kubernetes 面试题汇总

1.kubernetes 是什么？

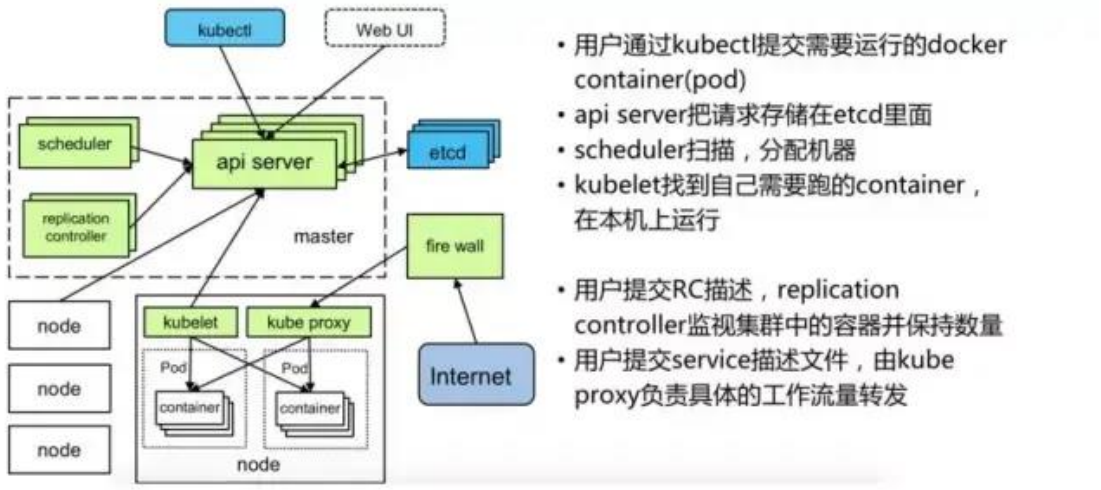
Kubernetes（k8s）是自动化容器操作的开源平台，这些操作包括部署，调度和节点集群间扩展。如果你曾经用过 Docker 容器技术部署容器，那么可以将 Docker 看成 Kubernetes 内部使用的低级别组件。Kubernetes 不仅仅支持 Docker，还支持 Rocket，这是另一种容器技术。
使用 Kubernetes 可以：

- 自动化容器的部署和复制
- 随时扩展或收缩容器规模
- 将容器组织成组，并且提供容器间的负载均衡
- 很容易地升级应用程序容器的新版本
- 提供容器弹性，如果容器失效就替换它，等等...

2.Kubernetes 有哪些特点？

- 可移植: 支持公有云，私有云，混合云，多重云（multi-cloud）
- 可扩展: 模块化，插件化，可挂载，可组合
- 自动化: 自动部署，自动重启，自动复制，自动伸缩/扩展

