

[原创](#) [推荐](#)

总结一下：运维工程师面试的经历及面试相关问题（会持续更新）

甘兵 [关注](#)

2018-01-04 19:40:43 104280人阅读 80人评论

2018年1月4号面试

笔者其实没有想到去面试，只是在智联上更新了一下简历，就陆陆续续接到很多猎头的邮件和电话，实在是没准备好要去面试，就推掉了几家公司的面试了。正因为笔者也很久没有面试了，笔者也想去面试学习一下，闲话少说，下面就分享给大家笔者在2018年1月4号上午10点30分的面试经历：

首先，猎头或者公司人资会把公司的介绍及岗位要求发到你邮箱（或者QQ、微信），下面这份是猎头发给我的岗位说明，为了职业道德操守，公司的介绍和面试通知信息我就不贴出来了，我就把岗位要求贴出来：

职位描述：

- 1、负责应用服务器的安装、配置、优化与维护；
- 2、负责应用系统的日志信息备份、管理、维护与分析；
- 3、负责应用系统的日常监测于维护、故障处理、性能分析与优化；
- 4、负责应用部署系统、环境配置系统、监控系统的开发、部署、升级与维护，建设高性能的运维平台。



岗位要求：

- 1、熟悉Linux操作系统的基础知识，熟练使用Linux常用操作命令；
- 2、熟练配置Nginx、HAProxy等应用相关软件的部署、配置与优化维护；
- 3、熟悉网络基础知识、熟悉TCP/IP的工作原理，会配交换机或路由器，能熟练的对网络情况进行分析
- 4、熟悉shell/perl/python中的一种或多种进行运维程序的开发；
- 5、熟悉Nagios,Ganglia等监控软件

看着上面的要求大家是不是觉得要求也不高啊，你要细看就会发现，这家公司要求的还挺多，不仅要会网络知识（熟悉TCP/IP好像是每家单位的都会写这样的要求），还要会开发技能。相信很多做运维的兄弟在网络这一块是个头疼的事情，都对交换机和路由器不怎么会配置和管理。

然后，笔者详细了解他们公司，了解岗位要求，在突击复习一下可能会问到的知识点和技术点。到了面试的这天时间，早早的起床，把牙一定要刷干净，特别是有口臭的兄弟，最好准备点口香糖，到达面试公司前嚼块口香糖，以免因为口气的原因熏到面试官，让你在面试官心里减分。早要记得吃，如果你是下午面试的话也要吃午饭，吃早点了精气神就有了。还要注意，带上你的简历和一支笔，虽然他们那边也会有你的简历，为了以防万一还是准备好简历。

[在线客服](#)

两个问题实在想不起来了，如果大家有更恰当的回答一定要贴出来一起探讨和进步：

1、介绍下自己？（几乎每家公司首先都会让你做个自我介绍，好像是必修课一样）

笔者回答：此处省略笔者的自我介绍，笔者建议介绍自己的时间不宜过长，3-4分钟为宜，说多了面试官会觉得你太啰嗦了。说太少了也不行，那样会让人感觉你的经历太简单了、太空了。正常情况下，一般你在做自我介绍的同时，面试官这个时候在看你的简历，他需要一边看简历、一边听你介绍自己，如果你说个几句话就把自己介绍完了，他肯定还没缓过神来，对你的映像会减分的。在介绍的同时思维要清晰，逻辑要清楚，最好是根据你简历上写的经历来介绍，这样可以把面试官的思路带到你这里来，让他思路跟着你走。不要东扯一句，西扯一句。尽量少介绍自己的性格、爱好（最好能不说就不说），你可以简单罗列干过几家公司（最多罗列3家公司/也包含目前所在的公司，注意顺序不要乱），都在那几家公司负责什么工作，都用过什么技术，在着重介绍一下你目前所在的公司是负责哪些工作的，可以稍微详细一点介绍，不要让面试官听着晕头转向的感觉。

2、灰度发布如何实现？

笔者回答：其实对这个问题笔者也答的不好，就不写出来误导大家了。大家有好的方法可以共享出来。不过笔者事后在知乎上看到了一位网友的建议觉得不错，大家可以参考看一下：<https://www.zhihu.com/question/20584476>

