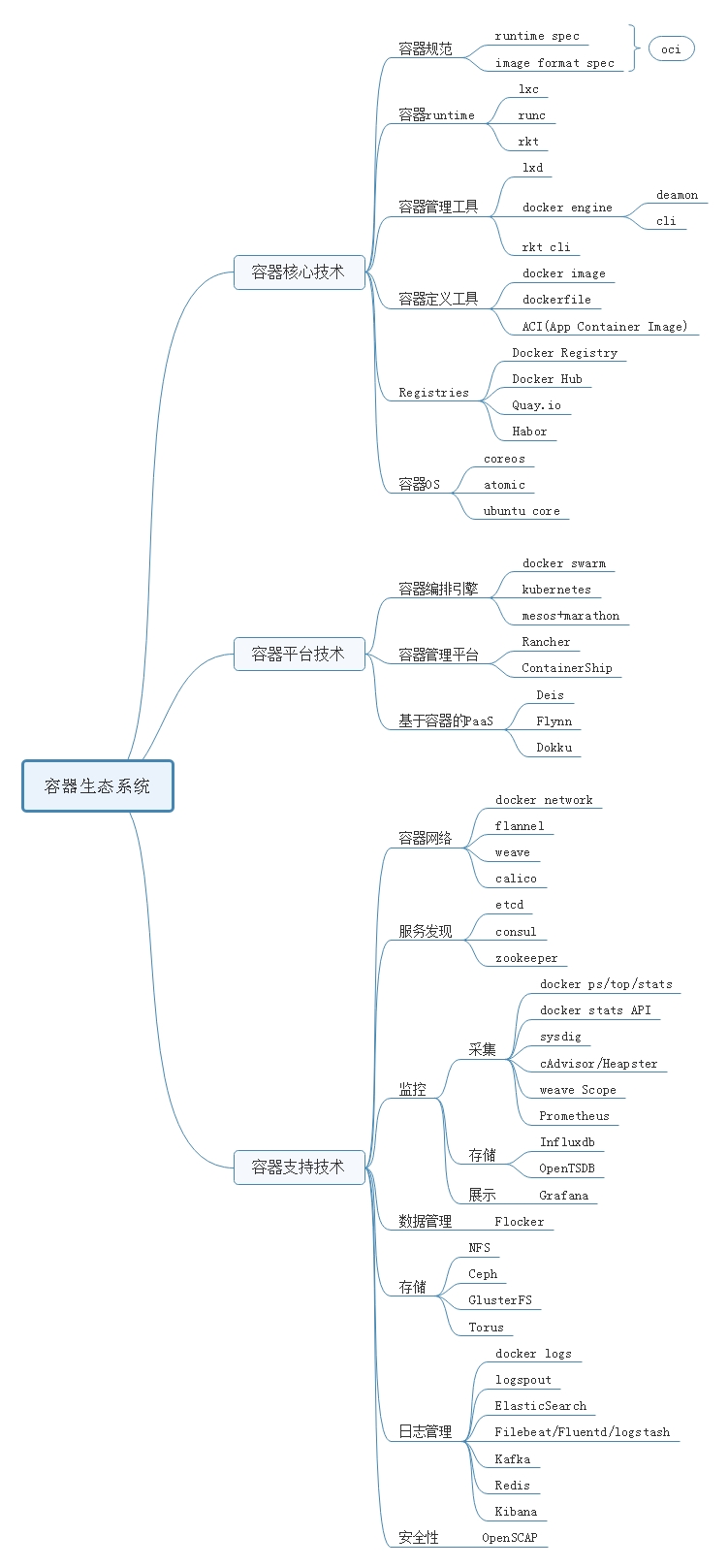
**容器生态系统**

容器技术是继大数据和云计算之后很火的一门技术，具有快速弹性伸缩（根据业务处于峰值或低峰状态实现容器数量的扩缩容，高效合理利用资源）、敏捷开发快速上线（平台提供持续集成/持续交付的工具链，使开发者只需专注于业务的开发与测试）、微服务架构应用（支持容器粒度的隔离，保证应用每个微服务的安全和稳定）等优点，在未来很长时间内都将很热门。

由于容器技术属平台级别的技术，覆盖范围很广，包含了计算、网络、存储、日志、监控、安全、高可用等很多方面，因而在了解并学习掌握容器技术之前，对整个容器的生态系统有所个整体的了解是很必要的，这也是的本文的目的所在--带你了解容器的生态系统。容器技术的发展带动了一个生态系统的发展，而容器生态系统的发展又促进了容器技术的迅速发展。下图给出的是容器生态系统的概览图：

