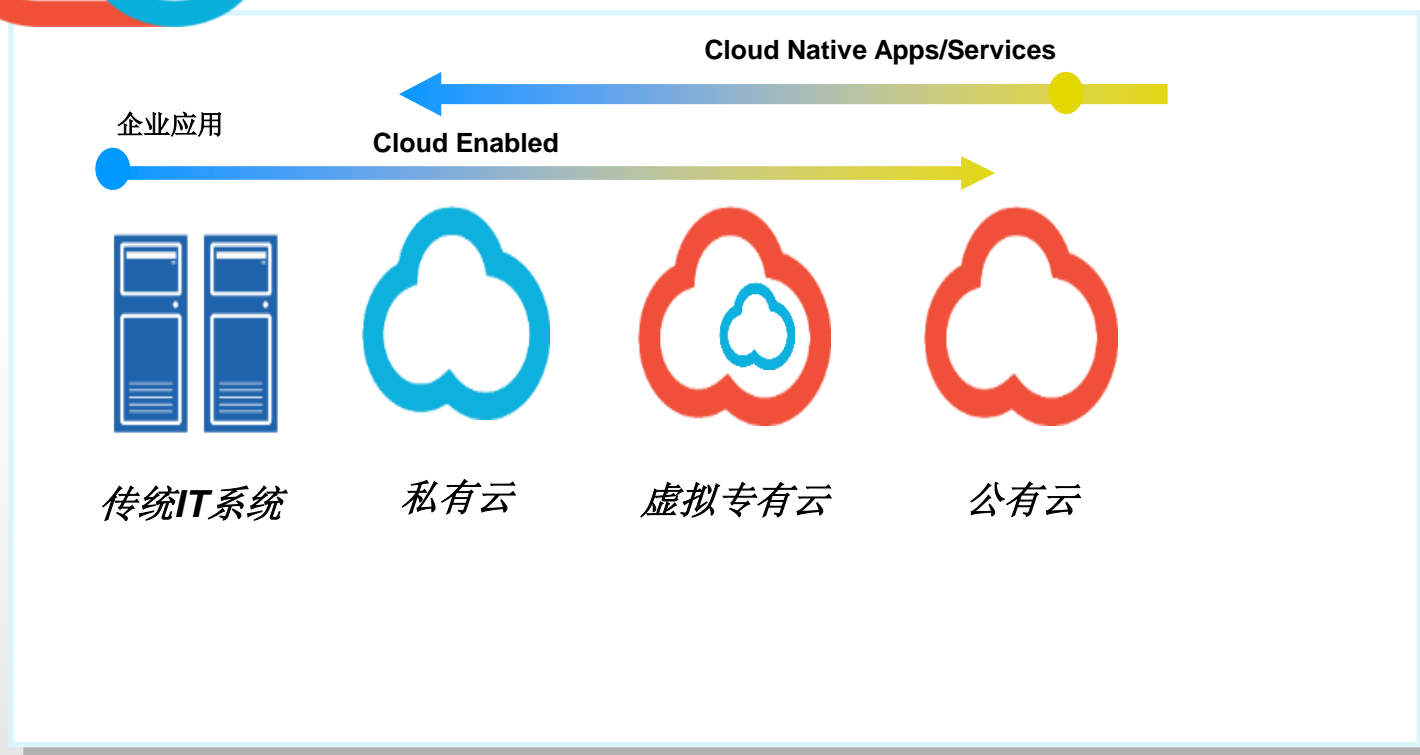
企业同时面临优化和创新的挑战



混合云为企业提供更多选择



混合云是对来自两个及其以上源的服务的安全消费，从而使用户或者客户可以根据业务需求进行灵活选择



主要内容

企业应用需求与混合云



基于**Docker**的容器服务实践

用于**Docker**的专家模式（**Pattern Engine**）

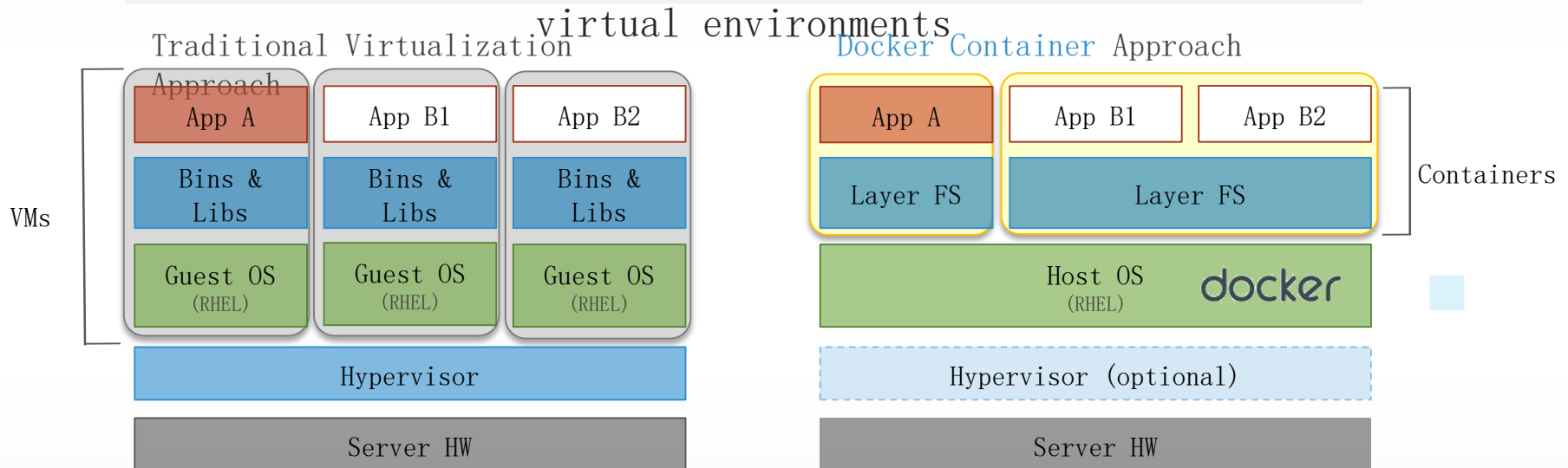
IBM 云平台Bluemix/Pure 中Docker服务介绍

企业应用特点

- 较复杂的拓扑
- 应用多，很多来自外包
- 大而单一的应用，较少采用微服务
- 使用相对传统的技术
- 采用商用中间件，对配置和调优有较强的依赖
- 较强的可靠性和安全性要求
- 需要简单易用的运维工具

容器技术与 Docker

Containers are a **lightweight** alternative to Virtual Machines for running software in **portable** and **isolated**



Attribute	VM	Container
