


Kubernetes探秘-多master节点容错部署

Kubernetes的master服务主要包括etcd(数据存储)、control-manager(控制器)、scheduler(调度器)、apiserver(服务接口)，我们将其部署到多节点实现容错。在《[Kubernetes探秘-etcd节点和实例扩容](#)》中，已经将etcd服务扩展到多个节点。这里我们将control-manager(控制器)、scheduler(调度器)、apiserver(服务接口)扩展到多个节点运行。

GYC 2018

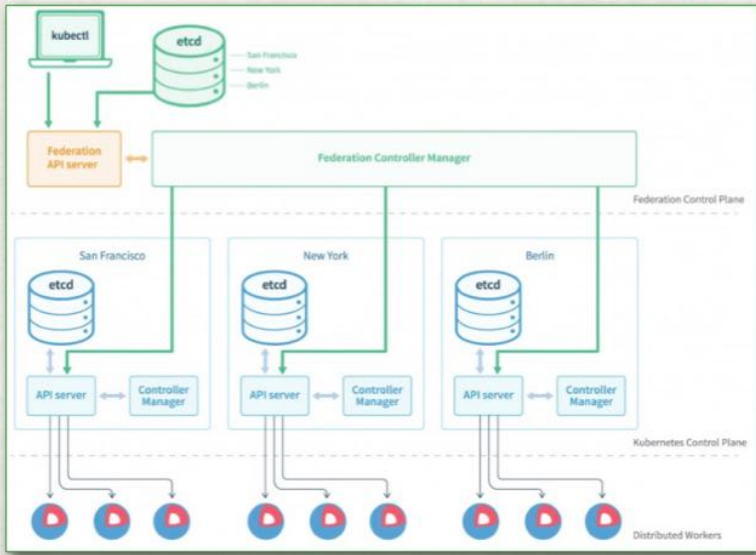
高级：集群联邦、高可用、备份与还原，云应用市场



Kubernetes联邦

- ① K8s支持Federation(联邦)管理，多个Kubernetes集群组成K8s联邦。
- ② K8s联邦与单个集群管理方式一样，将配置信息合并，即可合并进行管理。
 - 联邦管理接口和配置命令与集群兼容，只是增加了一些联邦管理的命令。
- ③ K8s联邦支持跨集群的Pod负载分配和故障转移。
- ④ K8s联邦支持跨集群的DNS，跨集群服务发现。

→ 目前，K8s联邦还在Beta阶段，管理还比较复杂，稳定性还有待测试。



A. Kubernetes集群系统的高可用

- ① etcd的多活
 - etcd是K8s集群状态存储服务。
 - etcd支持分布式存储和分布式一致性。
 - 部署etcd集群，把K8s的etcd参数指向etcd集群即可。
- ② Master主节点的多活
 - 实现下列服务的多实例运行：
 - kube-apiserver
 - kube-scheduler
 - kube-controller-manager
 - 需要专门配置，使用外部工具。
 - 使用HAProxy、Keep-alived等。
- ③ 工作节点故障恢复
 - 工作节点由主节点自动管理，支持多个副本。
 - 出现故障后负载自动转移到其它节点，Pod将会自动重建（内部状态会丢失）。

B. Kubernetes应用的高可用

- ① 设计Pod应该为无状态，以便随时转移。
- ② 创建Deployment，指定ReplicaSet的数量。
- ③ 运行时自动分配Pod到可用节点。
- ④ 可以按需重新设定Pod副本数量。

C. 备份与还原

- ① 镜像仓库，registry。
- ② 主控节点状态，etcd存储。
- ③ 应用配置文件，统一管理yaml文件。
- ④ 应用数据存储，统一备份网络存储。

D. 云应用市场

by wangzhi@supermap.com, 2018.05

1、多master节点部署

control-manager(控制器)和scheduler(调度器)通过apiserver(服务接口)来进行集群容器实例的管理，通过向apiserver中的Endpoint加锁的方式来进行leader election，当目前拿到leader的实例无法正常工作时，别的实例会拿到锁，变为新的leader。缺省情况下election=true，默认安装即支持多实例的自动选主。