3、Mongodb熟悉吗，一般部署几台？

笔者回答：部署过，没有深入研究过，一般mongodb部署主从、或者mongodb分片集群；建议3台或5台服务器来部署。MongoDB分片的基本思想就是将集合切分成小块。这些块分散到若干片里面，每个片只负责总数据的一部分。对于客户端来说，无需知道数据被拆分了，也无需知道服务端哪个分片对应哪些数据。数据在分片之前需要运行一个路由进程，进程名为mongos。这个路由器知道所有数据的存放位置，知道数据和片的对应关系。对客户端来说，它仅知道连接了一个普通的mongod，在请求数据的过程中，通过路由器上的数据和片的对应关系，路由到目标数据所在的片上，如果请求有了回应，路由器将其收集起来回送给客户端。



4、如何发布和回滚，用jenkins又是怎么实现？

笔者回答：发布：jenkins配置好代码路径（SVN或GIT），然后拉代码，打tag。需要编译就编译，编译之后推送到发布服务器（jenkins里面可以调脚本），然后从分发服务器往下分发到业务服务器上。

回滚：按照版本号到发布服务器找到对应的版本推送

5、Tomcat工作模式？

笔者回答：Tomcat是一个JSP/Servlet容器。其作为Servlet容器，有三种工作模式：独立的Servlet容器、进程内的Servlet容器和进程外的Servlet容器。

进入Tomcat的请求可以根据Tomcat的工作模式分为如下两类：

Tomcat作为应用程序服务器：请求来自于前端的web服务器，这可能是Apache, IIS, Nginx等；

Tomcat作为独立服务器：请求来自于web浏览器；

6、监控用什么实现的？

笔者回答：现在公司的业务都跑在阿里云上，我们首选的监控就是用阿里云监控，阿里云监控自带了ECS、RDS等服务的监控模板，可结合自定义报警规则来触发监控项。上家公司的业务是托管在IDC，用的是zabbix监控方案，zabbix图形界面丰富，也自带很多监控模板，特别是多个分区、多个网卡等自动发现并进行监控做得非常不错，不过需要在每台客户机（被监控端）安装zabbix agent。

在线客服

笔者回答：在生产环境下，不管是应用数据、还是数据库数据首先在部署的时候就会有主从架构、或者集群，这本身就是属于数据的热备份；其实考虑冷备份，用专门一台服务器做为备份服务器，比如可以用rsync+inotify配合计划任务来实现数据的冷备份，如果是发版的包备份，正常情况下有台发布服务器，每次发版都会保存好发版的包。

总结一下面试注意几点事项，可能笔者也说得不太对，为了我们运维工作的兄弟们都能拿到高薪，大家一定要指证出来一起进步、一起探讨：

第一，你要对自己的简历很熟悉，简历上的写的技能自己一定要能说出个一二，因为面试官的很多问题都会挑你简历上写的问。比如你简历上写了这么一条技能“熟悉mysql数据库的部署安装及原理”。你既然写了这么一条技能，你在怎么不熟悉你也要了解mysql的原理，能说出个大概意思。万一面试官问到了你写的这一条，你都答不上来，那在他心里你又减分了，基本上这次面试希望不大。

第二，如果面试官问到你不会的问题，你就说这个不太熟悉，没有具体研究过，千万别不懂装懂，还扯一堆没用的话题来掩饰，这样只会让面试官反感你。

第三，准备充分，竟可能多的记住原理性的知识，一般面试问的多的就是原理。很少问具体的配置文件是怎么配置的。面试前也要了解清楚“职位描述”和“岗位要求”，虽然有时候大多数不会问到岗位要求的问题，但也要了解和熟悉。

第四，面试完后一定要总结，尽量记住面试官问的每一个问题，回去记录下来，如果问到不会的问题，事后要立马查百度或者找朋友搞清楚、弄明白，这样你才能记劳，下次面试说不定又问到同样的问题。

问完之后，面试官就跟我聊薪资待遇了，问我多少钱能达到自己的要求，我就不便透露了，可以私聊，哈哈，后续笔者会陆陆续续更新以前面试的经历和问题，有需要的朋友可以转载或者收藏起来一起讨论。

