

# Kubernetes 面试问题（附答案）

## 基本的 Kubernetes 面试问题

这部分问题将包含您需要了解的与 Kubernetes 工作相关的所有基本问题。

### Q1. Kubernetes 与 Docker Swarm 的区别如何？

Features	Kubernetes	Docker Swarm
<b>Installation &amp; Cluster Configuration</b>	Installation is complicated; but once setup, the cluster is very strong	Installation is very simple; but cluster is not very strong
<b>GUI</b>	GUI is the Kubernetes Dashboard	There is no GUI
<b>Scalability</b>	Highly scalable & scales fast	Highly scalable & scales 5x faster than Kubernetes
<b>Auto-Scaling</b>	Kubernetes can do auto-scaling	Docker Swarm cannot do auto-scaling
<b>Load Balancing</b>	Manual intervention needed for load balancing traffic between different containers in different Pods	Docker Swarm does auto load balancing of traffic between containers in the cluster
<b>Rolling Updates &amp; Rollbacks</b>	Can deploy Rolling updates & does automatic Rollbacks	Can deploy Rolling updates, but not automatic Rollbacks
<b>Data Volumes</b>	Can share storage volumes only with other containers in same Pod	Can share storage volumes with any other container
<b>Logging &amp; Monitoring</b>	In-built tools for logging & monitoring	3rd party tools like ELK should be used for logging & monitoring

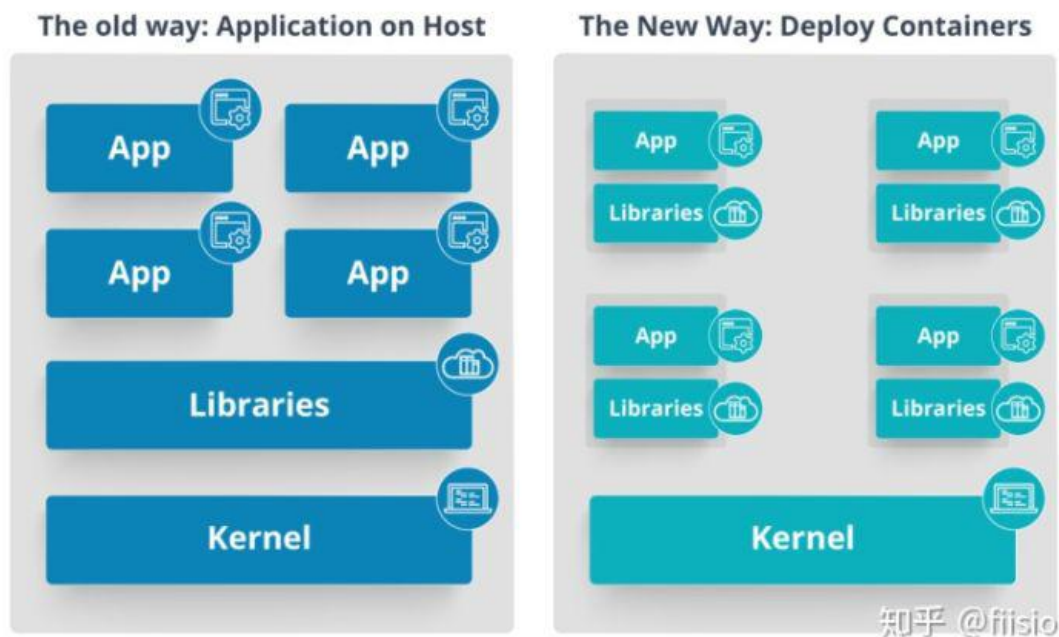
### Q2. 什么是 Kubernetes？

Kubernetes 是一个开源容器管理工具，负责容器部署，容器扩缩容以及负载平衡。作为 Google 的创意之作，它提供了出色的社区，并与所有云提供商合作。因此，我们可以说 Kubernetes 不是一个容器化平台，而是一个多容器管理解决方案。