Start-up time & Performance	Slow (minutes) HV overhead	Fast (seconds) no HV overhead
Footprint	Large (nothing shared)	Small (OS kernel shared)
Resource Constraints	Yes	Yes (CPU, Memory)
Isolation & Security	High	High
Portability	Low	High

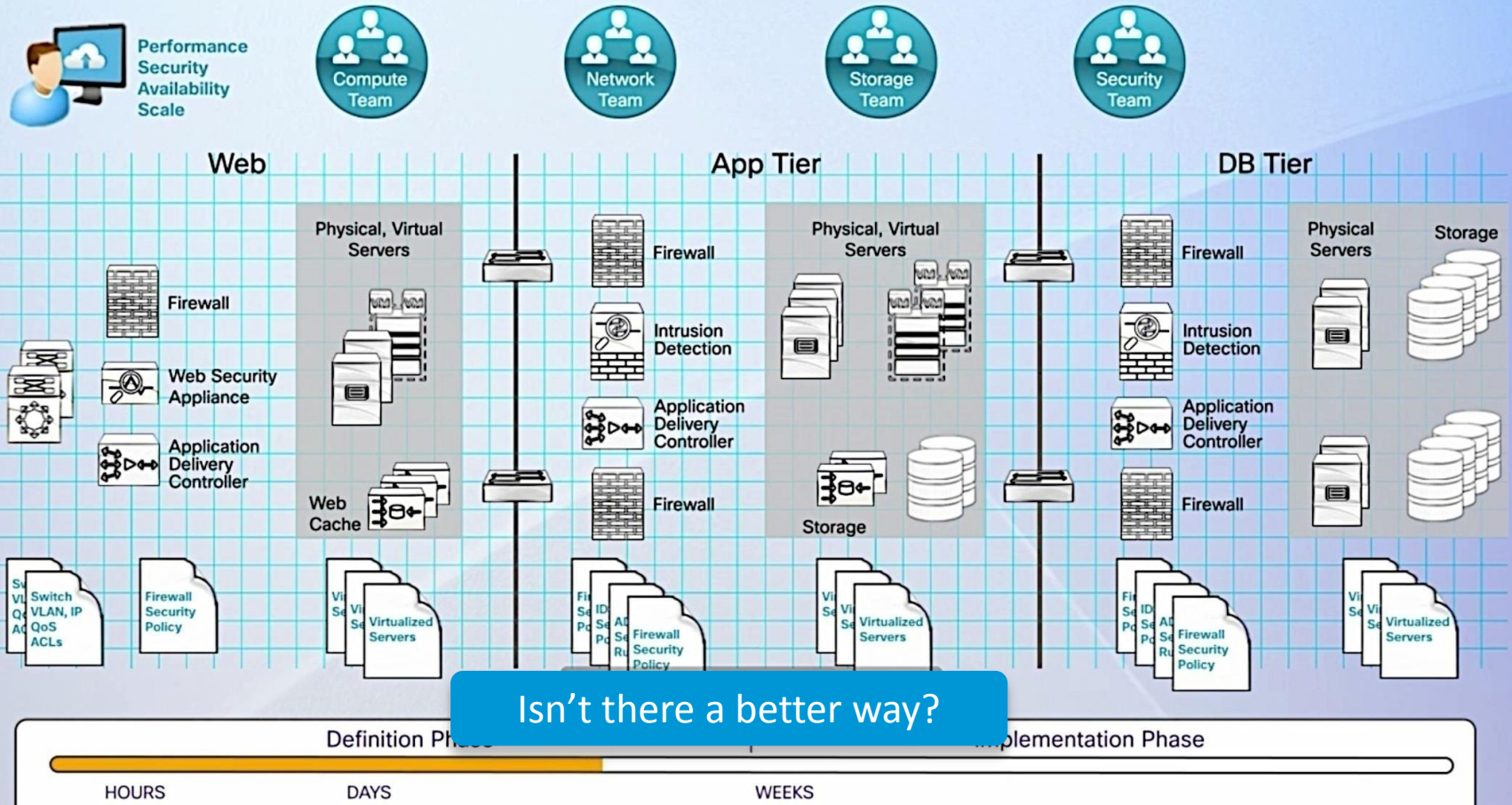
面向企业应用的混合云采用Docker的推动力

- 轻量
 - 有帮助，更多在于提高开发效率
- 性能
 - 依赖于工作负载
- 启动速度
 - 有帮助，更多在于提高开发效率
- 隔离与安全性
 - 不是优势
- 生态系统？
 - 有帮助，但不关键
- 标准化
 - 简化了CI/CD, 满足企业新应用创新需求
- 可移植性
 - 解决了遗留系统的迁移问题
- 灵活性
 - 解决了企业应用对特定中间件依赖和复杂配置问题

企业应用从部署到运维需要考虑很多因素

物理资源分配、部署、监控、安全、更新升级...

