

作者：李振良

## 职业发展

---

### 技术专家

---

发展规划：初中级工程师 -> 高级工程师 -> 架构师 -> 专家

适宜人群：比较喜欢挑战，热爱技术，有较强钻研精神，在某一领域有深入的理解，性格比较内向。

### 技术管理

---

发展规划：初中级工程师 -> 高级工程师 -> 主管/经理 -> 总监 -> CTO（首席技术官）

适宜人群：技术知识面广，有一些管理思维，善于交际，表达沟通能力强，经常关注行业内动态和主流技术。

## 目标

---

- 熟悉运维技术学习路线及相关技能
  - 熟悉职业发展方向及面试分析
- 

## 初级

---

### 主要工作

---

- 修电脑，设备巡检
- 服务器上下架
- 网络服务部署
- 网站平台搭建与维护

### 1、Linux基础

---

刚开始阶段需要熟悉Linux操作系统安装，目录结构、启动流程等。

### 2、系统管理

---

主要学习Linux系统，生产环境中基本都在字符界面完成工作，所以要掌握常用的几十个基本管理命令，包括用户管理、磁盘分区、软件包管理、文件权限、文本处理、进程管理、性能分析工具等。

### 3、网络基础

---

OSI和TCP/IP模型一定要熟悉。基本的交换机、路由器概念及实现原理要知道。

## 4、Shell脚本编程基础

---

掌握Shell基本语法结构，能编写简单的脚本即可。

---

# 中级

---

## 主要工作

---

- 集群网站平台搭建
- 数据库基本管理
- 监控系统搭建
- 脚本化工作

## 1、网络服务

---

常见的网络服务要会部署，比如vsftp、nfs、samba、bind、dhcp等。代码版本管理系统少不了，可以学习下主流的GIT，能部署和简单使用就可以了。经常在服务器之间传输数据，所以要会使用：rsync和scp。数据同步：inotify/sersync。重复性完成一些工作，可写成脚本定时去运行，所以得会配置Linux下的定时任务服务cron。

## 2、Web服务

---

每个公司基本都会有网站，能让网站跑起来，就需要搭建Web服务平台了。如果是用PHP语言开发的，通常搭建LNMP网站平台，这是一个技术名词组合的拼写，分开讲就是得会部署Nginx、MySQL和PHP。如果是JAVA语言开发的，通常使用Tomcat运行项目，为了提高访问速度，可以使用Nginx反向代理Tomcat，Nginx处理静态页面，Tomcat处理动态页面，实现动静分离。不是会部署这么简单，还要知道HTTP协议工作原理、简单的性能调优。

## 3、负载均衡器

---

单台服务器终究资源有限，抵抗高访问量肯定是无法支撑的，解决此问题最关键的技术就是采用负载均衡器，水平扩展多台Web服务器，同时对外提供服务，这样就成倍扩展性能了。负载均衡器主流开源技术有LVS、HAProxy和Nginx。一定要熟悉一两个！

## 4、数据库

---

数据库选择MySQL，它是世界上使用最为广泛的开源数据库。学它准没错！也要会一些简单的SQL语句、用户管理、常用存储引擎、数据库备份与恢复。想要深入点，必须会主从复制、性能优化、主流集群方案：MHA、MGR等。NoSQL这么流行当然也少不了，学下Redis、MongoDB这两个就好了。

## 5、监控系统

---

监控必不可少，是及时发现问题和追溯问题的救命稻草。可以选择学习主流的Zabbix、Prometheus开源监控系统，功能丰富，能满足企业级监控需求。监控点包括服务器硬件、服务器性能、API、业务、PV/UV、日志等方面。也可以弄个仪表盘展示几个实时关键的数据，比如Grafana，会非常炫酷。

## 6、日志分析系统

---

日志也很重要，定期的分析，可发现潜在隐患，提炼出有价值的东西。主流日志系统：ELK Stack 学会部署使用，能分析日志并可视化，方便故障排查。

## 7、安全防范

---

安全很重要，不要等到系统被攻击了，再做安全策略，此时已晚！所以，当一台服务器上线后应马上做安全访问控制策略，比如使用iptables限制只允许信任源IP访问，关闭一些无用的服务和端口等。一些常见的攻击类型一定得知道啊，否则怎么对症下药呢！比如CC、DDOS、ARP等。

## 8、Shell脚本编程进阶

---

Shell脚本是Linux自动完成工作的利器，必须得熟练编写，所以得进一步学习函数、数组、信号、发邮件等。文本处理三剑客（grep、sed、awk）得玩6啊，Linux下文本处理就指望它们了。

## 9、Python/Go开发基础

---

Shell脚本只能完成一些基本的任务，想要完成更复杂些的任务，比如调用API、多进程等。就需要学高级语言了。Python是运维领域使用最多的语言，简单易用，学它准没错！此阶段掌握基础就可以了，例如基本语法结构、文件对象操作、函数、迭代对象、异常处理、发邮件、数据库编程等。

# 高级

---

## 主要工作

---

- 亿级网站架构设计
- 性能优化
- 容器化落地
- 大规模自动化运维
- 平台化工作

## 1、Web静态缓存

---

用户老喊着访问网站慢，看看服务器资源还很富裕啊！网站访问慢也许不是服务器资源饱和导致的，影响因素很多，例如网络、转发层数等。对于网络，存在南北通信问题，之间访问会慢，这个可以使用CDN解决，同时缓存静态页面，尽可能将请求拦截在最上层响应，减少后端请求和响应时间。如果不用CDN，也可以使用Squid、Varnish、Nginx这样的缓存服务实现静态页面缓存，放到流量入口处。