### Q3. Kubernetes 与 Docker 有什么关系？

众所周知，Docker 提供容器的生命周期管理，Docker 镜像构建运行时容器。但是，由于这些单独的容器必须通信，因此使用 Kubernetes。因此，我们说 Docker 构建容器，这些容器通过 Kubernetes 相互通信。因此，可以使用 Kubernetes 手动关联和编排在多个主机上运行的容器。

### Q4. 在主机和容器上部署应用程序有什么区别？



请参考上图。左侧架构表示在主机上部署应用程序。因此，这种架构将具有操作系统，然后操作系统将具有内核，该内核将在应用程序所需的操作系统上安装各种库。因此，在这种框架中，您可以拥有  $n$  个应用程序，并且所有应用程序将共享该操作系统中存在的库，而在容器中部署应用程序时，体系结构则略有不同。

这种架构将有一个内核，这是唯一一个在所有应用程序之间唯一共同的东西。因此，如果有一个需要 Java 的特定应用程序，那么我们将获得访问 Java 的特定应用程序，如果有另一个需要 Python 的应用程序，则只有该特定应用程序才能访问 Python。

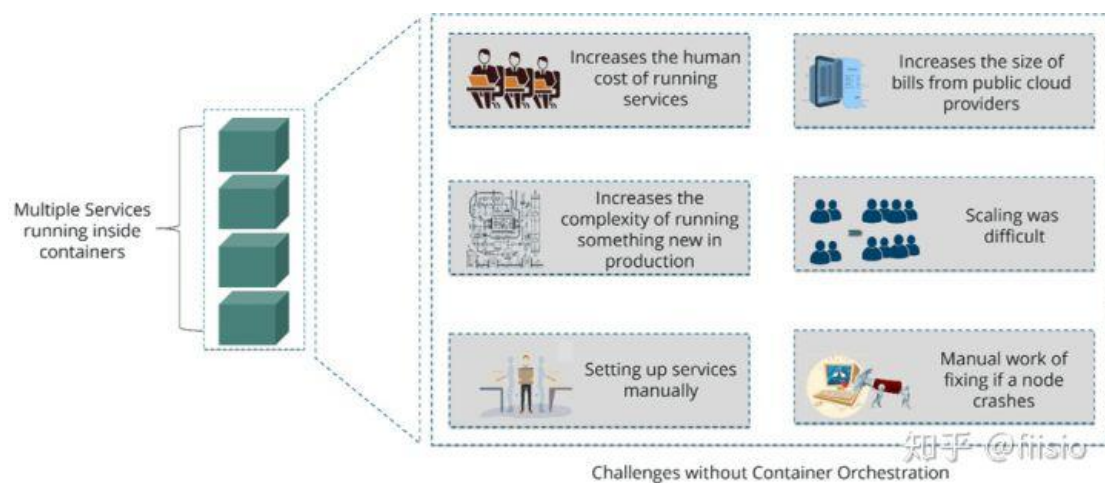
您可以在图表右侧看到的各个块基本上是容器化的，并且这些块与其他应用程序隔离。因此，应用程序具有与系统其余部分隔离的必要库和二进制文件，并且不能被任何其他应用程序侵占。

## Q5. 什么是 Container Orchestration?

考虑一个应用程序有 5-6 个微服务的场景。现在，这些微服务被放在单独的容器中，但如果没有容器编排就无法进行通信。因此，由于编排意味着所有乐器在音乐中和谐共处，所以类似的容器编排意味着各个容器中的所有服务协同工作以满足单个服务器的需求。