Defining the infrastructure can take an enormous amount of time



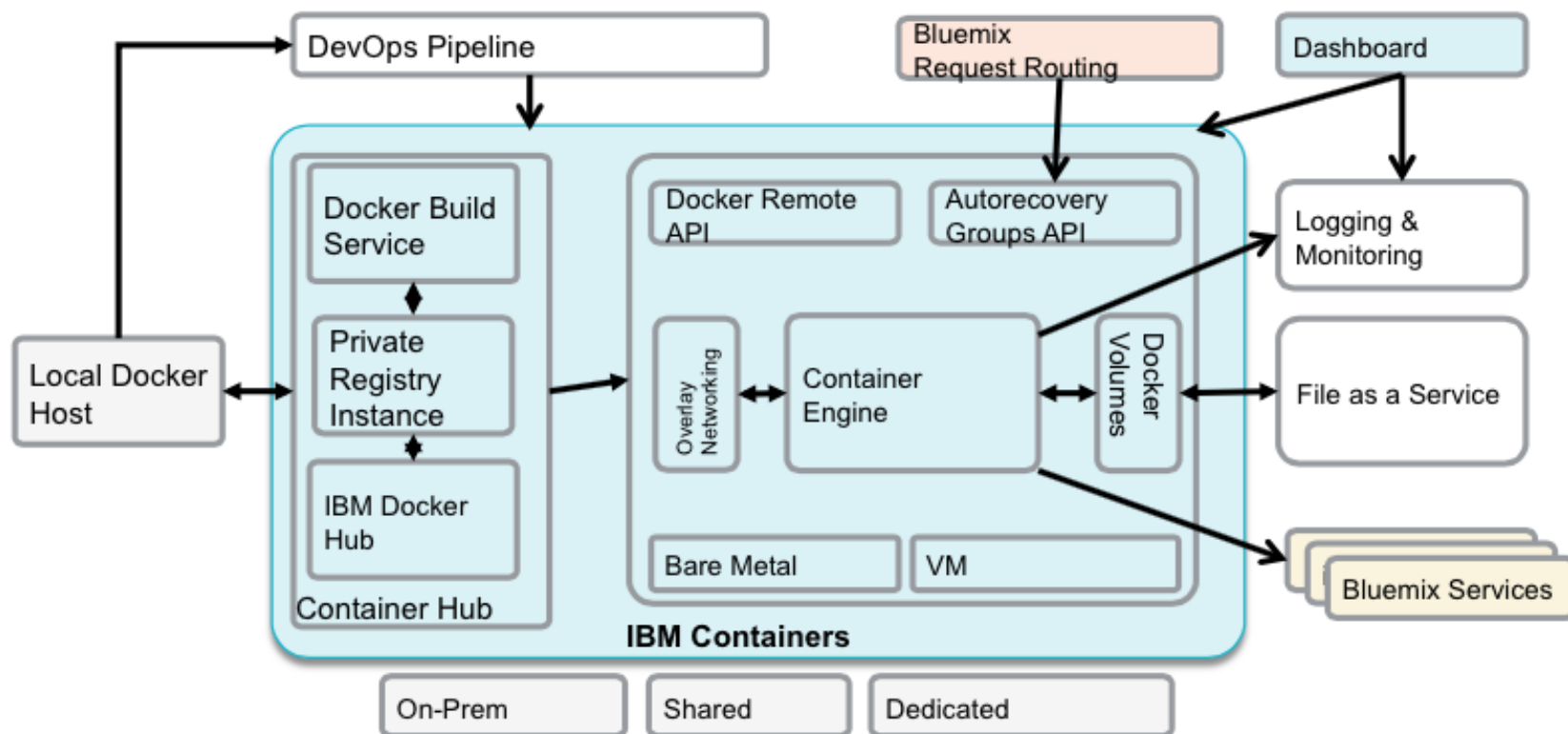
面向企业应用的云平台中容器服务设计要点（1）

- Container Hub
 - 支持私有以及公有 Docker Registry
 - 提供针对企业应用的镜像
- 完整的生命周期管理
 - 单个容器
 - 多个容器构成的组合应用
 - 容器运行时环境
- 容器服务
 - 提供单租户以及多租户选择
 - 基于物理机
 - 支持单主机多容器以及多主机多容器模式
 - 基于策略的资源管理 包括 placement, cleanup, movement