2017年2月24号面试



基于大家热情高昂的气氛，笔者又花了一个下午的时间回忆并整理在2017年2月24号笔者在东三环边上（快到东四环了，没有地铁过去，到了四惠还要转公交车）的一家传媒公司的面试经历，还好笔者有做笔记的习惯，把之前面试的问题都记录在案，这一次的面试笔者可是记忆犹新，因为这次这家公司都跟笔者发offer了，实在是真心不想去这家公司就找原因推掉了，大家可别学我这么不靠谱。下面是这家公司中的岗位要求说明：

岗位职责：

- 1、负责公司产品的版本控制、构建和发布管理；
- 2、负责公司统一配置库管理工作，权限管理与分配准确及时，定期完成配置备份；
- 3、负责公司内部开发/测试服务器的运行管理工作；
- 4、负责Linux操作系统的安装、配置、监控和维护、问题处理、软件升级、数据备份、应急响应、故障排除等、保证线上环境的稳定运行；
- 5、负责支撑平台24×7稳定运行，并进行前瞻性容量规划；
- 6、负责公司机房服务器日常维护及网络系统安装、部署、维护工作。

岗位要求：

- 1、计算机相关专业本科及以上学历，2年以上运维或配置管理工作经验；
- 2、至少熟悉一种监控系统搭建，如Nagios/Zabbix等；
- 3、至少熟悉一种集群管理工具，如Ansible/SaltStack等；
- 4、有使用集成发布工具发布构建经验优先。比如：bamboo或者Jenkins；
- 5、熟悉Unix/Linux操作系统，熟悉Weblogic/tomcat等中间件，能够编写shell脚本，熟悉软件开发过程及过程产品，有一定的网络基础；
- 6、熟悉rsyslog, flume等日志收集和处理系统；
- 7、具有强烈的安全意识及较强的沟通协调和学习能力，良好的团队合作精神，工作积极主动。

在线客服

0)，他说他是负责运维部的，然后开始就叫我先自我介绍，都是一个套路，免不了介绍的，所以兄弟们一定要把自我介绍练好。然后开始问我问题了，跟面试官聊得还行，问我应该有不下10个以上的问题，我记住了下面有10个问题：

1、LVS负载的原理，和Nginx负载有啥区别？

笔者回答：这个问题我觉得面试官问没问好，正常都会这么问“LVS有哪些负载均衡技术和调度算法？”。我回答就是按我说的这种问法回答的，反正他也频繁点头，当然，笔者回答的可能没有下面我整理出来的那么详细，大概意思我都明白了。

LVS是Linux虚拟服务器的简称，利用LVS提供的负载均衡技术和linux操作系统可实现高性能、高可用的服务器集群，一般LVS都是位于整个集群系统的最前端，由一台或者多台负载调度器（Director Server）组成，分发给应用服务器（Real Server）。它是工作在4层（也就是TCP/IP中的传输层），LVS是基于IP负载均衡技术的IPVS模块来实现的，IPVS实现负载均衡机制有三种，分别是NAT、TUN和DR，详述如下：

▣ **VS/NAT**：即（Virtual Server via Network Address Translation）

也就是网络地址翻译技术实现虚拟服务器，当用户请求到达调度器时，调度器将请求报文的目标地址（即虚拟IP地址）改写成选定的Real Server地址，同时报文的目标端口也改成选定的Real Server的相应端口，最后将报文请求发送到选定的Real Server。在服务器端得到数据后，Real Server返回数据给用户时，需要再次经过负载调度器将报文的源地址和源端口改成虚拟IP地址和相应端口，然后把数据发送给用户，完成整个负载调度过程。

可以看出，在NAT方式下，用户请求和响应报文都必须经过Director Server地址重写，当用户请求越来越多时，调度器的处理能力将称为瓶颈。

▣ **VS/TUN**：即（Virtual Server via IP Tunneling）

也就是IP隧道技术实现虚拟服务器。它的连接调度和管理与VS/NAT方式一样，只是它的报文转发方法不同，VS/TUN方式中，调度器采用IP隧道技术将用户请求转发到某个Real Server，而这个Real Server将直接响应用户的请求，不再经过前端调度器，此外，对Real Server的地域位置没有要求，可以和Director Server位于同一个网段，也可以是独立的一个网络。因此，在TUN方式中，调度器将只处理用户的报文请求，集群系统的吞吐量大大提高。