## Q6. Container Orchestration 需要什么?

考虑到你有 5-6 个微服务用于执行各种任务的单个应用程序，所有这些微服务都放在容器中。现在，为了确保这些容器彼此通信，我们需要容器编排。



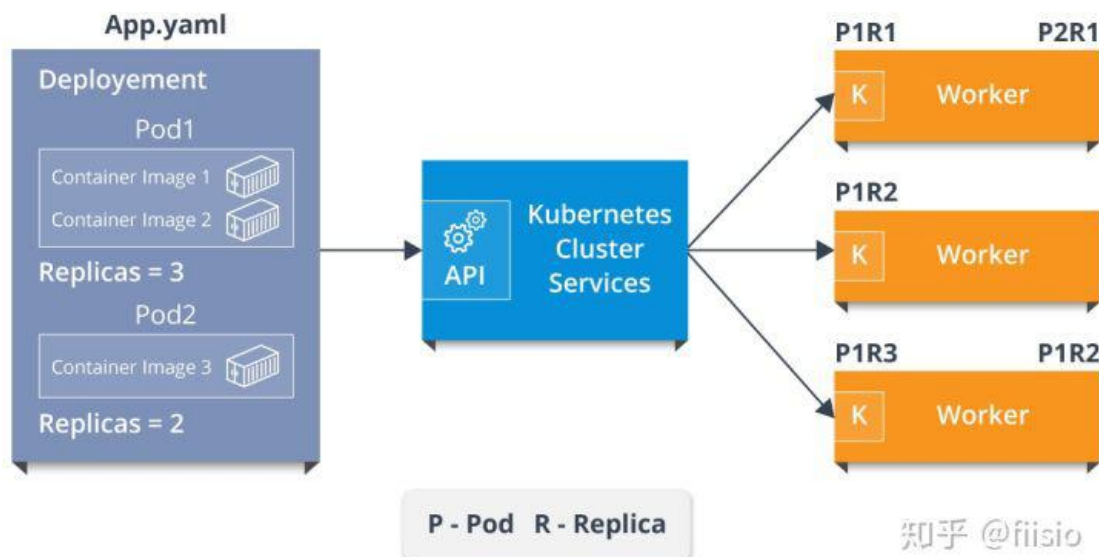
正如您在上图中所看到的，在没有使用容器编排的情况下，还存在许多挑战。因此，为了克服这些挑战，容器编排就到位了。

## Q8. Kubernetes 如何简化容器化部署?

由于典型应用程序将具有跨多个主机运行的容器集群，因此所有这些容器都需要相互通信。因此，要做到这一点，你需要一些能够负载平衡，扩展和监控容器的东西。由于 Kubernetes 与云无关并且可以在任何公共/私有提供商上运行，因此必须是您简化容器化部署的选择。

## Q9. 您对 Kubernetes 的集群了解多少?

Kubernetes 背后的基础是我们可以实施所需的状态管理，我的意思是我们可以提供特定配置的集群服务，并且集群服务将在基础架构中运行并运行该配置。



因此，正如您在上图中所看到的，部署文件将具有提供给集群服务所需的所有配置。现在，部署文件将被提供给 API，然后由集群服务决定如何在环境中安排这些 pod，并确保正确运行的 pod 数量。

因此，位于服务前面的 API，工作节点和节点运行的 Kubelet 进程，共同构成了 Kubernetes 集群。

## Q10. 什么是 Google 容器引擎？

Google Container Engine (GKE) 是 Docker 容器和集群的开源管理平台。这个基于 Kubernetes 的引擎仅支持在 Google 的公共云服务中运行的群集。

## Q11. 什么是 Heapster？

Heapster 是由每个节点上运行的 Kubelet 提供的集群范围的数据聚合器。此容器管理工具在 Kubernetes 集群上本机支持，并作为 pod 运行，就像集群中的任何其他 pod 一样。因此，它基本上发现集群中的所有节点，并通过机上 Kubernetes 代理查询集群中 Kubernetes 节点的使用信息。

