

版权注意事项：

- 1、书籍版权归作者和出版社所有
- 2、本PDF仅限用于个人获取知识，进行私底下的知识交流
- 3、PDF获得者不得在互联网上以任何目的进行传播
- 4、如觉得书籍内容很赞，请购买正版实体书，支持作者
- 5、请于下载PDF后24小时内删除本PDF。



● 视频课程 ● 案例素材 ● 交流社区 ● QQ 讨论组

大型网站架构与 自动化运维

主 编 肖 睿 罗保山 刘丽军

课工场介绍

课工场是专注互联网教育的生态平台，汇聚了中国和北美数百位来自知名互联网企业的行业大咖，向寻求就业和技术提升的人群提供直播、录播、面授等多模式教学场景，并通过遍布全国的线下服务中心提供成熟的学习服务，形成完善的“互联网+教育”解决方案。同时，课工场也为高校、企业、行业提供教育技术赋能，依托 Transform 智能教育生态平台，打造智慧校园、企业大学、行业培训的教育场景，提供一站式教育解决方案。

课工场于 2016 年荣膺新浪网“2016 中国影响力科技创新教育机构”，腾讯网“2016 中国影响力教育品牌”，网易“2016 年度最受信赖教育机构”，小米“2016 教育行业突出贡献奖”。



扫一扫关注课工场公众号
关注微信 立减204币
可购买收费课程



课工场APP客户端下载
产品/设计/开发/运维/运营
随时随地随心学

主要内容

本书以Linux为基础，介绍网络工程师必备的知识和技能。全书共分10章，主要内容包括：Linux基础、网络基础、网络配置、网络故障排除、网络性能优化、网络安全管理、网络应用开发、网络工程实践等。本书可作为网络工程、网络管理、网络工程等相关专业教材，也可供从事网络工程工作的工程技术人员参考。

云计算工程师系列

大型网站架构与自动化运维

编委：杨欢 潘贞玉 张健 祁洪波 谢国民

主 编 肖 睿 罗保山 刘丽军

董泰森 曾译译 王俊杰 (云计算工程师系列)

编委：李超阳 祁春鹏 祁 龙 滕传雨 尚永涛

张雪妮 吴宇迪 曹紫涵 吉志坚

李晓川 黄斌 宗 强 陈 强 王 强

编委：李超阳 祁春鹏 祁 龙 滕传雨 尚永涛

刘景元 袁娇娇	王 强
DAXING WANGZHAN JIAOOU YU ZIDONGHUA YUNWEI	王 强
中国水利水电出版社	发行
(北京市西城区南滨河路1号D座100078)	
网 址: www.waterpub.com.cn	
E-mail: mchance@263.net (万水)	
sales@waterpub.com.cn	
电 话: (010) 68367828 (营销中心); 82562819 (万水)	
全国各省市新华书店均有代售	售 经
北京万水电子商务有限公司	经 销
北京新华印刷有限公司	印 刷
184mm x 260mm 16开本 11.2印张 266千字	开 本
2017年2月第1版 2017年2月第1次印刷	次 数
0001—3000册	册 数
36.00元	定 价

内 容 提 要

本书针对具备 Linux 基础的人群,主要介绍了缓存代理、高性能内存对象缓存 Memcached、分布式文件服务、大型网站架构、自动化运维的相关知识与应用,以企业级的实战项目案例,使读者能够掌握应用运维的工作内容。项目案例包括 MFS 分布式文件系统、百万/千万 PV 网站架构、Ansible、SaltStack、Puppet 自动化运维,通过以上项目案例的训练,读者能够理解大型网站的架构,达到运维自动化的高度。

本书通过通俗易懂的原理及深入浅出的案例,并配以完善的学习资源和支持服务,为读者带来全方位的学习体验,包括视频教程、案例素材下载、学习交流社区、讨论组等终身学习内容,更多技术支持请访问课工场 www.kgc.cn。

图书在版编目(CIP)数据

大型网站架构与自动化运维 / 肖睿, 罗保山, 刘丽军主编. — 北京: 中国水利水电出版社, 2017.5

(云计算工程师系列)

ISBN 978-7-5170-5404-7

I. ①大… II. ①肖… ②罗… ③刘… III. ①网站—建设 IV. ①TP393.092.1

中国版本图书馆CIP数据核字(2017)第105392号

策划编辑: 祝智敏 责任编辑: 李 炎 加工编辑: 赵佳琦 封面设计: 梁 燕

书 名	云计算工程师系列 大型网站架构与自动化运维 DAXING WANGZHAN JIAGOU YU ZIDONGHUA YUNWEI
作 者	主 编 肖 睿 罗保山 刘丽军
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路1号D座 100038) 网 址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电 话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京泽宇印刷有限公司
规 格	184mm×260mm 16开本 11.5印张 266千字
版 次	2017年5月第1版 2017年5月第1次印刷
印 数	0001—3000册
定 价	36.00元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