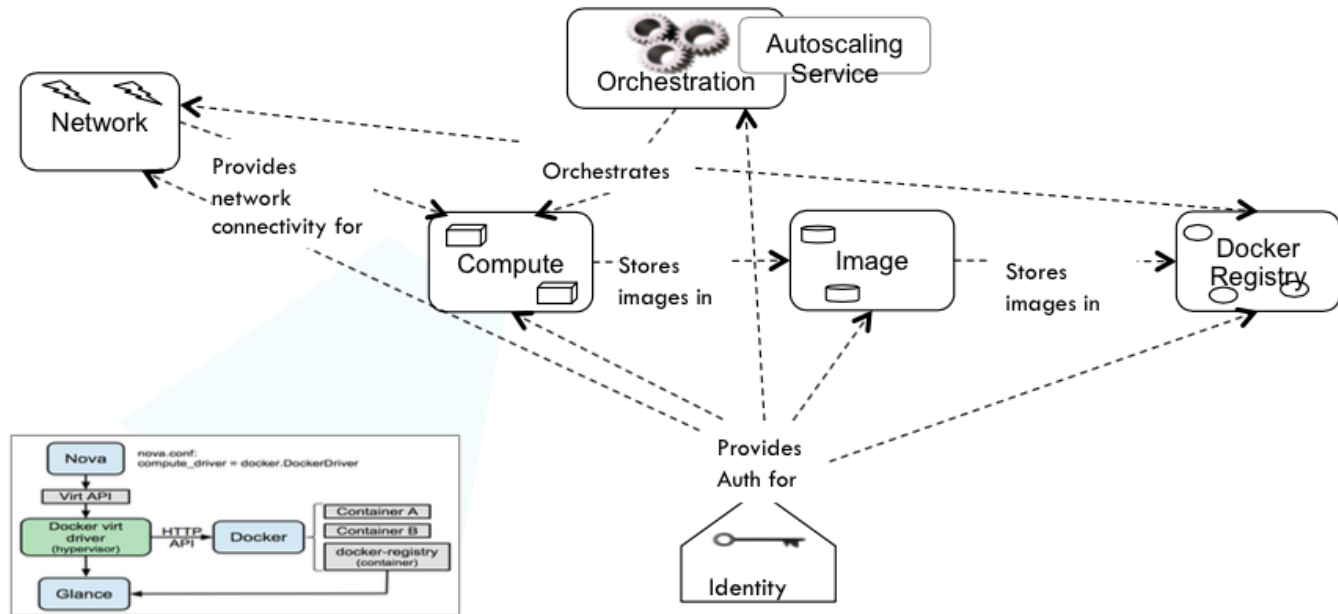
面向企业应用的云平台中容器服务设计要点（2）

- 网络
 - container-as-an-IP-host
 - 支持容器组间的private overlay network
 - 租户网络使用子网和安全组进行细粒度连接控制
- 存储
 - 支持容器在主机间移动时保持核心文件系统
 - 支持持久，非脆性的卷挂载（ persistent non-brittle volume attach ）
- 更多企业特性
 - 安全性，性能，可用性，可见性，企业内容
 - CI/CD 工具集成
- 与现有IaaS/PaaS平台以及传统系统的集成