▣ **VS/DR**：即（Virtual Server via Direct Routing）

也就是用直接路由技术实现虚拟服务器。它的连接调度和管理与VS/NAT和VS/TUN中的一样，但它的报文转发方法又有不同，VS/DR通过改写请求报文的MAC地址，将请求发送到Real Server，而Real Server将响应直接返回给客户，免去了VS/TUN中的IP隧道开销。这种方式是三种负载调度机制中性能最高最好的，但是必须要求Director Server与Real Server都有一块网卡连在同一物理网段上。

回答负载调度算法，IPVS实现在八种负载调度算法，我们常用的有四种调度算法（**轮叫调度、加权轮叫调度、最少链接调度、加权最少链接调度**）。一般说了这四种就够了，也不会需要你详细解释这四种算法的。你只要把上面3种负载均衡技术讲明白面试官就对这道问题很满意了。接下来你在简单说下与nginx的区别：

LVS的优点：

- 抗负载能力强、工作在第4层仅作分发之用，没有流量的产生，这个特点也决定了它在负载均衡软件里的性能最强的；无流量，同时保证了均衡器IO的性能不会受到大流量的影响；
- 工作稳定，自身有完整的双机热备方案，如LVS+Keepalived和LVS+Heartbeat；
- 应用范围比较广，可以对所有应用做负载均衡；
- 配置性比较低，这是一个缺点也是一个优点，因为没有可太多配置的东西，所以并不需要太多接触，大大减少了人为出错的几率。

LVS的缺点：

- 软件本身不支持正则处理，不能做动静分离，这就凸显了Nginx/HAProxy+Keepalived的优势。
- 如果网站应用比较庞大，LVS/DR+Keepalived就比较复杂了，特别是后面有Windows Server应用的机器，实施及配置还有维护过程就比较麻烦，相对而言，Nginx/HAProxy+Keepalived就简单一点

Nginx的优点：



在线
客服

- Nginx对网络的依赖非常小，理论上能ping通就能进行负载功能，这个也是它的优势所在；
- Nginx安装和配置比较简单，测试起来比较方便；
- 可以承担高的负载压力且稳定，一般能支撑超过几万次的并发量；
- Nginx可以通过端口检测到服务器内部的故障，比如根据服务器处理网页返回的状态码、超时等等，并且会把返回错误的请求重新提交到另一个节点；
- Nginx不仅仅是一款优秀的负载均衡器/反向代理软件，它同时也是功能强大的Web应用服务器。LNMP现在也是非常流行的web环境，大有和LAMP环境分庭抗礼之势，Nginx在处理静态页面、特别是抗高并发方面相对apache有优势；
- Nginx现在作为Web反向加速缓存越来越成熟了，速度比传统的Squid服务器更快，有需求的朋友可以考虑用其作为反向代理加速器；

Nginx的缺点：

- Nginx不支持url来检测。
- Nginx仅能支持http和Email，这个它的弱势。
- Nginx的Session的保持，Cookie的引导能力相对欠缺。

2、redis集群的原理，redis分片是怎么实现的，你们公司redis用在了哪些环境？

笔者回答：reids集群原理：

其实它的原理不是三两句话能说明白的，redis 3.0版本之前是不支持集群的，官方推荐最大的节点数量为1000，至少需要3(Master)+3(Slave)才能建立集群，是无中心的分布式存储架构，可以在多个节点之间进行数据共享，解决了Redis高可用、可扩展等问题。集群可以将数据自动切分(split)到多个节点，当集群中的某一个节点故障时，redis还可以继续处理客户端的请求。

redis分片：

分片(partitioning)就是将你的数据拆分到多个 Redis 实例的过程，这样每个实例将只包含所有键的子集。当数据量大时，把数据分散存入多个数据库中，减少单节点的连接压力，实现海量数据存储。分片部署方式一般分为以下三种：