3.Kubernetes 架构和组件



4.Kubernetes 所能识别的最小单元什么？

Pod 就是 Kubernetes 所能识别的最小单元。它包含了一个或多个的容器并看做是一个整体的单元。基本上，可以说 Pod 就是一个单一的微服务。

5.K8S 与 Swarm 的共同点是什么？

Docker Swarm 和 Kubernetes 都是用来编排容器的，但是是以不同的方式。

6.关于集群

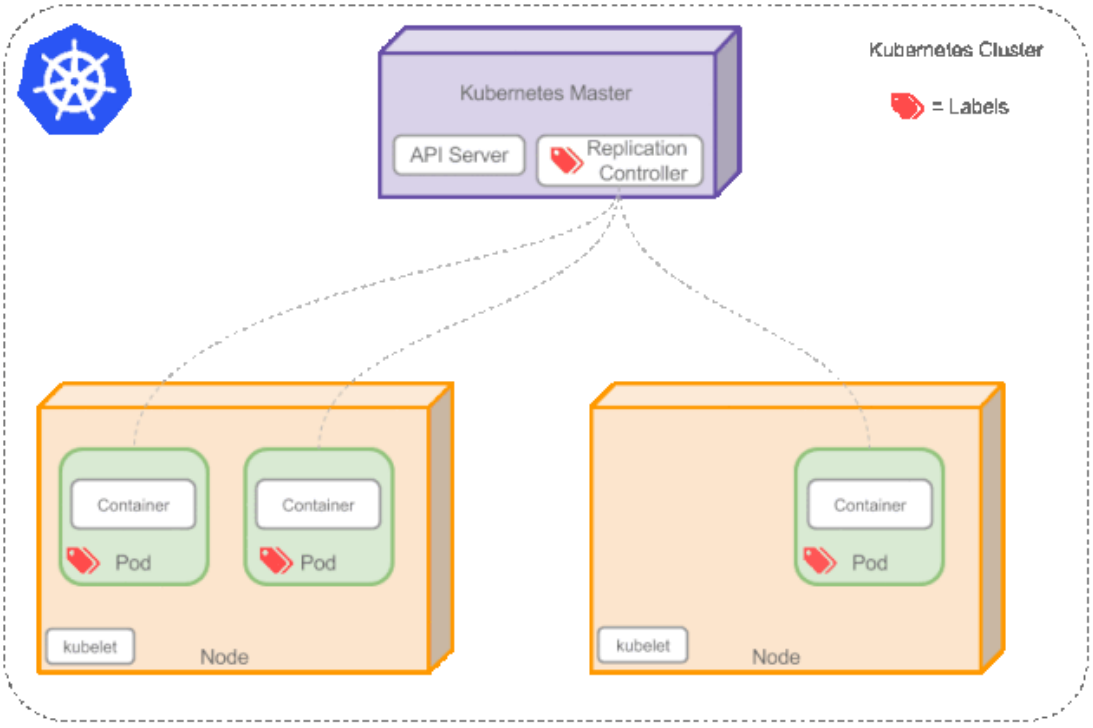
集群是一组节点，这些节点可以是物理服务器或者虚拟机，之上安装了 Kubernetes 平台。下图展示这样的集群。注意该图为了强调核心概念有所简化。这里可以看到一个典型的 Kubernetes 架构图。

7. 在安装 Kubernetes 时会因为无法拉取 gcr.io 镜像

在已知镜像名称和标签的情况，可以通过阿里云镜像仓库+GitHub 用 Dockerfile 重新打包 gcr.io 镜像，然后安装时从阿里云镜像仓库直接下载再重命名为 gcr.io 镜像。在未知镜像名称和标签的情况，需要先找一台可以科学上网的机器来装一遍，再通过 docker images 查看准确的镜像名称和标签。

8. 是否手动创建 Pod，如果想要创建同一个容器的多份拷贝，需要一个个分别创建出来么，能否将 Pods 划到逻辑组里？

Replication Controller 确保任意时间都有指定数量的 Pod“副本”在运行。如果为某个 Pod 创建了 Replication Controller 并且指定 3 个副本，它会创建 3 个 Pod，并且持续监控它们。如果某个 Pod 不响应，那么 Replication Controller 会替换它，保持总数为 3. 如下面的动画所示：



9.centos 下如何配置主机互信

- 在每台服务器需要建立主机互信的用户名执行以下命令生成公钥/密钥，默认回车即可。

```
ssh-keygen -t rsa
```

可以看到生成个公钥的文件

- 互传公钥，第一次需要输入密码，之后就OK了。

```
ssh-copy-id -i /root/.ssh/id_rsa.pub root@192.168.199.132 (-p 2222)
```

-p 端口 默认端口不加-p，如果更改过端口，就得加上-p，可以看到是在ssh/下生成了个authorized_keys的文件，记录了能登陆这台服务器的其他服务器的公钥。