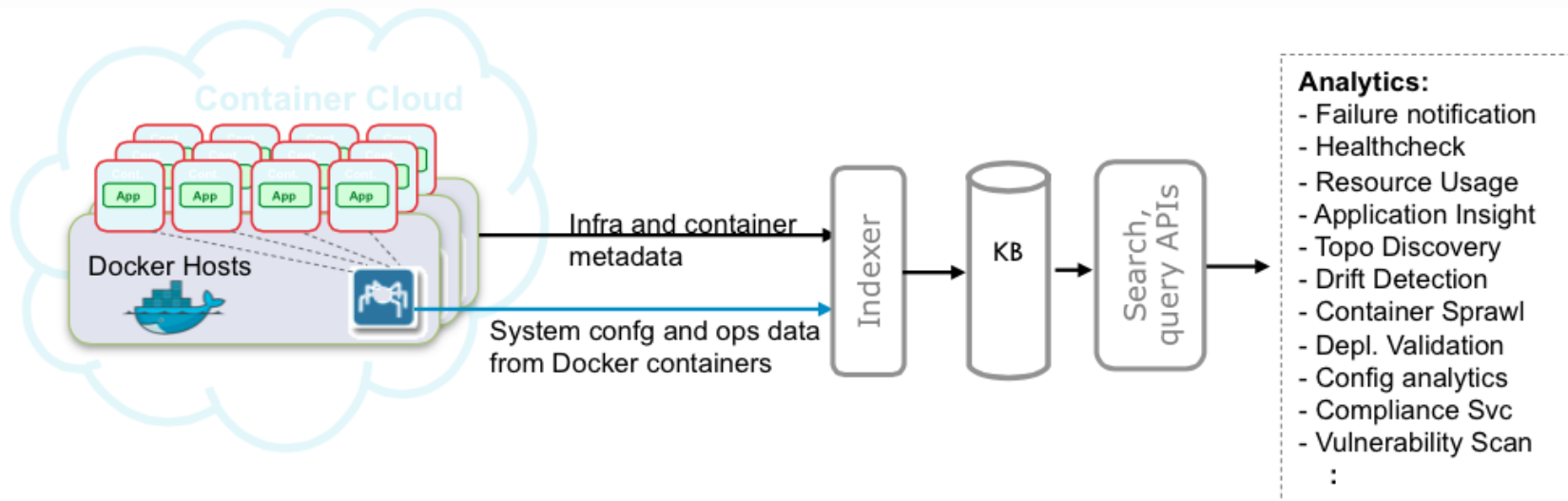
基于Docker的容器服务参考架构



使用OpenStack Docker driver部署和管理Docker容器（演化中）



Docker容器的监控

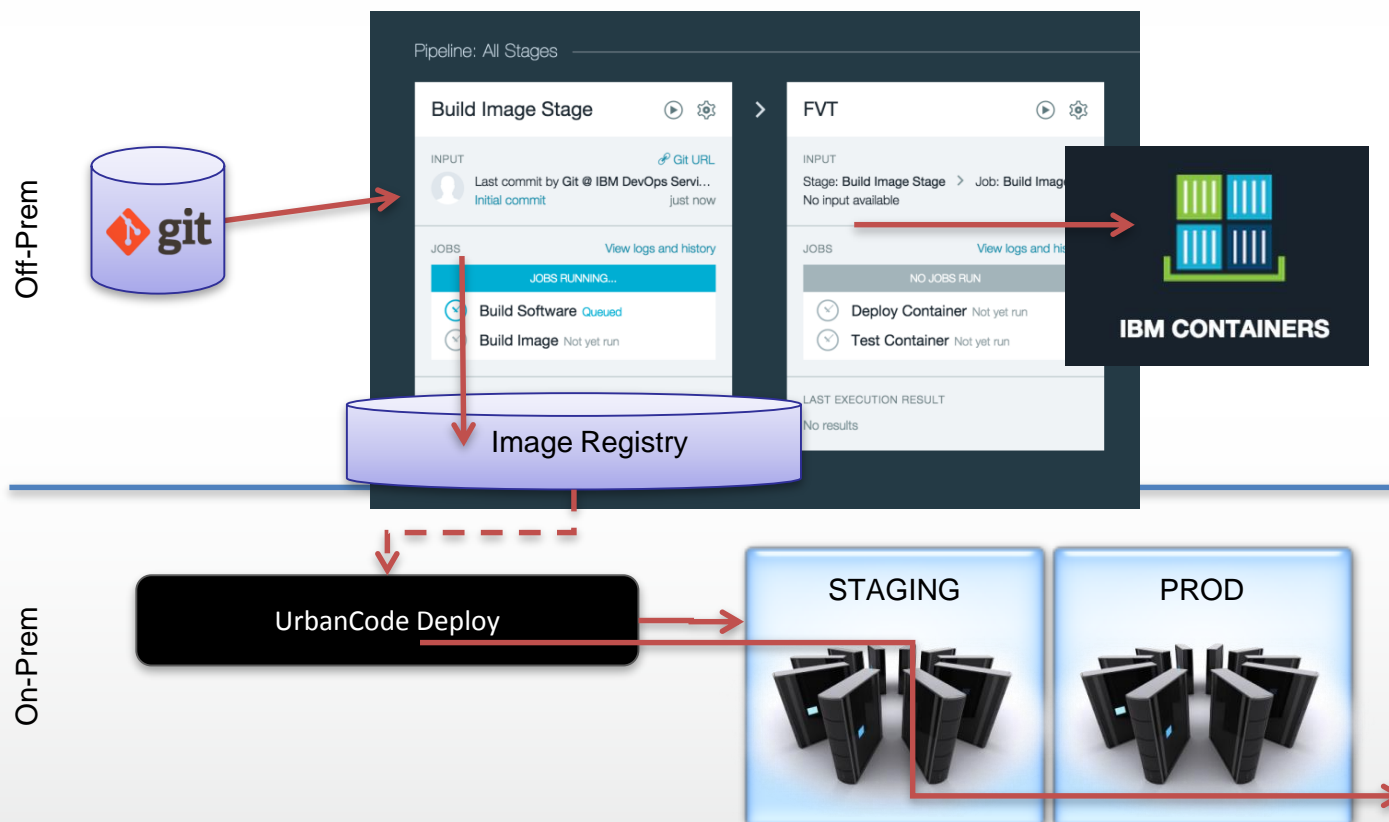


Our Approach:

- Seamlessly “crawl” the cloud like we crawl the web
- Query/mine the cloud like we query/mine the web



集成的CI/CD用于混合云环境Docker应用的发布



主要内容

企业应用需求与混合云

基于**Docker**的容器服务实践



用于**Docker**的专家模式（**Pattern Engine**）

IBM 云平台**Bluemix/Pure** 中**Docker**服务介绍

Docker + Patterns



Enterprise Strength Docker

1. Build, deploy and run [Patterns with Docker containers](#) for Cloud
2. Brings [Enterprise-grade lifecycle management](#) to Docker
3. Included private [Docker registry Pattern](#) deployable as a [shared service](#)