## 2、集群

---

单台服务器终究资源有限，抵抗高访问量肯定无法支撑的，解决此问题最关键的技术就是采用负载均衡器，水平扩展多台Web服务器，同时对外提供服务，这样就成倍扩展性能了。负载均衡器主流开源技术有LVS、HAProxy和Nginx。一定要熟悉一两个！Web服务器性能瓶颈解决了，数据库更为关键，还是采用集群，就拿MySQL来说，可以一主多从架构，在此基础上读写分离，主负责写，多从负责读，从库可水平扩展，前面再来个四层负载均衡器，承载千万级PV，妥妥的！高可用软件也得会，避免单点的利器，主流的有Keepalived、Heartbeat等。

随着业务扩展，网站图片越来越多！NFS共享存储支撑不过了，处理很慢！上分布式文件系统，并行处理任务，无单点，高可靠，高性能等特性，主流的有FastDFS、MFS、HDFS、Ceph、GFS等。初期的话我建议学习下FastDFS，能满足中小规模需求。

### 3、容器化

---

硬件服务器资源利用率很低，甚是浪费！可以把空闲多的服务器虚拟化，弄成很多个的虚拟机，每个虚拟机就是一个完整的操作系统。可以很大程度提高资源利用率。主流私有云平台实现：KVM+OpenStack 虚拟机作为基础平台还可以，但应用业务弹性伸缩也太重量了吧！启动好几分钟，文件又这么大，快速扩展太费劲了！好说，上容器，容器主要特点就是快速部署和环境隔离。一个服务封装到镜像中，分分钟钟可创建几百个容器。主流的容器技术非Docker莫属了。当然，生产环境单机Docker大多数情况下是无法满足业务需求的，可以基于Kubernetes构建容器平台，集群化管理容器，形成一个大的资源池，为基础架构提供有力的支撑。

### 4、自动化运维

---

反反复复重复的工作，不但提高不了效率，价值也得不到体现。一切运维工作标准化，例如环境版本、目录结构、操作系统等统一。在标准化基础上才能更方面的自动化，点点鼠标或者敲几个命令即可完成一项复杂的工作任务，爽哉爽哉！因此，所有的操作尽可能自动化，减少人为失误，提高工作效率。主流服务器集中管理工具：Ansible、Saltstack 持续集成工具：Jenkins、Gitlab

### 5、Python开发进阶

---

可以再深入学习下Python开发，掌握面向对象编程。最好也掌学习一个Web框架开发网站，例如Django、Flask，主要是开发运维管理系统，将一些复杂的流程写到平台中，再集成集中管理工具，可打造一个属于运维自己的管理平台。

### 7、性能优化

---

只会部署是远远不够的，性能优化能最大化提升服务承载量。这块也是比较难的，也是高薪的关键点之一，为了钱也得下点功夫学习啊！可以从硬件层、操作系统层、软件层和架构层维度展开思考。

### 小结

---

武器类型	武器名称
文本处理	grep、sed、awk
数据传输	Rsync、Scp、Inotify/Sersync
进程管理	Supervisor
性能分析	top、free、df、iftop、iostat、vmstat、dstat、sar、sysdig
网络服务	vsftp、nfs、samba、bind、dhcp、postfix
Web服务	Apache、 <b>Nginx</b> 、 <b>Tomcat</b> 、JBoss、Resin
数据库	<b>MySQL</b> 、MariaDB、PostgreSQL、Oracle
NoSQL	Redis、MongoDB
消息中间件	RabbitMQ、ActiveMQ
版本管理	SVN、 <b>Git</b>
静态缓存	Squid、Varnish、 <b>Nginx</b>
负载均衡	<b>LVS</b> 、HAProxy、 <b>Nginx</b>
高可用软件	<b>Keepalived</b> 、Heartbeat、DRBD、corosync+pacemaker
集中管理工具	<b>Ansible</b> 、Saltstack、Chef、Puppet
虚拟化	KVM、Xen、Openstack、Cloudstack
容器化	<b>Docker</b> 、 <b>Kubernetes</b> 、Rancher、Openshift
自动装机	Kickstart、Cobbler
抓包分析	Tcpdump、Wireshark
持续集成	<b>Jenkins</b> 、Gitlab
MySQL代理	Altas、Cobar、Mycat
压测	ab、fio、sysbench、mysqlslap、Jemter
日志系统	<b>ELK Stack</b> 、Graylog
监控	Zabbix、 <b>Prometheus</b> 、Open-falcon
分布式文件系统	<b>Ceph</b> 、GlusterFS、FastDFS

## 金三银四，找工作难？

“金三银四”跳槽季，准备好换工作了吧！心里想得美滋滋，现实却残酷得赤裸裸，投了很多简历，面试率低的可怜，都说金三银四，可为什么我换工作就这么难呢？

主要原因如下：

1. 简历不够明确清晰 (10%)
2. 没掌握面试技巧 (10%)
3. 选择工作的态度 (10%)
4. 不明确自己的价值 (5%)
5. 专业技能不足 (55%)
6. 学历 (5%) 力。
7. 行业不景气：停招、裁员！ (5%)

## 意识

---

### 1、坚持

---

学习是一个很漫长的过程，是我们每个人需要用一生去坚持的事业。贵在坚持，难在坚持，成在坚持！

### 2、目标

---

没有目标的不叫工作，没有量化的不叫目标。每到一个阶段，制定一个目标。比如：先定一个能达到的小目标，挣它一个亿！

### 3、兴趣

---

如果你热爱这个行业，那它就不只是一份工作。

## 方向对了，就不怕路远了！

---

细节请看视频：<https://ke.qq.com/course/266656>