## Q12. 什么是 Minikube？

Minikube 是一种工具，可以在本地轻松运行 Kubernetes。这将在虚拟机中运行单节点 Kubernetes 群集。

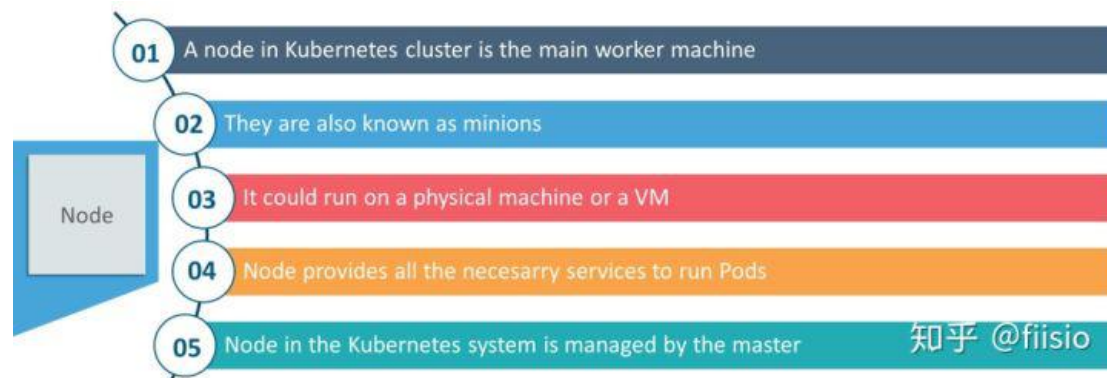
## Q13. 什么是 Kubectl?

Kubectl 是一个平台，您可以使用该平台将命令传递给集群。因此，它基本上为 CLI 提供了针对 Kubernetes 集群运行命令的方法，以及创建和管理 Kubernetes 组件的各种方法。

## Q14. 什么是 Kubelet?

这是一个代理服务，它在每个节点上运行，并使从服务器与主服务器通信。因此，Kubelet 处理 PodSpec 中提供给它的容器的描述，并确保 PodSpec 中描述的容器运行正常。

## Q15. 你对 Kubernetes 的一个节点有什么了解?



## 基于架构的 Kubernetes 访谈问题

这部分问题将涉及与 Kubernetes 架构相关的问题。

## Q1. Kubernetes Architecture 的不同组件有哪些?

Kubernetes Architecture 主要有两个组件 - 主节点和工作节点。如下图所示，master 和 worker 节点中包含许多内置组件。主节点具有 kube-controller-manager, kube-apiserver, kube-scheduler 等。而工作节点具有在每个节点上运行的 kubelet 和 kube-proxy。

## Q2. 你对 Kube-proxy 有什么了解?

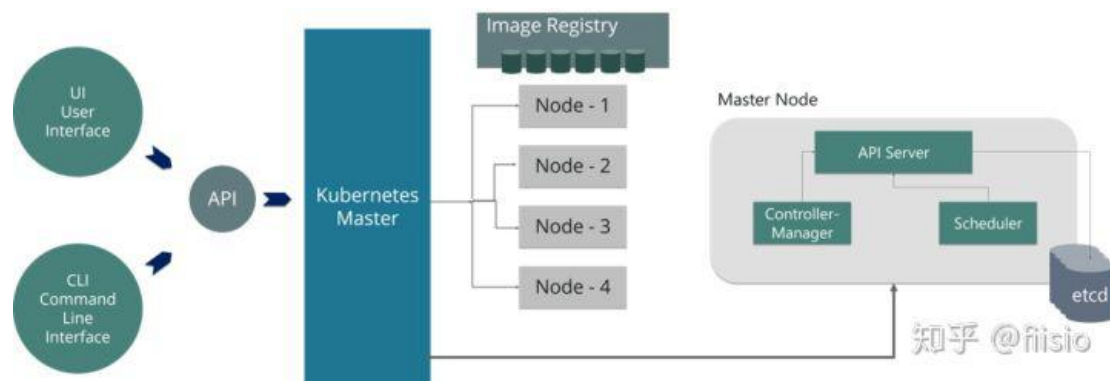
Kube-proxy 可以在每个节点上运行，并且可以跨后端网络服务进行简单的 TCP / UDP 数据