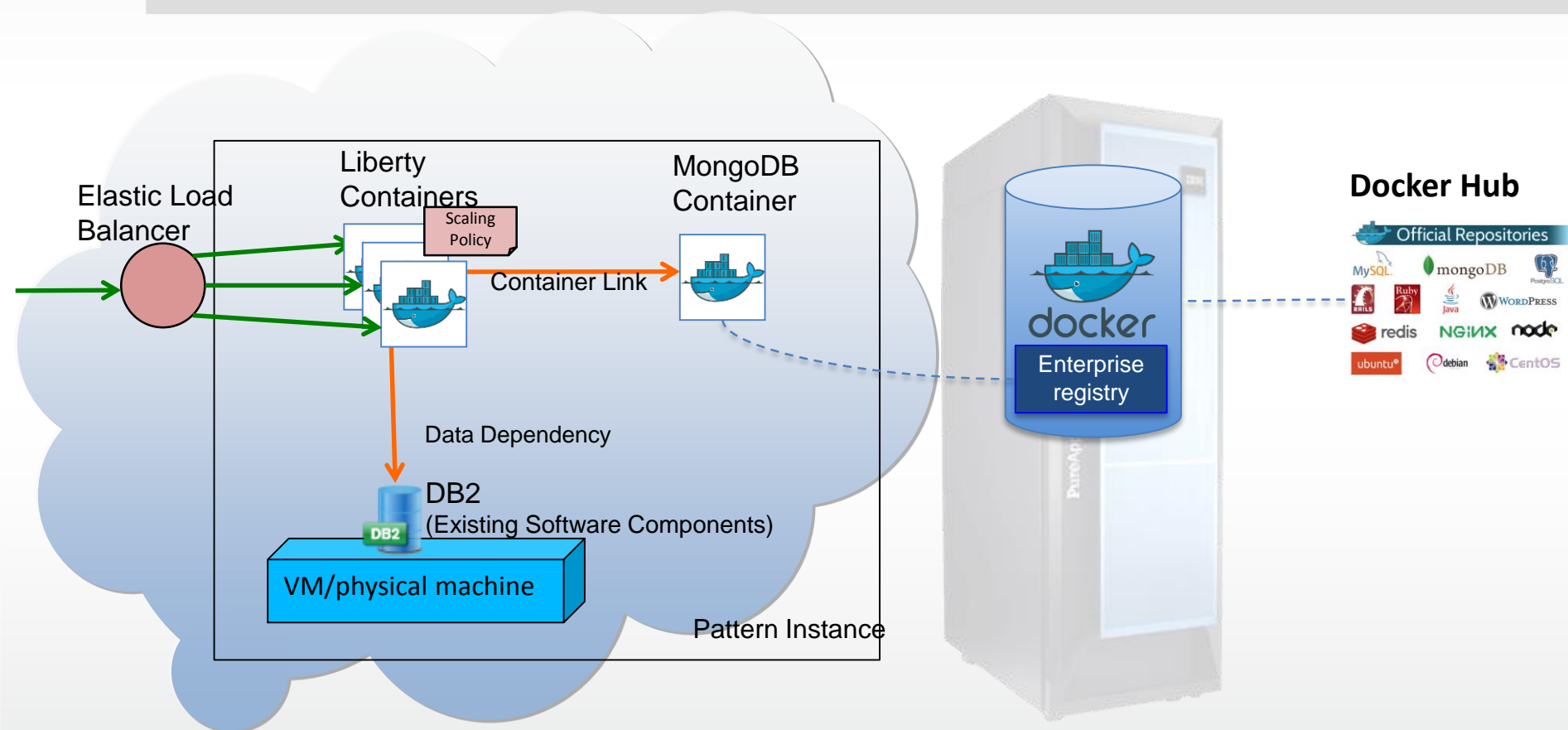
Improved Performance

- [Faster](#) application deployment, start-up and scaling
- [Higher density](#) deployments
- More [efficient](#) use of hardware

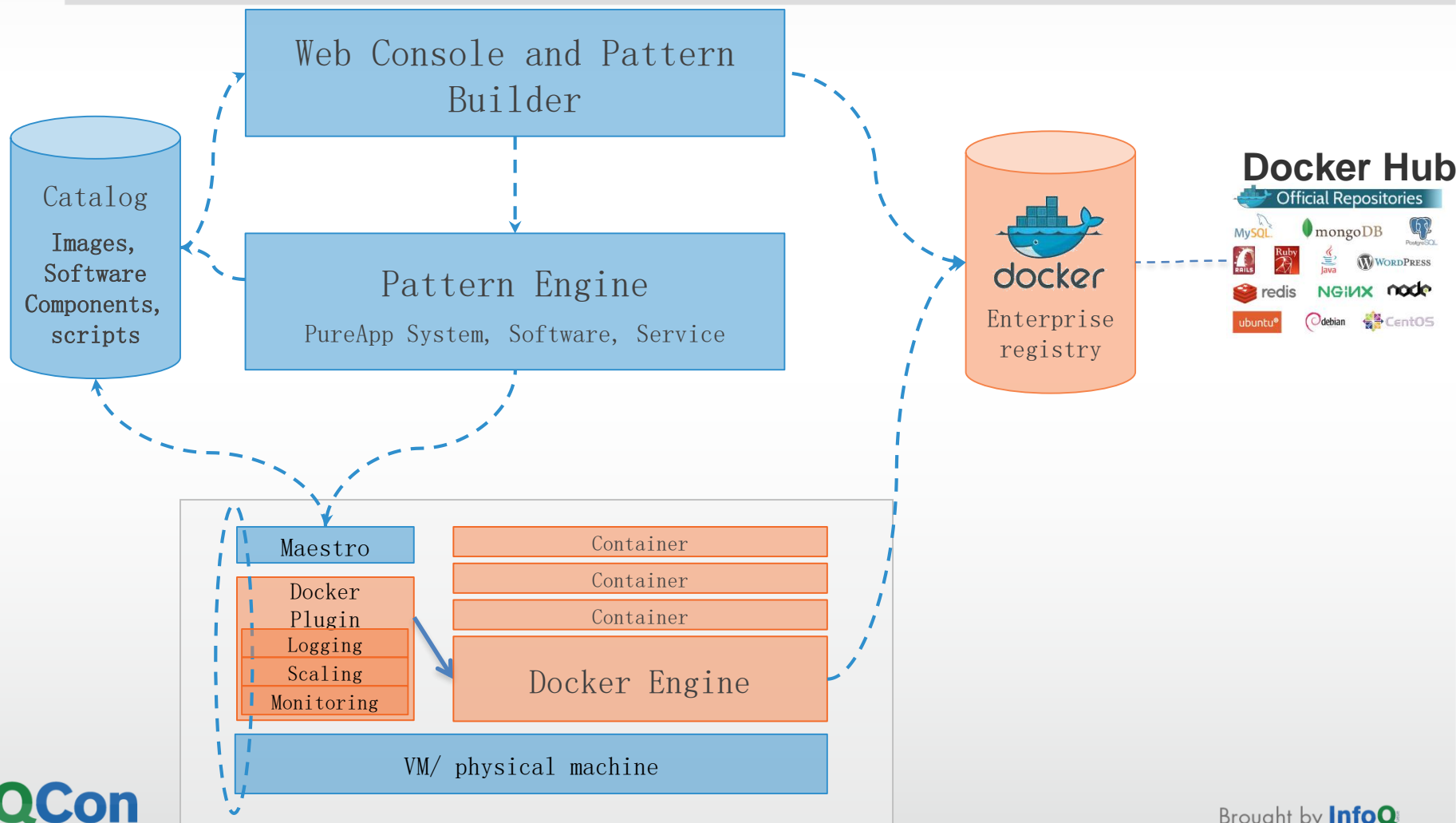
Portability, Hybrid Cloud, Open ecosystem, Productivity