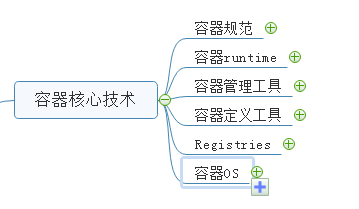


图：容器生态系统

接下来，对容器的生态系统从容器核心技术、容器平台技术和容器支持技术3个方面进行介绍。

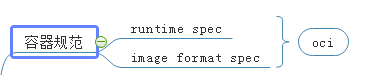
1. 容器核心技术

容器的核心技术是指让容器（container）在主机（host）上运行起来的技术，包括容器规范、容器runtime、容器管理工具、容器定义工具、Registry和容器OS。



1. 容器规范

为了使目前存在的多种容器，像OpenVZ、rkt、Docker等，能相互兼容，Docker、CoreOS、Google等公司成立了Open Container Initiative（OCI）的组织来制定容器规范，目前已发布了两个规范：runtime spec和image format spec。



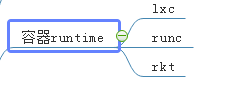
2.容器runtime

容器runtime是容器运行的地方，与操作系统的kernel一起为容器提供运行的环境。目前主流有lxc、runc、rkt3种运行环境。

lxc是Linux提供的容器runtime，Docker早期便是以它作为runtime；

runc是Docker自己开发的容器runtime，目前是docker的默认runtime；

rkt是coreos开发的容器runtime，符合oci规范，也能够运行Docker容器。



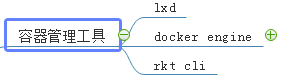
3.容器管理工具

容器管理工具用于容器的管理，它们对内与runtime 进行交互，对外为用户提供接口，比如CLI（command line interface），使用户通过这些接口对容器进行管理。目前主流的管理工具是：lxd，docker engine，rkt cli。

lxd是lxc对应的管理工具；

docker engine是runc对应的管理工具，包含后台deamon和cli；

rkt cli是rkt对应的管理工具。



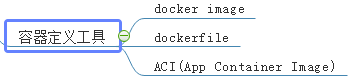
4.容器定义工具

容器定义工具用来定义容器的内容和属性，这样容器就能被保存、共享和重建。目前主流的容器定义工具有3种：docker image，dockerfile和ACI（App Container Image）。

docker image是docker容器的模板，runtime依据它来创建容器；

dockerfile是包含若干命令的文本文件，可以用它创建docker image；

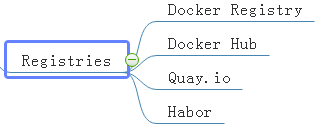
ACI（App Container Image）是由coreos开发的rkt 容器的image格式。



5.Registry

Registry是镜像仓库，是存储image的地方。

Docker Registry、Docker Hub（https://hub.docker.com）、Quay.io（https://quay.io/）是主流的Registry。Habor是VMware公司开源的DockerRegistry项目（https://github.com/vmware/harbor），其目标是帮助用户迅速搭建一个企业级的DockerRegistry服务，提供图形界面和权限控制。



6.容器OS

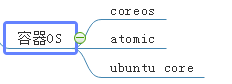
容器runtime使得容器可以在Linux、MAC OS、Windows上运行，而容器OS是专门运行容器的操作系统，它与常规的OS相比，体积更小，启动更快，运行容器效率更高。

目前主流的容器OS有CoreOS、atomic 和 ubuntu core 。

CoreOS：以容器为中心的操作系统，在配置管理、自动扩容、安全等方面有一套完整的工具。

Atomic：一个轻量级的操作系统，可以运行docker、kubernetes、rpm和systemd。

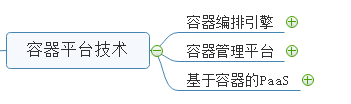
Ubuntu Core：适合运行容器集群的轻量级ubuntu操作系统。



1. 容器平台技术

容器核心技术能使容器在单个主机上运行，而容器平台技术使得容器作为集群在分布式环境中运行。

容器平台技术包括容器编排引擎、容器管理平台和基于容器的PaaS。



1. 容器编排引擎

容器的应用通常会采用微服务架构，把应用分为不同的组件，组件以服务的形式在容器中运行，对外暴露端口以提供服务。为了保证应用的高可用，每个组件会包含多个相同的容器。实际应用中，容器集群中的容器会根据业务需要进行扩容和缩容，动态的创建、迁移和销毁。

