

0面试题目

ATS

Exported on 02/24/2023

Table of Contents

No headings included in this document

题目类型	等级	面试题目录	参考答案	面试建议
操作系统	初级题目	<ol style="list-style-type: none"> 1. cpu的使用率百分比怎么查看？ 2. 内存使用查看 3. dns 解析 4. shell 的简单使用！ 5. ansible 常用模块 6. http 常见状态码；http和https的区别。 7. 常见的网络协议 8. 时间同步 	<ol style="list-style-type: none"> 1. top 命令 sar 命令 vmstat 等 2. 使用 free, top 命令等 3. dns解析域名到DNS服务器的过程 4. shell 自由发挥 5. 能说几个即可 shell ,ping,copy等 6. 200,404,500, SSL证书 7. http, https,tcp,udp,web services, 	能回答出来即可
操作系统	中级	<ol style="list-style-type: none"> 1. 怎么判断cpu比较繁忙，压力比较大 2. 怎么看空闲内存 3. 网络基本故障排查流程 4. ansible playbook 关键字 5. tcp的三次握手 6. tcp和udp的区别。 	<ol style="list-style-type: none"> 1. 从服务到性能，使用率分析 2. top free 等命令 3. 工具与思路 4. 能说几个即可 5. 能说出关键点即可 6. 安全和非安全等 	能回答出来，并带有自己的理解
操作系统	高级	<ol style="list-style-type: none"> 1. cpu的负载怎么查看，负载合理值是多少。 2. 负载和cpu的使用率百分比关系。 3. linux系统内存的分配 	<ol style="list-style-type: none"> 1. 命令, loadover 2. 3. 阐述内存分配机制 	有深入的理解，不止于表面答案
docker 相关	初级题目	<ol style="list-style-type: none"> 1. docker exec, cp 命令，常用命令 2. Docker 网络模式，暴露端口。 3. docker inspect 什么作用。 	<ol style="list-style-type: none"> 1. 能说出使用方式即可 2. Host模式container模式 bridge模式（默认） None模式, -p 3. 查看容器详细信息 	能回答出来即可
docker 相关	中级	<ol style="list-style-type: none"> 1. docker 存储方式 2. Dockerfile 优化。 3. docker start和run的区别。 	<ol style="list-style-type: none"> 1. 四种存储方式:默认、volumes数据卷、bind mounts挂载、tmpfs mount(仅在linux环境中提供) 2. 如何优化思路，层 3. run命令只在第一次运行镜像操作时使用，相当于执行了两步操作，将镜像放入容器中然后将容器启动；而start命令在重新启动已经存在的镜像时使用，使用该命令需要知道容器的id或者名字。 	

docker 相关	高级	<ol style="list-style-type: none"> 1. docker 日志驱动类型 2. ci/cd 使用gitlab runner 等工具进行 ci/cd, 包括 镜像构建, 容器化应用 yml部署。 3. CMD 和ENTRYPOINT。 4. 联合文件系统。 	<ol style="list-style-type: none"> 1. 分五大类 <ul style="list-style-type: none"> • none 是 disable 容器日志功能 • syslog 和 journald 是 linux上的两种日志管理服务 • awslogs, splunk 和 gcplogs logentries是第三方日志托管服务 • gelf 和 fluentd 是两种开源的日志管理方案 • json-file默认使用 2. 简述流程与构建过程与配置。 3. ENTRYPOINT和CMD都是用来指定容器启动时运行的命令。 CMD指令的目的是为容器提供默认的执行命令, 命令行参数可以覆盖CMD指令的设置, 这只是重写, 不能给CMD通过命令行传递参数。 ENTRYPOINT指令为exec模式时命令行上的参数会作为参数添加到ENTRYPOINT指定命令的参数列表中 4. 联合文件系统(UnionFS)是一种分层、轻量级并且高性能的文件系统,它支持对文件系统的修改作为一次提交来一层层的叠加,同时可以将不同目录挂载到同一个虚拟文件系统下。 	
-----------	----	--	---	--

Kubernetes	初级	<ol style="list-style-type: none"> 1. Kubernetes 组件有哪些。 2. deployment 是什么，有什么用。 3. service 是什么，有什么用。 4. 一个pod创建流程 5. 如何暴露内部服务到集群外。 6. configmap 是什么，有什么用。 7. pod运行失败了怎么处理。 8. 如何设定POD cpu 内存大小。 9. kubernetes发布策略（4种） 10. 容器的驱逐时间是。 11. 为什么k8s要用申明式？ 12. 如何使用k8s实现滚动更新，配置文件机制。 	<ol style="list-style-type: none"> 1. api-server,控制器, etcd,kubelet等 2. 提供负载PODIP功能 3. service IP, 用来负载 PodIP 4. apiserver,调度器, 控制器,node节点, 创建, 控制器。 5. 使用ingress ,或者LB 6. 配置文件, 为POD提供额外配置。 7. 检查为什么失败, 并处理, 删除或重建。 8. 使用kubectl 编辑容器, 修改request,limit 9. 重建 (recreate),滚动, 蓝绿部署, 金丝雀发布 10. 默认5分钟 11. 只需描述期望状态就行,剩下的交给k8s。 12. deployment strategy RollingUpdate 	
------------	----	--	--	--