- 测试看是否能登陆

```
ssh 192.168.199.132 (-p 2222)
```

10.如何监控部署在 Docker 容器上的应用程序？

- Kubernetes 可以通过设定 livenessProbe 属性来为一个 Pod 做健康检测。

11.怎样从外面访问一个跑着许多 Docker 实例的应用程序？

通过使用 Kubernetes 的 Service 资源，可以有多种方案实现对一个跑在 Kubernetes 里的带有多个实例的 Docker 应用的访问。可以使用一个公网 IP 来创建一个 Service，一个负载均衡 Service，或者说，如果是 HTTP 的情况下，用一个 Kubernetes 的 Ingress 资源。

12.Docker + Kubernetes 只能在 Linux 环境下运行吗？

不，Docker 加入对 Windows 的支持已经有一段时间了，而就在 1.5 版本的时候，Kubernetes 加入了对 Windows Server 容器的支持，控制器仍然还跑在 Linux 上，然后 Kubelet 和 Kubeproxy 则可以在 Windows 上运行。

13.Kubernetes 和 Openstack 发展方向是怎样的？它们之间存在很多分歧吗？

Kubernetes 和 Openstack 是两个完全不同的东西；真的没有必要去比较它们，因为它们根本从来都碰不到一起。你可以在 Openstack 上跑 Kubernetes，你也可以使用 Kubernetes 来编排 Openstack，但是它们始终还是两个截然不同的东西。

14.Mirantis 提供对 Kubernetes 的支持吗？

到目前为止，Mirantis 的产品只限于 Openstack，这也即是我们所支持的全部；当我们加入对 Kubernetes 的支持时，事情可能会有一定程度的转变，但是就目前而言，情况就是这样。

15.怎么把一个公网 IP 分配给一个跑在 Openstack 虚拟机里的 Docker 容器？

- 只要像分配任何其他基于 Openstack 的公网 IP 一样通过浮动 IP 去做就行。

16.应用和运行时平台是怎样解耦的？

容器是设计成自包含的。因此可以创建一个包含了系统的所有内容，让它拥有完备的移植性。我们也应该明白一点，应用程序不可能完全和运行时平台解耦。举个例子，你如果有一个应用是用 Mono（Linux 版本的.NET）写的，你可以用 Linux 上的 Kubernetes 来运行它，但是直接用 Windows Server 容器跑的话就只能运行在 Windows 上的 Kubernetes 了。

17.Docker/Kubernetes 可以用在 Windows 服务或者实际的应用，数据库，还有存储吗，或者说你可以创建 windows 的虚拟机然后在 Kubernetes 下面跑吗？

听上去所说的“实际应用”真的有点像是在说“宠物”类应用。如果是的话，那么最好还是用虚拟机来跑吧。

18.是不是可以这样说，Kubernetes 的编排就像一个流程图？一系列一个接一个的动作？

理想情况下，这是对的，但是实际上它并不是这样 —— 反正不是直接如此。当你在 YAML 文件里包含了多个定义时，没有办法保证它们会以怎样特定的顺序去执行。要解决这个问题实现“流程图”效果的话，你可以看下 Kubernetes 新的 APPController。

19.虽然容器是分层的，在宿主操作系统这块每个分层也是重复部署的。Openstack 会为此提供一个轻量级的容器宿主虚拟机吗？

与其操心有没有一个轻量级的容器宿主虚拟机镜像，还不如考虑下用一个最小集操作系统作为容器的基础层，比如 Alpine Linux。

20.企业部署 k8s，多少节点合适

多少个节点看业务，还有物理机配置，假如一个物理机配置特别高，那你部署十个节点，比你用 100 台低配置的机器都要跑的业务更多。

21.上万规模的容器的 kubernetes 集群，使用 kubernetes 时需要注意哪些问题？

上万规模需要用 ipvs 做转发，网络用 calico 性能更好。当 kubernetes 规模达到上万时，会出现如下问题
1）etcd 中出现了大量的读写延迟，并且产生了拒绝服务的情形，同时因其空间的限制也无法承载 Kubernetes 存储大量的对象；
2）API Server 查询 pods/nodes 延迟非常的高，并发查询请求可能地址后端 etcd oom；
3）Controller 不能及时从 API Server 感知到在最新的变化，处理的延时较高；

4）当发生异常重启时，服务的恢复时间需要几分钟；
5）Scheduler 延迟高、吞吐低，无法适应阿里业务日常运维的需求，更无法支持大促态的极端场景。
如何解决?