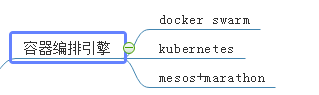
容器编排引擎就是对容器集群中的容器进行管理。‘编排’通常包含容器管理、调度、集群定义和服务发现等，通过容器编排引擎，容器有机的组合成微服务应用，满足业务需求。

当前主流的容器编排引擎是docker swarm、kubernetes和mesos。

docker swarm是Docker开发的容器编排引擎；

Kubernetes是Google领导开发的开源容器编排引擎，支持Docker和CoreOS的容器，是目前Docker容器最常使用的容器编排引擎；

Mesos是一个通用的集群资源调度平台，与marathon一起提供容器编排引擎功能。



1. 容器管理平台

容器管理平台抽象了编排引擎的底层实现细节，架构于容器编排引擎之上，且支持多种容器编排引擎，为用户提供更方便的功能，比如一键部署应用和application catalog等。

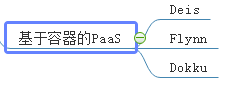
主流的容器管理平台是Rancher和ContainerShip。



1. 基于容器的PaaS

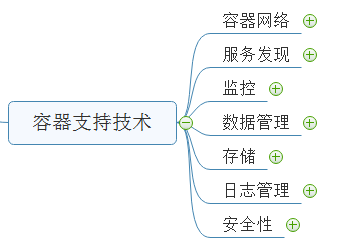
基于容器的PaaS提供开发、部署和管理应用的平台，使开发者专注于应用的开发，而不必关心底层的基础设施。

开源容器PaaS的代表有Deis、Flynn、Dokku。



1. 容器支持技术

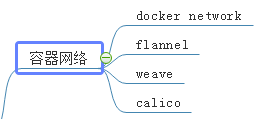
容器支持技术指的是支持基于容器的基础设施的技术，包括容器网络、服务发现、监控、数据管理、存储、日志管理和安全性。



1. 容器网络

容器网络拓扑比较复杂且是动态的，用户需要管理容器与容器，容器与其他实体之间的连通性与隔离性，因而产生了许多的容器网络解决方案。

Docker network是Docker原生的网络解决方案，我们也可采用第三方开源的网络方案，比如flannel、weave、calico（一个纯三层的网络解决方案，使用BGP协议进行路由，可以集成到openstack和docker）等，由于各个网络方案各有优缺点，实际应用中应根据需要进行选型。



1. 服务发现

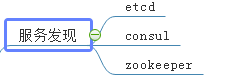
动态变化是微服务应用的一大特点，容器会根据业务需要创建、销毁和迁移。当负载增加时，会创建容器；当负载减少时，会销毁部分容器；容器也会根据主机资源的使用情况在不同的主机间迁移，这些过程中容器的IP和端口也会发生变化。在这样动态变化的环境中，需要让用户知道如何访问容器提供的服务，而这就是服务发现的工作。服务发现会保存容器集群中所有微服务的最新信息，比如IP和端口，并对外提供API，提供服务查询的功能。

服务发现典型的解决方案有：etcd、consul和zookeeper。

etcd：CoreOS开源的分布式key-value存储，通过HTTP协议提供服务，因此使用起来简单。但是etcd只是一个key-value存储，默认不支持服务发现，需要第三方工具来集成。Kubernetes默认就使用etcd作为存储。

consul：HashiCorp开源的服务发现和配置管理工具，自带服务发现特性（DNS Server）。它是强一致性的数据存储，使用gossip协议形成动态集群。

zookeeper：起源于Hadoop社区，优点是成熟、可靠、功能丰富，缺点是使用Java开发，配置麻烦。



1. 监控

容器的动态特征增加了监控的难度，针对容器环境，已出现了许多监控工具和方案。

1）采集

docker ps/top/stats是Docker原生的命令行监控工具，同时，Docker也提供了stats API使用户能通过HTTP请求获取容器的状态信息。

sysdig、cAdvisor/Heapster和Weave Scope是其他开源的容器监控方案。

Prometheus是一套开源的监控报警系统，它将所有信息都存储为时间序列数据，实时分析系统运行的状态、执行时间、调用次数等，以找到系统的热点，为性能优化提供依据。