- 说明：各个节点的kubelet会自动启动/etc/kubernetes/manifest/*.yaml里定义的pod，作为static pod运

行。

首先，使用kubeadm部署第一个主节点。

第二步，安装副节点。

- 使用kubeadm部署多个副节点。
- 或者先用kubeadm join部署为工作节点。
- 然后将/etc/kubernetes/manifest下的文件复制到各个副节点对应目录，以及上级对应的*.conf文件。文件包括：
 - etcd.yaml （之前已经修改，不能覆盖）
 - kube-apiserver.yaml
 - kube-controller-manager.yaml
 - kube-scheduler.yaml
- 重启kubelet服务，运行命令：systemctl restart kubelet。
- 此时，在Kubernetes的Dashboard中可以看到上面的pod，但是不能进行删除等操作。

2、apiserver的负载均衡

通过上面的方法设置多个master服务后，kube-apiserver的URL主地址全部指向的是第一个master节点IP地址，仍然存在单点失效的风险。为了实现多点容错，有几种方案（原理都是一样的，只是实现方式不同）：

第一种，外部负载均衡器。

使用外部的负载均衡器分配的高可用IP作为apiserver的服务地址，所有的外部访问以及scheduler.conf、controller-manager.conf中的server参数均指向该地址，然后将该地址映射到具体的内部服务器IP上，由外部负载均衡器来分配访问负载。

- 在上面的各master节点上使用高可用IP作为服务地址，如--
apiserver-advertise-address=10.1.1.201。
 - 参考：[多网卡Ubuntu服务器安装Kubernetes](#)
- 把副节点的IP地址加入负载均衡器。
- 将所有节点的scheduler.conf、controller-manager.conf中的server参数指向该高可用IP。

这种方式部署较为简单，但依赖云服务商提供的负载均衡器。

如果自己安装负载均衡器设备或软件，需要确保其本身是高可用的。

第二种，虚拟IP+负载均衡。

使用keepalived实现虚拟IP，主节点不可用时将IP自动漂移到其它节点，工作节点基本不受影响。k8s集群按照虚拟IP进行配置，与第一种方案类似，但通过简单的软件即可实现k8s集群主节点的容错。

虚拟IP（实际上是直接修改真实IP）每一时刻只运行于单个节点上。因此，其它的副节点上的apiserver服务处于standby模式。

通过添加HAProxy等做apiserver的负载均衡，之上再用keepalived做多节点的虚拟IP，可以将多节点变为支持负载均衡的互备模式。

- 在每一个副节点运行keepalived，配置为同一组和IP地址加入负载均衡器。
- 将所有节点的scheduler.conf、controller-manager.conf中的server参数指向该高可用IP。
- 注意，kubeadm安装的kubernetes证书只能支持本机单节点授权。这种模式可能需要更换新的授权证书。

第三种，多主分治+反向代理。

每个节点单独运行，通过etcd共享数据。

- 各个节点的scheduler.conf、controller-manager.conf的server参数指向本地apiserver。
- 部署nginx做反向代理，外部访问通过反向代理服务分发到各个apiserver。
- 各个节点完全自治，授权证书也不相同，需要反向代理进行处理。
- 反向代理应该是高可用的，与第一种方式类似。

3、Kube-dns高可用

kube-dns并不算是Master组件的一部分，可以跑在Node节点上，并用Service向集群内部提供服务。但在实际环境中，由于默认配置只运行了一份kube-dns实例，在其升级或是所在节点当机时，会出现集群内部dns服务不可用的情况，严重时会影响线上服务的正常运行。

为了避免故障，请将kube-dns的replicas值设为2或者更多，并用anti-affinity将他们部署在不同的Node节点上。这项操作比较容易被疏忽，直到出现故障时才发现原来是kube-dns只运行了一份实例导致的故障。

更多参考

- [Kubernetes 1.13.1的etcd集群扩容实战技巧](#)
- [Kubernetes的etcd数据查看和迁移](#)
- [etcd集群备份和数据恢复](#)
- [Kubernetes探秘—配置文件目录结构](#)
- [Kubernetes探秘-etcd节点和实例扩容](#)
- [Kubernetes探秘—etcd状态数据及其备份](#)
- etcd动态扩容，

<https://blog.csdn.net/ShouTouDeXingFu/article/details/81172308>

- [快速建立Kubernetes集群，从零开始](#)

- kube-keepalived-vip，

<https://github.com/kubernetes/contrib/tree/master/keepalived-vip>

- 使用 keepalived 部署高可用 Kubernetes Master，

<https://lonf.me/2017/02/15/high-availability-Kubernetes-Master/>