# Docker 深入解读

## 容器技术生态系统

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 容器技术生态系统 | | | | | | | | | | | | |
| 安全 | 镜像库 | 托管服务 | 容器化应用支撑平台 | 服务发现 | 配置管理工具 | 监控 | 存储 | 网络 | 标准化组织 | 容器运行时 | 容器操作系统 | 编排/调度 |
| Notary | Docker hub | Tutum | deis | etcd | Puppt | Cavisor+heapster | flocker | Docker networking | oci | runc | coreos | kubernetes |
| clair | Docker registry | softlayer | fig | consul | chef | prometheus | torus | weave | cncf | rkt | Project atomic | Compose+swarm+machine |
| Intel clear containers | Quay.io | Google container engine | flynn | zookeeper | Slat |  |  | flannel |  | warden | Canonical snappy | Mesos+marathon+chonos |
|  | harbor | Aws | lattice |  | ansible |  |  | cni |  |  | Vmware photon |  |
|  |  | azuer | Cloud foundry |  |  |  |  | calico |  |  |  |  |
|  |  | Opentstack | openshift |  |  |  |  | pipework |  |  |  |  |

## **容器技术带来的好处**

1：持续部署与测试

容器消除了线上线下的环境差异，保证了应用生命周期的环境一致性和标准化。开发人员使用镜像实现开发环境的构建，开发完成后通过封装着完整环境和应用的镜像进行迁移，由此，测试和运维人员可以直接部署软件镜像进行测试和发布，大大简化持续集成、测试和发布的过程。

2：跨云平台支持

容器带来的最大的好得就是其造配性。

3：环境标准化和版本控制

4：高资源利用率与隔离

5：容器跨平台性与镜像

6：易于理解且易用

7：应用镜像仓库

## **：docker 基础**

Docker 安装

Docker 操作相关命令

参数解读

因为国内电商，普遍使用redhat 和 centos ，我只使用centos 7来作为示例

1：docker的安装

安装docker 的基本要求：

* Docker 只支持64位的cpu架构的计算机，目前不支持32位的cpu

|  |
| --- |
|  |

* 建议内核版本在3.10以上版本
* liunx内核需要开启cgoups和namespace功能
* 对于非liunx内核的平台，如windows 和os x 需要安装boot2docker 工具。

2：使用主流的操作系统进行安装

使用rehl/centos系统中安装docker

|  |
| --- |
| sudo yum -y install docker-engine |
| sudo systemctl start docker |
| 安装完成后的验证 |
| docker run hello-world |
| *Hello from Docker!* |

使用windows系统中安培训docker