- （1）在客户端做分片；这种方式在客户端确定要连接的redis实例，然后直接访问相应的redis实例；
- （2）在代理中做分片；这种方式中，客户端并不直接访问redis实例，它也不知道自己要访问的具体是哪个redis实例，而是由代理转发请求和结果；其工作过程为：客户端先将请求发送给代理，代理通过分片算法确定要访问的是哪个redis实例，然后将请求发送给相应的redis实例，redis实例将结果返回给代理，代理最后将结果返回给客户端。
- （3）在redis服务器端做分片；这种方式被称为“查询路由”，在这种方式中客户端随机选择一个redis实例发送请求，如果所请求的内容不再当前redis实例中它会负责将请求转交给正确的redis实例，也有的实现中，redis实例不会转发请求，而是将正确redis的信息发给客户端，由客户端再去向正确的redis实例发送请求。

redis用在了哪些环境：

java、php环境用到了redis，主要缓存有登录用户信息数据、设备详情数据、会员签到数据等

80

分享



3、你会怎么统计当前访问的IP，并排序？

笔者回答：统计用户的访问IP，用awk结合uniq、sort过滤access.log日志就能统计并排序好。一般这么回答就够了，当然你还可以说出其它方式来统计，这都是你的加分项。

4、你会使用哪些虚拟化技术？

笔者回答：vmware vsphere及kvm，我用得比较多的是vmware vsphere虚拟化，基本上生产环境都用的vmware vsphere，kvm我是用在测试环境中使用。vmware 是属于原生架构虚拟化技术，也就是可直接在硬件上运行。kvm属于寄居架构的虚拟化技术，它是依托在系统之上运行。vmware vcenter

管理上比较方便，图形管理界面功能很强大，稳定性强，一般比较适合企业使用。KVM管理界面稍差点，需

在线
客服

5、假如有人反应，调取后端接口时特别慢，你会如何排查？

笔者回答：其实这种问题都没有具体答案，只是看你回答的内容与面试官契合度有多高，能不能说到他想要的点上，主要是看你排查问题的思路。我是这么说的：问清楚反应的人哪个服务应用或者页面调取哪个接口慢，叫他把页面或相关的URL发给你，首先，最直观的分析就是用浏览器按F12，看下是哪一块的内容过慢（DNS解析、网络加载、大图片、还是某个文件内容等），如果有，就对症下药去解决（图片慢就优化图片、网络慢就查看内网情况等）。其次，看后端服务的日志，其实大多数的问题看相关日志是最有效分析，最好用tail -f跟踪一下日志，当然你也要点击测试来访问接口日志才会打出来。最后，排除sql，，找到sql去mysql执行一下，看看时间是否很久，如果很久，就要优化SQL问题了，expain一下SQL看看索引情况啥的，针对性优化。数据量太大的能分表就分表，能分库就分库。如果SQL没啥问题，那可能就是写的逻辑代码的问题了，一行行审代码，找到耗时的地方改造，优化逻辑。

6、mysql数据库用的是主从读写分离，主库写，从库读，假如从库无法读取了、或者从库读取特别慢，你会如何解决？

笔者回答：这个问题笔者觉得回答的不太好，对mysql比较在行的朋友希望能给点建议。以解决问题为前提条件，先添加从库数量，临时把问题给解决，然后抓取slow log，分析sql语句，该优化就优化处理。慢要不就是硬件跟不上，需要升级；要不就是软件需要调试优化，等问题解决在细化。

7、cpu单核和多核有啥区别？

笔者回答：很少有面试官会问这样的问题，既然问到了，也要老实回答。还好笔者之前了解过CPU，我是这么说的：双核CPU就是能处理多份任务，顺序排成队列来处理。单核CPU一次处理一份任务，轮流处理每个程序任务。双核的优势不是频率，而是对付同时处理多件事情。单核同时只能干一件事，比如你同时在后台B T下载，前台一边看电影一边拷贝文件一边QQ。

8、机械磁盘和固态硬盘有啥区别？

笔者回答：我擦，啥年代了，还问磁盘的问题，这面试官有点逗啊。那也要回答啊：

HDD代表机械硬盘，SSD代表固态硬盘。首先，从性能方面来说，固态硬盘几乎完胜机械硬盘，固态硬盘的读写速度肯定要快机械硬盘，因为固态硬盘和机械硬盘的构造是完全不同的（具体的构造就没必要解释了）。其次，固态盘几乎没有噪音、而机械盘噪音比较大。还有就是，以目前的市场情况来看，一般机械盘容量大，价格低；固态盘容量小，价格偏高。但是企业还是首选固态盘。