包转发。基本上，它是一个网络代理，它反映了每个节点上 Kubernetes API 中配置的服务。因此，Docker 可链接的兼容环境变量提供由代理打开的群集 IP 和端口。

### Q3. 您能否介绍一下 Kubernetes 中主节点的工作情况？

Kubernetes master 控制容器存在的节点和节点内部。现在，这些单独的容器包含在容器内部和每个容器内部，您可以根据配置和要求拥有不同数量的容器。因此，如果必须部署 pod，则可以使用用户界面或命令行界面部署它们。然后，在节点上调度这些 pod，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通信。



### Q4. kube-apiserver 和 kube-scheduler 的作用是什么？

kube-apiserver 遵循横向扩展架构，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 负责工作节点上工作负载的分配和管理。因此，它根据资源需求选择最合适的节点来运行未调度的 pod，并跟踪资源利用率。它确保不在已满的节点上调度工作负载。

### Q5. 你能简要介绍一下 Kubernetes 控制管理器吗？

多个控制器进程在主节点上运行，但是一起编译为单个进程运行，即 Kubernetes 控制器管理器。因此，Controller Manager 是一个嵌入控制器并执行命名空间创建和垃圾收集的守护程序。它拥有责任并与 API 服务器通信以管理端点。

因此，主节点上运行的不同类型的控制器管理器是：

## Q6. 什么是 ETCD?

Etcd 是用 Go 编程语言编写的，是一个分布式键值存储，用于协调分布式工作。因此，Etcd 存储 Kubernetes 集群的配置数据，表示在任何给定时间点的集群状态。

## Q8. 你对 Kubernetes 的负载均衡器有什么了解?

负载均衡器是暴露服务的最常见和标准方式之一。根据工作环境使用两种类型的负载均衡器，即内部负载均衡器或外部负载均衡器。内部负载均衡器自动平衡负载并使用所需配置分配容器，而外部负载均衡器将流量从外部负载引导至后端容器。

## Q9. 什么是 Ingress 网络，它是如何工作的?

Ingress 网络是一组规则，充当 Kubernetes 集群的入口点。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从外部提供服务。因此，Ingress 是一个 API 对象，通常通过 HTTP 管理集群中服务的外部访问，是暴露服务的最有效方式。

现在，让我以一个例子向您解释 Ingress 网络的工作。

有 2 个节点具有带有 Linux 桥接器的 pod 和根网络命名空间。除此之外，还有一个名为 flannel0（网络插件）的新虚拟以太网设备被添加到根网络中。

现在，假设我们希望数据包从 pod1 流向 pod 4.请参阅下图。

因此，数据包将 pod1 的网络保留在 eth0，并进入 veth0 的根网络。

然后它被传递给 cbr0，这使得 ARP 请求找到目的地，并且发现该节点上没有人具有目的地 IP 地址。

因此，桥接器将数据包发送到 flannel0，因为节点的路由表配置了 flannel0。

现在，flannel 守护程序与 Kubernetes 的 API 服务器通信，以了解所有 pod IP 及其各自的节点，以创建 pods IP 到节点 IP 的映射。

网络插件将此数据包封装在 UDP 数据包中，其中额外的标头将源和目标 IP 更改为各自的节点，并通过 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 并返回到 flannel0 以解封装并在根网络命名空间中将其发回。

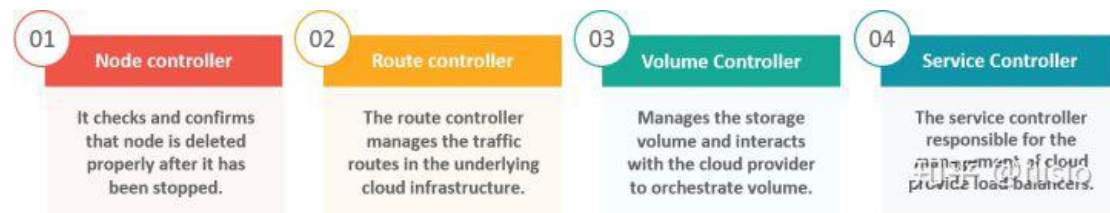
同样，数据包被转发到 Linux 网桥以发出 ARP 请求以找出属于 veth1 的 IP。