1）通过将索引和数据分离、数据 shard 等方式提高 etcd 存储容量，并最终通过改进 etcd 底层 bbolt db 存储引擎的块分配算法，大幅提高了 etcd 在存储大数据量场景下的性能，通过单 etcd 集群支持大规模 Kubernetes 集群，大幅简化了整个系统架构的复杂性；

2）通过落地 Kubernetes 轻量级心跳、改进 HA 集群下多个 API Server 节点的负载均衡、ListWatch 机制中增加 bookmark、通过索引与 Cache 的方式改进了 Kubernetes 大规模集群中最头疼的 List 性能瓶颈，使得稳定的运行万节点集群成为可能；

3）通过热备的方式大幅缩短了 controller/scheduler 在主备切换时的服务中断时间，提高了整个集群的可用性；

22. kubernetes 的运维中有哪些注意的要点？

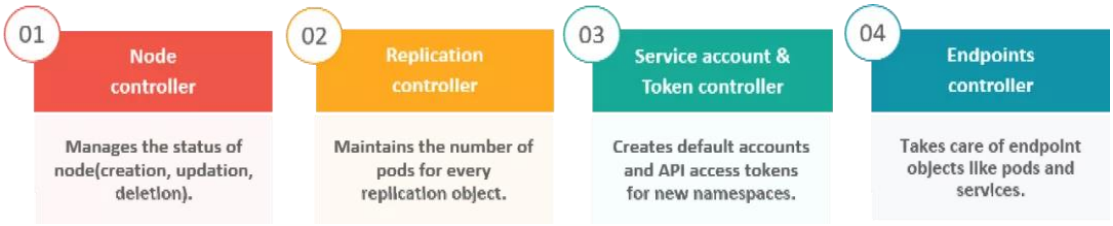
注意做好 hpa，还有 livenessProbe 和 readnessProbe 这种探测，数据持久化，监控等

23. kube-apiserver 和 kube-scheduler 的作用是什么？

kube－apiserver 遵循横向扩展架构，并且是主节点控制面板的前端。这将公开 Kubernetes 主节点组件的所有 API，并负责在 Kubernetes 节点和 Kubernetes 主组件之间建立通信。kube-scheduler 负责在工作节点上分配和管理工作负载。因此，它根据资源需求选择最合适的节点来运行 Pod，并跟踪资源利用率。它可以确保未在已满的节点上调度工作负载。

24. 您能简要介绍一下 Kubernetes 控制器管理器吗？

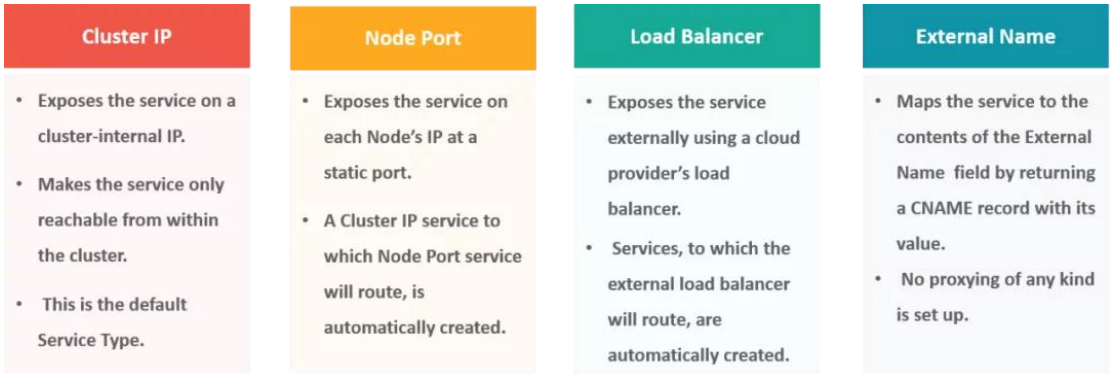
多个控制器进程在主节点上运行，但被编译在一起以作为单个进程（即 Kubernetes Controller Manager）运行。因此，Controller Manager 是一个守护程序，它嵌入控制器并执行名称空间创建和垃圾回收。它负责并与 API 服务器通信以管理端点。因此，在主节点上运行的不同类型的控制器管理器为：