- More [seamless](#) workload movement in [hybrid & borderless cloud](#) scenarios
- Access thousands of [pre-built applications](#) on DockerHub
- [Simplified](#) configuration management
- More [consistency](#) across development, test and production

使用Pattern来编排容器，软件部件和服务

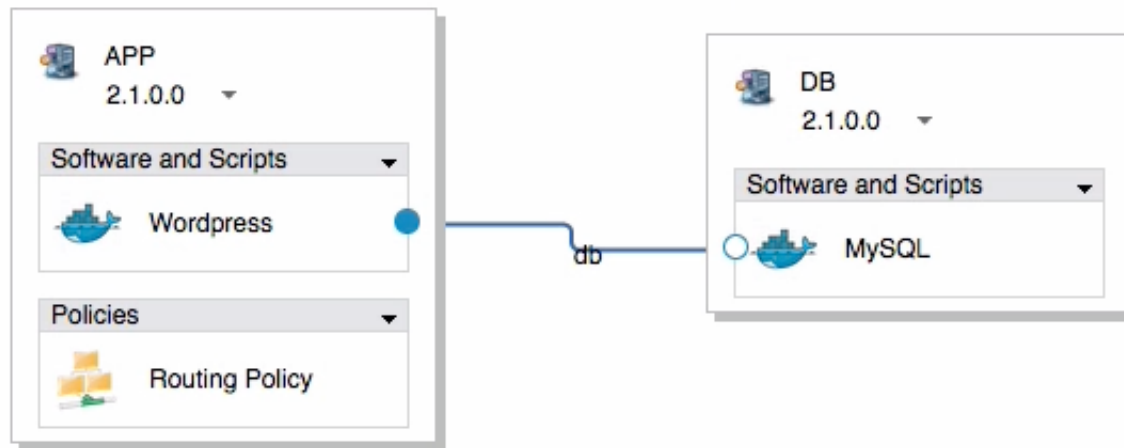


支持Docker的Pattern 架构



使用Pattern 来部署和管理容器化的应用

Pattern of Containerized “WordPress”



Value:

Docker content gets access to Pure's enterprise-grade lifecycle
Patterns gain ability to run 40,000+ dockerized apps

使用Pattern 部署和管理混合应用

IBM Pattern Builder

Pattern Name: demo_libertyContainer DB2 | Pattern Type: Virtual System Pattern Type 1.0