数据包最终穿过根网络并到达目标 Pod4。

## Q10. 您对云控制器管理器有何了解？

Cloud Controller Manager 负责持久存储，网络路由，从核心 Kubernetes 特定代码中抽象出特定于云的代码，以及管理与底层云服务的通信。它可能会分成几个不同的容器，具体取决于您运行的是哪个云平台，然后它可以使云供应商和 Kubernetes 代码在没有任何相互依赖的情况下开发。因此，云供应商开发他们的代码并在运行 Kubernetes 时与 Kubernetes 云控制器管理器连接。

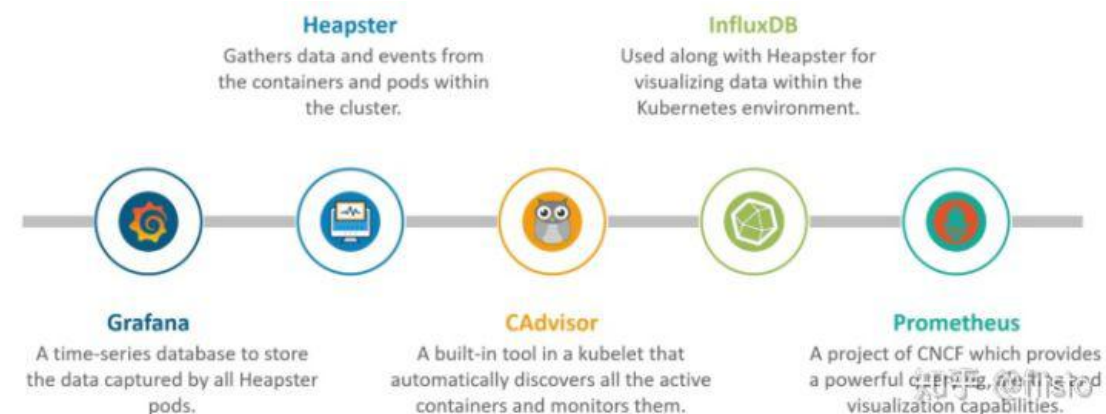
各种类型的云控制器管理器如下：



## Q11. 什么是 Container 资源监控？

对于用户而言，了解应用程序的性能和所有不同抽象层的资源利用率非常重要，Kubernetes 通过在容器，pod，服务和整个集群等不同级别创建抽象来考虑集群的管理。现在，可以监视每个级别，这只是容器资源监视。

各种容器资源监控工具如下：



## Q12. Replica Set 和 Replication Controller 之间有什么区别？

Replica Set 和 Replication Controller 几乎完全相同。它们都确保在任何给定时间运行指定数量的 pod 副本。不同之处在于复制 pod 使用的选择器。Replica Set 使用基于集合的选择



器，而 Replication Controller 使用基于权限的选择器。

**Equity-Based 选择器：**这种类型的选择器允许按标签键和值进行过滤。因此，在外行术语中，基于 Equity 的选择器将仅查找与标签具有完全相同短语的 pod。示例：假设您的标签键表示 app = nginx, 那么, 使用此选择器, 您只能查找标签应用程序等于 nginx 的那些 pod。

**Selector-Based 选择器：**此类型的选择器允许根据一组值过滤键。因此，换句话说，基于 Selector 的选择器将查找已在集合中提及其标签的 pod。示例：假设您的标签键在 (nginx, NPS, Apache) 中显示应用程序。然后，使用此选择器，如果您的应用程序等于任何 nginx, NPS 或 Apache，则选择器将其视为真实结果。