Kubernetes	中级	<ol style="list-style-type: none"> 1. Replica Set 和 Replication Controller之间有什么区别。 2. deployment部署了, pod没有怎么排查。 3. 多个pod能挂载pvc么。 4. 什么是Ingress网络, 它是如何工作的。 5. 什么是Headless Service, headless-service使用场景。 6. pod如何和node网络通信的。 7. etcd高可用及备份恢复方案。 8. 网络选型需要注意什么。 9. pod中pending状态, 是什么原因产生的。 10. pod出现问题, 排查思路。 11. 容器的驱逐时间是。 12. 节点notready是什么导致的? 13. notready会发生什么。 14. service 都有哪些类型, 如果应用需要粘性会话如何配置?。 15. k8s的service和ep是如何关联和相互影响的。 16. 你们监控用的什么, 怎么利用普罗米修斯监控pod信息, k8s状态, 如果来设计相关的监控如何落地。 	<ol style="list-style-type: none"> 1. 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 2. 排查思路与相关命令使用。 3. 能 4. Ingress网络是一组规则, 充当Kubernetes集群的入口点。暴露端口。 5. 特殊的service, 其 spec:clusterIP表示为 None, 自主选择权, 单独Pod 访问。 6. 设置POD网络, 使用转发, 共享网络。 7. 架构与流程思路, 方案思路。 8. pod跨node的通信, 网络性能, 网络插件成熟度, 网络策略, (最常用的应该是flannel和calico) 9. 节点资源不足, 不满足 nodeSelector 与 affinity, Node 存在 Pod 没有容忍的污点, kube-scheduler 未正常运行, 镜像不存在。 10. kubectl describe pod 查看具体信息, 到相应 node上查看具体日志。 11. node notready时, 默认驱逐时间是5min, 可修改, 是由kube-controller-manager的 pod-eviction-timeout控制的。node资源不足时也会发生驱逐, 是按照qos等级来驱逐的。 12. kubelet异常, 不能和 apiserver通信, 网络异常, kubelet version, kernel 13. node notready 5min, pod 会驱逐到其他节点。
------------	----	---	---

14. ClusterIP, NodePort, LoadBalancer以及 ExternalName, sessionAffinity设置
15. service中有selector时才会创建endpoints. kube-proxy 会监视 Kubernetes 控制节点对 Service 对象和 Endpoints 对象的添加和移除。只要服务中的pod集合发生改变, endpoints 就会被更新。
16. 使用prometheus-operator, 数据持久化, 如果集群过大可以考虑使用thanos高可用。

pod 的metrics信息已经集成到kubelet, 直接用 cAdvisor提供的metrics接口获取到所有容器相关的性能指标数据。

自定义相关模板 通过 grafana展示。(grafana上有许多优秀的模板)

Kubernetes	高级	<ol style="list-style-type: none"> 1. traefik的实现原理？ 2. 使用什么pod控制器, crd都需要定义什么, 存活性监测和就绪性监测的实现方式。 3. deployment的历史记录默认保存多少?, 如何设定? 这个历史记录和对应的rs有什么关系。 4. deployment部署的时候日志报错, 应用无权限写到挂载的目录, 如何处理?。 5. etcd用的什么算法, 简单解释一下。 6. 你们监控用的什么, 怎么利用普罗米修斯监控pod信息, k8s状态, 如果你来设计相关的监控如何落地。 7. statefulset是怎么实现滚动更新的。 8. 如何实现schedule水平扩展。 9. api-server到etcd怎么保证事件不丢失。 10. sidecar要保证顺序启动怎么保证? 几种方式可以做到。 11. 有了解过qos吗? 怎么实现的。 12. 详述kube-proxy原理。 13. k8s的pause容器有什么用。是否可以去掉。 14. StatefulSets和operator区别。 	<ol style="list-style-type: none"> 1. 答: Traefik 作为一种边缘路由器, 动态感知后端服务实例变化, 进行动态调整转发配置, 会与ApiServer进行交互, 发现k8s集群内部容器状态变化。 2. aa 3. 保存所有的, (多少个历史记录就是保存多少个rs) 4. initContainer改目录权限等方式 5. raft算法 强一致性 同一时间只能有一个leader, 所有的操作都在leader上。 6. prometheus,zabbix等, 使用export提供metrics, 获取实时指标, 设计流程与思路。 7. 多个pod时, 会按照顺序(序列号)滚动更新, 每一个Pod都正常运行时才会继续处理下一个Pod。 8. scheduler 只能有一个leader, 一般有三个节点就够。 9. 10. 11. qos三种 <i>Guaranteed</i>, <i>Burstable</i>, and <i>Best-Effort</i>, 它们的QoS级别依次递减。 Guaranteed 如果Pod中所有Container的所有Resource的limit和request都相等且不大于0, 则这个Pod的QoS Class就是Guaranteed。 Best-Effort 如果Pod中所有容器的所有Resource的request和limit都没有赋值, 则这个Pod的QoS Class就是Best-Effort。
------------	----	---	---

			<p>Burstable 除了符合 Guaranteed和Best-Effort的场景, 其他场景的Pod QoS Class都属于Burstable。</p> <p>node资源不足时会按qos级别驱逐pod。 最先驱逐的是Best-Effort, 重要组件一定要设置limit和request.</p> <p>12. 监听 API server 中 service 和 endpoint 的变化情况, 并通过 iptables 等来为服务配置负载均衡</p> <p>kube-proxy的作用主要是负责service的实现 service另外一个重要作用是, 一个服务后端的Pods可能会随着生存灭亡而发生IP的改变, service的出现, 给服务提供了一个固定的IP, 而无视后端Endpoint的变化。</p> <p>kube-proxy 模式</p> <ul style="list-style-type: none"> • userspace 已弃用 • iptables 大规模下会有性能问题且不支持会话保持 • ipvs <p>13. 为pod中的多个容器提供共享网络空间, 实现pod里容器间的通信。不可以去掉。</p> <p>主要有两个职责：</p> <ol style="list-style-type: none"> a. 是pod里其他容器共享Linux namespace的基础 b. 扮演PID 1的角色, 负责处理僵尸进程 <p>14. StatefulSet是为了解决有状态服务的。opertator用来扩展k8s api,自定义crd。</p>	