丛书编委会

主任：肖 睿

副主任：刁景涛

委员：杨 欢 潘贞玉 张德平 相洪波 谢伟民

庞国广 张惠军 段永华 李 娜 孙 苹

董泰森 曾淳淳 王俊鑫 俞 俊

课工场：李超阳 祁春鹏 祁 龙 滕传雨 尚永祯

张雪妮 吴宇迪 曹紫涵 吉志星 胡杨柳依

李晓川 黄 斌 宗 娜 陈 璇 王博君

刁志星 孙 敏 张 智 董文治 霍荣慧

刘景元 袁娇娇 李 红 孙正哲 史爱鑫

周士昆 傅 峥 于学杰 何娅玲 王宗娟



前言

“互联网+人工智能”时代，新技术的发展可谓是一日千里，云计算、大数据、物联网、区块链、虚拟现实、机器学习、深度学习等等，已经形成一波新的科技浪潮。以云计算为例，国内云计算市场的蛋糕正变得越来越诱人，以下列举了2016年以来发生的部分大事。

1. 中国联通发布云计算策略，并同步发起成立“中国联通沃云+云生态联盟”，全面开启云服务新时代。

2. 内蒙古斥资500亿元欲打造亚洲最大云计算数据中心。

3. 腾讯云升级为平台级战略，旨在探索云上生态，实现全面开放，构建可信赖的云生态体系。

4. 百度正式发布“云计算+大数据+人工智能”三位一体的云战略。

5. 亚马逊AWS和北京光环新网科技股份有限公司联合宣布：由光环新网负责运营的AWS中国（北京）区域在中国正式商用。

6. 来自Forrester的报告认为，AWS和OpenStack是公有云和私有云事实上的标准。

7. 网易正式推出“网易云”。网易将先行投入数十亿人民币，发力云计算领域。

8. 金山云重磅发布“大米”云主机，这是一款专为创业者而生的性能王云主机，采用自建11线BGP全覆盖以及VPC私有网络，全方位保障数据安全。

DT时代，企业对传统IT架构的需求减弱，不少传统IT企业的技术人员，面临失业风险。全球最知名的职业社交平台LinkedIn发布报告，最受雇主青睐的十大职业技能中“云计算”名列前茅。2016年，中国企业云服务整体市场规模超500亿元，预计未来几年仍将保持约30%的年复合增长率。未来5年，整个社会对云计算人才的需求缺口将高达130万。从传统的IT工程师转型为云计算与大数据专家，已经成为一种趋势。

基于云计算这样的大环境，课工场（www.kgc.cn）的教研团队几年前开始策划的“云计算工程师系列”教材应运而生，它旨在帮助读者朋友快速成长为符合企业需求的、优秀的云计算工程师。这套教材是目前业界最全面、专业的云计算课程体系，能够满足企业对高级复合型人才的要求。参与编写的院校老师还有罗保山、刘丽军等。



课工场是北京大学下属企业北京课工场教育科技有限公司推出的互联网教育平台，专注于互联网企业各岗位人才的培养。平台汇聚了数百位来自知名培训机构、高校的顶级名师和互联网企业的行业专家，面向大学生以及需要“充电”的在职人员，针对与互联网相关的产品设计、开发、运维、推广和运营等岗位，提供在线的直播和录播课程，并通过遍及全国的几十家线下服务中心提供现场面授以及多种形式的教学服务，并同步研发出版最新的课程教材。

除了教材之外，课工场还提供各种学习资源和支持服务，包括：

- 现场面授课程
- 在线直播课程
- 录播视频课程
- 授课 PPT 课件
- 案例素材下载
- 扩展资料提供
- 学习交流社区
- QQ 讨论组（技术，就业，生活）