9、说一下用过哪些监控系统？

笔者回答：这个监控的问题又问到了，笔者在2018年1月4号也被问到类似这样的问题，笔者曾经用过zabbix、nagios、cacti等。但是在这次面试中只说过zabbix和nagios。说完了之后，面试官就让我说一下这两个监控有啥区别：

从web功能及画图来讲：

Nagios简单直观，报警与数据都在同一页面，红色即为问题项。Nagios web端不要做任何配置。Nagios需要额外安装插件，且插件画图不够美观。

Zabbix监控数据与报警是分开的，查看问题项需要看触发器，查看数据在最新数据查看。而且zabbix有很多其它配置项，zabbix携带画图功能，且能手动把多个监控项集在一个图中展示。

从监控服务来讲：

Nagios自带的监控项很少。对一些变动的如多个分区、多个网卡进行监控时需要手动配置。

Zabbix自带了很多监控内容，感觉zabbix一开始就为你做了很多事，特别是对多个分区、多个网卡等自动发现并进行监控时，那一瞬间很惊喜，很省心的感觉。

从批量配置和报警来讲：



Nagios对于批量监控主机，需要用脚本在server端新增host，并拷贝service文件。 Nagios用脚本来修改所有主机的services文件，加入新增服务。

Zabbix在server端配置自动注册规则，配置好规则后，后续新增client端不需要对server端进行操作。 Zabbix只需手动在模板中新增一监控项即可。

总体来讲：

Nagios要花很多时间写插件，Zabbix要花很多时间探索功能。

Nagios更易上手，Nagios两天弄会，Zabbix两周弄会。

Zabbix画图功能比Nagios更强大

Zabbix对于批量监控与服务更改，操作更简洁；Nagios如果写好自动化脚本后，也很简单，问题在于写自动化脚本很费神。

10、给你一套环境，你会如何设计高可用、高并发的架构？

笔者回答：

如果这套环境是部署在云端(比如阿里云)，你就不用去考虑硬件设计的问题。可直接上阿里云的SLB+ECS+RDS这套标准的高可用、高并发的架构。对外服务直接上SLB负载均衡技术，由阿里的SLB分发到后端的ECS主机；ECS主机部署多台，应用拆分在不同的ECS主机上，尽量细分服务。数据库用RDS高可用版本（一主一备的经典高可用架构）、或者用RDS金融版（一主两备的三节点架构）。在结合阿里其它的服务就完全OK，业务量上来了，主机不够用了，直横向扩容ECS主机搞定。

如果这套环境托管在IDC，那么你就从硬件、软件（应用服务）双面去考虑了。硬件要达到高可用、高并发公司必须买多套网络硬件设备（比如负载均衡F5、防火墙、核心层交换、接入层交换）都必须冗余，尤其是在网络设计上，设备之间都必须有双链连接。设备如果都是跑的单机，其中一个设备挂了，你整个网络都瘫痪了，就谈不上高可用、高并发了。其次是在考虑应用服务了，对外服务我会采用成熟的开源方案LVS+Keepalived或者Nginx+Keepalived，缓存层可以考虑redis集群及Mongodb集群，中间件等其它服务可以用kafka、zookeeper，图片存储可以用fastDFS或MFS，如果数据量大、又非常多，那么可采用hadoop这一套方案。后端数据库可采用“主从+MHA”。这样一套环境下来是绝对满足高可用、高并发的架构。



对了，在下周一（2018年1月8号）有个海外的电话面试、还有下周二（2018年1月9号）有个网络运维经理岗位的面试，其实这个网络运维经理职位的面试有点不太想去，主要是想了解一下都会问些啥问题，可以为大家分享出来，所以请大家敬请期待吧。。。

持续更新中.....

这篇文章实在是有点长，挤不下了，大家看后期更新的内容可以跳转到 [总结一下：运维工程师面试的经历及面试相关问题（续2）](#)

喜欢我的文章，请点击最上方右角处的《关注》支持一下！

在线客服

©著作权归作者所有：来自51CTO博客作者甘兵的原创作品，如需转载，请注明出处，否则将追究法律责任

赞赏

0人进行了赞赏支持

运维面试

面试

运维工程

Linux

80

收藏

分享

上一篇：nginx用反向代理机制解决跨域... 下一篇：总结一下：运维工程师面试的经历及...