## Q13. 什么是 Headless Service?

Headless Service 类似于“普通”服务，但没有群集 IP。此服务使您可以直接访问 pod，而无需通过代理访问它。

## Q14. 什么是集群联邦?

在联邦集群的帮助下，可以将多个 Kubernetes 集群作为单个集群进行管理。因此，您可以在数据中心/云中创建多个 Kubernetes 集群，并使用联邦来在一个位置控制/管理它们。

## 基于场景的面试问题

这部分问题将包含您在面试中可能遇到的各种基于场景的问题。

### 场景 1:

假设一家基于单一架构的公司处理众多产品。现在，随着公司在当今的扩展行业的扩展，他们的单一架构开始引发问题。

您如何看待公司从单一服务转向微服务并部署其服务容器？

解:

由于公司的目标是从单一应用程序转向微服务，它们最终可以逐个构建，并行构建，只需在后台切换配置。然后他们可以将这些内置微服务放在 Kubernetes 平台上。因此，他们可以从一次或两次迁移服务开始，并监控它们以确保一切运行稳定。一旦他们觉得一切顺利，他们就可以将其余的应用程序迁移到他们的 Kubernetes 集群中。

## 场景 2:

考虑一家拥有分布式系统的跨国公司，拥有大量数据中心，虚拟机和许多从事各种任务的员工。

您认为这样的公司如何以与 **Kubernetes** 一致的方式管理所有任务？

解：

正如我们所有人都知道 IT 部门推出了数千个容器，其任务在分布式系统中遍布全球众多节点。

在这种情况下，公司可以使用能够为基于云的应用程序提供敏捷性，横向扩展功能和 DevOps 实践的东西。

因此，该公司可以使用 **Kubernetes** 来定制他们的调度架构并支持多种容器格式。这使得容器任务之间的亲和性成为可能，从而提供更高的效率，并为各种容器网络解决方案和容器存储提供广泛支持。

## 场景 3:

考虑一种情况，即公司希望通过维持最低成本来提高其效率和技术运营速度。

您认为公司将如何实现这一目标？

解：

公司可以通过构建 **CI/CD** 管道来实现 **DevOps** 方法，但是这里可能出现的一个问题是配置可能需要一段时间才能启动并运行。因此，在实施 **CI/CD** 管道之后，公司的下一步应该是在云环境中工作。一旦他们开始处理云环境，他们就可以在集群上安排容器，并可以在 **Kubernetes** 的帮助下进行协调。这种方法将有助于公司缩短部署时间，并在各种环境中加快速度。

## 场景 4:

假设一家公司想要修改它的部署方法，并希望建立一个更具可扩展性和响应性的平台。

您如何看待这家公司能够实现这一目标以满足客户需求？

解:

为了给数百万客户提供他们期望的数字体验，公司需要一个可扩展且响应迅速的平台，以便他们能够快速地将数据发送到客户网站。现在，要做到这一点，公司应该从他们的私有数据中心（如果他们使用任何）转移到任何云环境，如 **AWS**。不仅如此，他们还应该实现微服务架构，以便他们可以开始使用 **Docker** 容器。一旦他们准备好基础框架，他们就可以开始使用最好的编排平台，即 **Kubernetes**。这将使团队能够自主地构建应用程序并快速交付它们。

## 场景 5:

考虑一家拥有非常分散的系统的跨国公司，期待解决整体代码库问题。

您认为公司如何解决他们的问题？

解

那么，为了解决这个问题，我们可以将他们的单片代码库转移到微服务设计，然后每个微服务都可以被视为一个容器。因此，所有这些容器都可以在 **Kubernetes** 的帮助下进行部署和协调。

## 场景 6:

我们所有人都知道，从单片到微服务的转变解决了开发方面的问题，但却增加了部署方面的问题。

公司如何解决部署方面的问题？

解

团队可以试验容器编排平台，例如 **Kubernetes**，并在数据中心运行。因此，通过这种方式，公司可以生成模板化应用程序，在五分钟内部署它，并在此时将实际实例集中在暂存环境中。这种 **Kubernetes** 项目将有数十个并行运行的微服务，以提高生产率，即使节点出现故障，也可以立即重新安排，而不会影响性能。