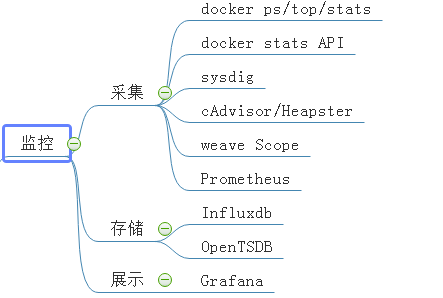
1. 存储

InfluxDB是一个开源的时序数据库，适用于处理和分析资源监控数据这种时序相关数据。

OpenTSDB是基于Hbase的分布式的、可伸缩的时间序列数据库。主要用于收集大规模集群的监控数据并进行存储，查询。

1. 展示

Grafana是一款可视化工具，大多使用在时序数据的监控方面，它的UI更加灵活，有丰富的插件，功能强大。



1. 数据管理

由于容器经常会在不同的host间迁移，如何保证持久化数据也能够动态迁移，就是Flocker这类数据管理工具提供的能力。



1. 存储

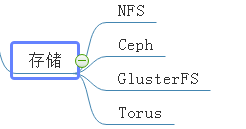
在容器集群中，运行服务需要将数据持久化保存，因而需要提供持久化存储解决方案。

NFS（Network File System）是网络文件系统，通过网络（一般是局域网）让不同的主机系统之间可以共享文件或目录。

Ceph是一个可靠的、自动重均衡、自动恢复的分布式存储系统，根据场景划分可以将Ceph分为三大块，分别是对象存储、块设备存储和文件系统服务。

GlusterFS是一个开源的分布式文件系统，具有强大的横向扩展能力，它借助TCP/IP或InfiniBandRDMA网络将物理分布的存储资源聚集在一起，使用单一全局命名空间来管理数据。

Torus是CoreOS专门为容器集群量身打造的开源分布式存储系统，可以为通过Kubernetes编排和管理的容器集群提供可靠可扩展的存储。



1. 日志管理

日志为问题的查找排查和事件管理提供了重要的依据。

Docker logs是Docker原生的日志工具，而logspout对日志提供了路由功能，能收集不同容器的日志并转发给其他工具进行处理。

ElasticSearch是一个分布式可扩展的实时搜索和分析引擎，是建立在全文搜索引擎Apache Lucene(TM)基础上的搜索引擎。

Filebeat是本地文件的日志数据采集器，在服务器上安装客户端后，Filebeat会监控日志目录或者指定的日志文件，追踪读取这些文件，并且转发这些信息到ElasticSearch或者logstash中存储。

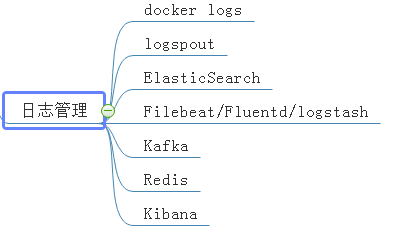
Fluentd是一个开源的数据收集器，有许多的插件，可以连接各种数据源和数据输出组件。Fluentd负责收集容器日志，然后发送给ElasticSearch。

Logstash是一个开源的数据收集引擎，具备实时数据传输的能力，可以统一过滤来自不同源的数据，并按照开发者的制定规范输出到目的地。

Kafka是Apache提供的一个快速、可扩展的、高吞吐、可容错的分布式发布订阅消息系统。Kafka具有高吞吐量、内置分区、支持数据副本和容错的特性，适合在大规模消息处理场景中使用。

Redis是一种支持Key-Value等多种数据结构的存储系统，可用于缓存、事件发布或订阅、高速队列等场景。该数据库使用ANSI C语言编写，支持网络，提供字符串、哈希、列表、队列、集合结构直接存取，基于内存，可持久化。

Kibana是一个开源的分析和可视化平台，可搜索查看，并和存储在ElasticSearch索引中的数据进行交互。它能执行高级数据分析，并且以各种图标、表格和地图的形式可视化数据。



1. 安全性

容器安全性保证容器的安全，不被攻击，OpenSCAP能够对容器镜像进行扫描，发现潜在的漏洞。



以上就是容器生态系统所包含的核心技术、平台技术和支持技术，涉及容器运行、编排、管理、运维等相关技术，实在是比较多，所以可能存在疏漏，但能让大家对整个容器生态系统及其常用解决方案有大致了解，便于更深入的研究学习容器云技术。另外，其中涉及的很多技术都是开源的，大家可以很好的加以利用。