甘兵

120篇文章，202W+人气，873粉丝

只写看得懂的文章

关注



提问和评论都可以，用心的回复会被更多人看到和认可

Ctrl+Enter 发布

取消

发布



80条评论

按时间正序 | 按时间倒序



静态路由

1楼 2018-01-04 22:17:49

2 5

干货分享！笔者在哪里工作啊？

作者 甘兵:@静态路由

谢谢，在北京一家石油单位工作，兄弟在哪高就。

2018-01-05 09:14:11 回复

静态路由:@甘兵

深圳

2018-01-05 14:20:20 回复

作者 甘兵:@静态路由

也想过去深圳工作，我老家在江西，离深圳也不远

2018-01-05 15:33:18 回复

还有2条回复



蓝天飞飞

2楼 2018-01-05 11:05:41

1 1

查看由该作者回复的所有评论

72 80

甘兵

关注

在线客服

总结一下：运维工程师面试的经历及面试相关问题（会持续更新）-DevOps(甘兵)-51CTO博客

谢谢认可，昨天面试完确实琢磨了一个下午，一边回想面试的过程，一边修改文章。

2018-01-05 12:48:50 回复



sery
3楼 2018-01-05 13:28:28

1

好文章，思路清晰，条理清楚，实事求是，大赞。

作者 甘兵:@sery
谢谢兄弟认可

2018-01-05 15:30:23 回复



土星王
4楼 2018-01-05 15:35:34

1

楼主工作多久了？看着懂得挺多啊

作者 甘兵:@土星王
工作了好些年了

2018-01-06 22:23:38 回复



静思知意
5楼 2018-01-06 20:26:31

1

谢谢分享

作者 甘兵:@静思知意
您的关注和点赞是我最大的动力，哈哈

2018-01-06 22:23:12 回复



Willson_luo
6楼 2018-01-08 00:04:34

1



好厉害的文章，拜读一下。

作者 甘兵:@Willson_luo
共同学习，谢谢

2018-01-08 18:55:02 回复



北座城市
7楼 2018-01-08 10:04:05

1

满满的干货

作者 甘兵:@北座城市
谢谢，一起进步

2018-01-08 18:55:20 回复



yeyinghua
8楼 2018-01-09 09:59:10

1

干货分享！谢谢！

作者 甘兵:@yeyinghua
谢谢认可，希望能指正写得不足的地方

2018-01-09 10:46:10 回复



我不是瘦子
9楼 2018-01-09 16:31:52

1

1

感谢，最近也要开始找工作了，我能加你qq私聊么？

在线客服



金小样

10楼 2018-01-10 11:56:40

1

感谢分享

作者

甘兵:@金小样

不客气，谢谢支持，保持住你的好奇心，继续关注文章哦

2018-01-10 14:49:09 回复

推荐专栏



基于Python的DevOps实战

自动化运维开发新概念

共20章 | 抚琴煮酒

¥51.00 301人订阅

订 阅



全局视角看大型园区网

路由交换+安全+无线+优化+运维

共40章 | 51CTO夏杰

¥51.00 781人订阅

订 阅



网工2.0晋级攻略 ——零基础入门Python/A...

网络工程师2.0进阶指南

共30章 | 姜汁啤酒

¥51.00 1175人订阅

订 阅



负载均衡高手炼成记

高并发架构之路

共15章 | sery

¥51.00 401人订阅

订 阅



带你玩转高可用

前百度高级工程师的架构高可用实战

共15章 | 曹林华

¥51.00 394人订阅

订 阅

猜你喜欢

- 【企业实战】：阿里云高可用架构之“CDN+WAF+SLB+...”
你必须知道的python运维常用脚本!(日常更新)
[实战] 基于Docker的EFK日志系统与使用经验(一)-EFK...
Prometheus配合Alertmanager报警系统
- (CC)与(WAF)之间的较量
海量日志下的日志架构优化：filebeat+logstash+kafka+...
Docker-compose一键部署gitlab中文版
22世纪真实链路式监控 设计理念分析

在线客服