## 场景 7:

假设一家公司希望通过采用新技术来优化其工作负载的分配。

公司如何有效地实现这种资源分配？

解

这个问题的解决方案就是 **Kubernetes**。**Kubernetes** 确保资源得到有效优化，并且只使用特定应用程序所需的那些资源。因此，通过使用最佳容器编排工具，公司可以有效地实现资源分配。

## 场景 8:

考虑一家拼车公司希望通过同时扩展其平台来增加服务器数量。

您认为公司如何处理服务器及其安装？

解

公司可以采用集装箱化的概念。一旦他们将所有应用程序部署到容器中，他们就可以使用 **Kubernetes** 进行编排，并使用像 **Prometheus** 这样的容器监视工具来监视容器中的操作。因此，利用容器的这种使用，在数据中心中为它们提供更好的容量规划，因为它们现在将受到更少的限制，因为服务和它们运行的硬件之间存在抽象。

## 场景 9:

考虑一种情况，公司希望向具有各种环境的客户提供所有必需的分发。

您认为他们如何以动态的方式实现这一关键目标？

解

该公司可以使用 **Docker** 环境，组建一个横截面团队，使用 **Kubernetes** 构建 **Web** 应用程序。这种框架将帮助公司实现在最短的时间内将所需产品投入生产的目标。因此，在这样的机器运行的情况下，公司可以向所有具有各种环境的客户发放电子邮件。

## 场景 10:

假设公司希望在不同的云基础架构上运行各种工作负载，从裸机到公共云。

公司将如何在不同界面的存在下实现这一目标？

解

该公司可以将其基础设施分解为微服务，然后采用 **Kubernetes**。这将使公司在不同的云基础设施架构上运行各种工作负载。

## 多项选择面试问题

这部分问题将包括多项面试问题，这些问题在面试中经常被问到。

### Q1. 什么是 Kubernetes 集群中的 minions?

它们是主节点的组件。

它们是集群的工作节点。[答案]

他们正在监控 **kubernetes** 中广泛使用的引擎。

他们是 **docker** 容器服务。

### Q2. Kubernetes 集群数据存储在以下哪个位置?

KUBE-API 服务器

Kubelet

ETCD [答案]

以上都不是

### Q3. 哪个是 Kubernetes 控制器?

ReplicaSet

Deployment

Rolling Updates

ReplicaSet 和 Deployment [答案]

### Q4. 以下哪个是核心 Kubernetes 对象?

Pods

Services

Volumes

以上所有[答案]



## Q5. Kubernetes Network 代理在哪个节点上运行?

Master Node

Worker Node

所有节点[答案]

以上都不是

## Q6. 节点控制器的职责是什么?

将 CIDR 块分配给节点

维护节点列表

监视节点的运行状况

以上所有[答案]

## Q7. Replication Controller 的职责是什么?

使用单个命令更新或删除多个 pod

有助于达到理想状态

如果现有 Pod 崩溃，则创建新 Pod

以上所有[答案]

## Q8. 如何在没有选择器的情况下定义服务?

指定外部名称[答案]

指定具有 IP 地址和端口的端点

只需指定 IP 地址即可

指定标签和 api 版本

## Q9. 1.8 版本的 Kubernetes 引入了什么?

Taints and Tolerations [答案]

Cluster level Logging

Secrets

Federated Clusters

## Q10. Kubelet 调用的处理检查容器的 IP 地址是否打开的程序是?

HTTPGetAction

ExecAction

TCPSocketAction [答案]

以上都不是

文章整理至:

原文: <https://medium.com/edureka/kubernetes-interview-questions-c9790e5abddb>

译者: 王国梁

来源: <https://zhuanlan.zhihu.com/p/74560934>