25. 什么是 ETCD?

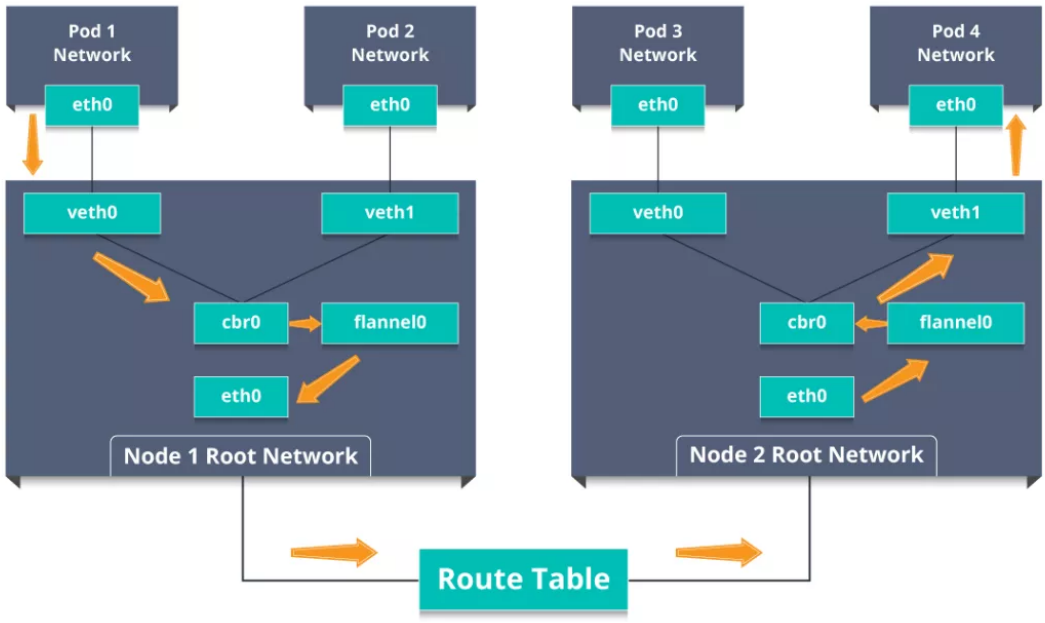
Etcd 用 Go 编程语言编写，并且是用于在分布式工作之间进行协调的分布式键值存储。因此，Etcd 存储 Kubernetes 集群的配置数据，该数据表示集群在任何给定时间点的状态。

26. Kubernetes 中有哪些不同类型的服务？



27. 什么是 Ingress 网络？它如何工作？

入口网络是充当 Kubernetes 集群入口点的规则的集合。这允许入站连接，可以将其配置为通过可访问的 URL，负载平衡流量或通过提供基于名称的虚拟主机在外部提供服务。因此，Ingress 是一个 API 对象，通常通过 HTTP 管理群集中对服务的外部访问，这是公开服务的最强大方法。现在，让我通过一个示例向您解释 Ingress 网络的工作原理。有 2 个节点具有带有 Linux 网桥的 pod 和根网络名称空间。除此之外，还在根网络中添加了一个名为 flannel0（网络插件）的新虚拟以太网设备。现在，假设我们希望数据包从 pod1 到 pod4。请参考下图。