COMPONENTS

Component name

DEBUG

Debug component

IMAGES

SCRIPTS

SOFTWARE COMPONENTS

OTHER COMPONENTS

1

2

OS Node_1

2.1.1.0

Software and Scripts

Liberty Container

OS Node

2.1.1.0

Software and Scripts

Database Server

CreateDB_OLTP

1.2.0.0

TradeLiteDB2TableCreate

1.0.0

Add-Ons

Default add disk

1.0.0

Default add disk_1

1.0.0

1 Compute node w/ Docker container, containing Liberty, connected to...

2 Compute node w/ DB2 Pattern, and several script packages

主要内容

企业应用需求与混合云

基于**Docker**的容器服务实践

用于**Docker**的专家模式（**Pattern Engine**）



IBM 云平台Bluemix/Pure 中Docker服务介绍

IBM Bluemix提供不同的选择来帮助构建混合云

按您的方式构建应用程序。

使用多项最前沿的开放式源代码计算技术来增强应用程序。之后，可将其他工作留给 Bluemix 处理。

即时运行时

基于 Cloud Foundry 的以应用程序为中心的运行时环境。



了解 Bluemix 上的运行时

容器

以可移植且一致的方式交付您的应用程序，而无需管理操作系统。



了解 Bluemix 上的容器

虚拟机

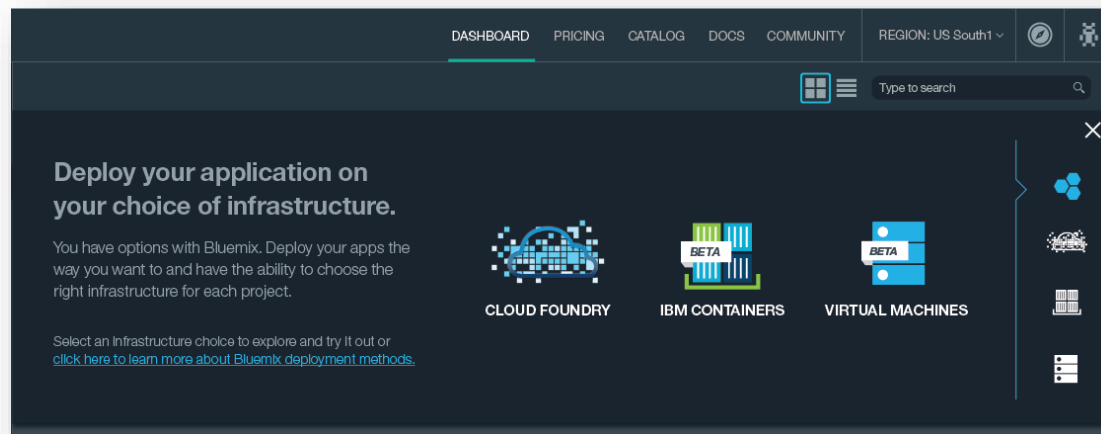
通过 VM 最大限度地灵活使用和控制您的环境。



POWERED

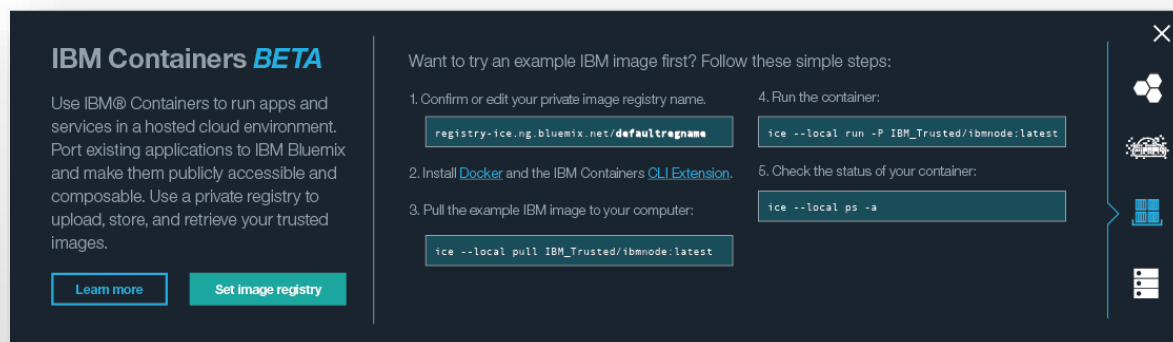
了解 Bluemix 上的 VM

IBM Bluemix 中基于Docker的容器服务




- ✓ 自动构建Docker 镜像
- ✓ 在私有Registry里管理和发布Docker 镜像

- ✓ 内嵌的自动扩展和恢复能力



- ✓ 内嵌的监控和日志能力
- ✓ 与Cloud Foundry的集成

IBM Bluemix 容器创建



Create A Container

Your Image Registry URL:
`registry-ice.ng.bluemix.net/gosiaregistry`

Select an image from your registry:

Name	Tag	ID	Created
ibm/liberty	latest	dc9b60d3-2c48-44d6-a2d7-272ecdfb31ae	3 days ago
ibm/node	latest	7fd09f11-c834-41cb-945a-2c32b119f37f	3 days ago
gosiaregistry/hellonode	latest	1120db94-cd2c-4062-9f83-70229a33bb37	2 weeks ago
gosiaregistry/ubuntu	latest	a2558596-8407-436b-ac41-a63c68573687	2 weeks ago
gosiaregistry/webapp	latest	2dd13e5d-45ec-4bdc-95e4-fdec34e6f4e4	2 weeks ago

Container Sizes Available

The free Beta plan for Containers includes 2 GB of memory and up to 2 Public IP Addresses.

Size	Memory	VCPUs	Disk Usage
m1.tiny	256	1 CPU	1 GB
m1.small	512	2 CPU	2 GB
m1.medium	1024	4 CPU	10 GB
m1.large	2048	8 CPU	10 GB

Container Settings

Name your container or group:

Selected image:

Container size:

m1.tiny

Scale and Network Settings

Deployment method:

Deploy as a single container

Public IP Address: (2 of 2 requested)

All of your public IP addresses are currently bound to other containers. To make them available, you must first unbind them. [View the docs](#) for more information.

Public Ports: (Separate ports by commas)

example: 80,443

Optional SSH Key:

[VIEW DOCS](#) [TERMS](#)

命令行工具 ICE

```
> ice --help
usage: ice [-h] [--verbose] [--cloud | --local]

        {logs, ip, images, rmi, login, help, ps, pause, group, namespace, start, version,
build, rm, unpause, run, inspect, stop, volume, restart, info, search, route, login}


> ice run --help
usage: ice run [-h] [--name NAME] [--memory MEMORY] [--env ENV]
        [--publish PORT] [--volume VOL] [--bind APP] [--ssh SSHKEY]
        IMAGE [CMD [CMD ...]]

positional arguments:
  IMAGE                image to run
  CMD                  command & args passed to container to execute

optional arguments:
  -h, --help            show this help message and exit
  --name NAME, -n NAME  assign a name to the container
  --memory MEMORY, -m MEMORY
                        memory limit in MB, default is 256
  --env ENV, -e ENV     set environment variable, ENV is key=value pair
  --publish PORT, -p PORT
                        expose PORT
  --volume VOL, -v VOL  mount volume, VOL is Volumeld:ContainerPath[:ro],
                        specifying ro makes the volume read-only instead of
                        the default read-write
  --bind APP, -b APP    bind to Bluemix app
  --ssh SSHKEY, -k SSHKEY
                        ssh key to be injected in container
```


- ✓ Docker CLI 扩展
- ✓ 调用Remote Docker API
- ✓ 支持多租户

Bluemix 容器仪表盘



mytrade

Routes: N/A | Private IP: 172.16.38.91 | Public IP: 129.41.252.17 | Ports: N/A
Volumes: N/A | Image: gosiarestory/trade:latest



GOSIAREGIST...
SIZE: tiny


INSTANCES:
1

MEMORY
256 MB

CPU
1

DISK
1

MEMORY UTILIZATION



Memory Usage Over Time (MB)

Memory
Used
204.04 MB

CONTAINER HEALTH

✓ Your container is running

STOP PAUSE

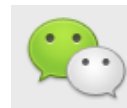
RESTART

Bind Bluemix Services from your existing Cloud Foundry Apps at Launch. View [the docs](#) to learn more about service binding to containers.

谢谢！



@InfoQ



infoqchina

软件
正在改变世界!