- 因此，数据包在 eth0 离开 pod1 的网络，在 veth0 进入根网络。
- 然后将其传递给 cbr0，后者发出 ARP 请求以查找目标，并且发现该节点上没有人具有目标 IP 地址。
- 因此，网桥将数据包发送到 flannel0，因为节点的路由表已配置了 flannel0。现在，法兰绒守护程序与 Kubernetes 的 API 服务器进行对话，以了解所有 Pod IP 及其各自的节点，以创建 Pod IP 到节点 IP 的映射。
- 网络插件将该数据包包装在带有额外报头的 UDP 数据包中，该报头将源 IP 和目标 IP 更改为它们各自的节点，然后通过 eth0 发送此数据包。

- 现在，由于路由表已经知道如何在节点之间路由流量，因此它将数据包发送到目标节点 2。
- 数据包到达 node2 的 eth0，然后返回 flannel0 进行解封装，然后将其发送回根网络名称空间。
- 再次将数据包转发到 Linux 网桥，以发出 ARP 请求，以找出属于 veth1 的 IP。
- 数据包最终穿过根网络并到达目标 Pod4。

28.什么是联合集群？

借助联合集群，可以将多个 Kubernetes 集群作为一个集群进行管理。因此，您可以在一个数据中心/云中创建多个 Kubernetes 集群，并使用联合在一个地方控制/管理所有集群。

29.您能否简要介绍一下 Kubernetes 中主节点的工作？

Kubernetes 主节点控制节点，并且在节点内部存在容器。现在，这些单独的容器包含在 Pod 内，每个 Pod 内，根据配置和要求，您可以拥有各种数量的容器。因此，如果必须部署 Pod，则可以使用用户界面或命令行界面来部署它们。然后，将这些 Pod 调度在节点上，并根据资源需求将 Pod 分配给这些节点。kube-apiserver 确保在 Kubernetes 节点和主组件之间建立了通信。

30.Kubelet 调用的处理检查容器的 IP 地址是否打开的程序是？

- TCPSocketAction

31.什么是 headless service？

headless service 用于与服务发现机制进行交互，而无需与 ClusterIP 绑定，因此使您可以直接访问 Pod，而不必通过代理访问它们。当既不需要负载平衡又不需要单个服务 IP 时，此功能很有用。

32.Kubernetes 和 Docker 有什么关系？

Docker 是用于处理软件开发的开源平台。它的主要优点是将将软件/应用程序运行所需的设置和依赖项打包到一个容器中，从而实现了可移植性和其他一些优点。Kubernetes 允许手动链接和编排多个容器，这些容器在使用 Docker 创建的多个主机上运行。

33.在 Kubernetes 中解释 Prometheus？

Prometheus 是一个开源工具包，用于基于度量的监视和警报应用程序。它提供了数据模型和查询语言,并且可以提供指标的详细信息和操作。它支持多种语言的工具应用程序语言。除了 Alertmanager 和 Grafana 之外，Prometheus 操作员还可以轻松监视部署和 k8s 服务。

34.Kubernetes 有什么优势？

Kubernetes 的优点如下：

Kubernetes 是开源且免费的

它具有高度的可扩展性，可以在任何操作系统中运行

它提供了更多的概念和更强大的 Docker 群

它提供调度程序，自动扩展，滚动升级和运行状况检查
它具有平坦的网络空间和自定义功能
建立有效的 CI / CD 管道很容易
它可以提高生产力

35.Kubernetes 的缺点是什么？

Kubernetes 的缺点如下：
安装过程和配置非常困难
管理服务并不容易
运行和编译需要很多时间
它比其他替代品更昂贵
对于简单的应用程序来说，这可能是一个过大的杀伤力

36.什么是容器？

容器是一种在运行时需要时为应用程序收集已编译代码的技术。每当容器运行时，每个容器都允许您运行可重复的标准依赖项和相同的行为。它将应用程序与基础主机基础结构分开，以使在云或 OS 平台中的部署更加容易。