以上资源请访问课工场网站 www.kgc.cn。

本套教材特点

（1）科学的训练模式

- 科学的课程体系。
- 创新的教学模式。
- 技能人脉，实现多方位就业。
- 随需而变，支持终身学习。

（2）企业实战项目驱动

- 覆盖企业各项业务所需的 IT 技能。
- 几十个实训项目，快速积累一线实践经验。

（3）便捷的学习体验

- 提供二维码扫描，可以观看相关视频讲解和扩展资料等知识服务。
- 课工场开辟教材配套版块，提供素材下载、学习社区等丰富的在线学习资源。

读者对象

（1）初学者：本套教材将帮助你快速进入云计算及运维开发行业，从零开始逐步成长为专业的云计算及运维开发工程师。

（2）初中级运维及运维开发者：本套教材将带你进行全面、系统的云计算及运维开发学习，逐步成长为高级云计算及运维开发工程师。

课程设计说明

课程目标

读者学完本书后，能够掌握自动化运维技术，设计、实施和维护大型网站架构。

训练技能

- 理解缓存代理的工作原理，并掌握其配置。
- 理解高性能内存对象缓存 Memcached 的工作原理，并掌握其部署。
- 掌握异地备份工具 rsync 的工作原理及其相关配置。
- 掌握 MFS 工作原理、部署与灾难恢复。
- 理解百万、千万 PV 网站架构及其相关部署。
- 理解自动化运维工具原理并进行自动化配置。

设计思路

本书采用了教材 + 扩展知识的设计思路，扩展知识提供二维码扫描，形式可以是文档、视频等，内容可以随时更新，能够更好地服务读者。

教材分为 3 个阶段来设计学习过程，即代理与缓存、网站与容灾、自动化运维，具体安排如下：

- 第 1 章～第 2 章介绍缓存代理与内存缓存的相关知识，理解 Squid 代理的工作模式、Memcached 的工作原理，实现 Squid 代理和 Memcached 相关部署。
- 第 3 章～第 6 章是构建企业网站相关内容，使用 rsync 技术实现企业网站的备份，MFS 实现灾难恢复与在线扩容，进而实现百万、千万 PV 网站架构部署与管理。
- 第 7 章～第 9 章介绍企业自动化运维工具 Ansible、SaltStack、Puppet 的部署与使用。

章节导读

- 技能目标：学习本章所要达到的技能，可以作为检验学习效果的标准。
- 本章导读：对本章涉及的技能内容进行分析并展开讲解。
- 操作案例：对所学内容的实操训练。
- 本章总结：针对本章内容的概括和总结。
- 本章作业：针对本章内容的补充练习，用于加强对技能的理解和运用。

- 扩展知识：针对本章内容的扩展、补充，对于新知识随时可以更新。

学习资源

- 学习交流社区（课工场）

- 案例素材下载

- 相关视频教程

更多内容详见课工场 www.kgc.cn。



目

第1章 绪论	139
1.1 课程介绍	139
1.2 课程目标	140
1.3 课程结构	141
1.4 课程资源	142
1.5 课程评价	143
1.6 课程总结	144
第2章 高性能内存缓存	145
2.1 内存缓存	145
2.2 内存缓存的架构	146
2.3 内存缓存的部署	147
2.4 内存缓存的维护	148
2.5 内存缓存的总结	149
第3章 分布式文件系统	150
3.1 分布式文件系统的概念	150
3.2 分布式文件系统的架构	151
3.3 分布式文件系统的部署	152
3.4 分布式文件系统的维护	153
3.5 分布式文件系统的总结	154
第4章 负载均衡	155
4.1 负载均衡的概念	155
4.2 负载均衡的架构	156
4.3 负载均衡的部署	157
4.4 负载均衡的维护	158
4.5 负载均衡的总结	159
第5章 网络交换	160
5.1 网络交换的概念	160
5.2 网络交换的架构	161
5.3 网络交换的部署	162
5.4 网络交换的维护	163
5.5 网络交换的总结	164
第6章 网络路由	165
6.1 网络路由的概念	165
6.2 网络路由的架构	166
6.3 网络路由的部署	167
6.4 网络路由的维护	168
6.5 网络路由的总结	169
第7章 网络存储	170
7.1 网络存储的概念	170
7.2 网络存储的架构	171
7.3 网络存储的部署	172
7.4 网络存储的维护	173
7.5 网络存储的总结	174
第8章 网络虚拟化	175
8.1 网络虚拟化的概念	175
8.2 网络虚拟化的架构	176
8.3 网络虚拟化的部署	177
8.4 网络虚拟化的维护	178
8.5 网络虚拟化的总结	179
第9章 网络性能优化	180
9.1 网络性能优化的概念	180
9.2 网络性能优化的架构	181
9.3 网络性能优化的部署	182
9.4 网络性能优化的维护	183
9.5 网络性能优化的总结	184
第10章 网络故障排除	185
10.1 网络故障排除的概念	185
10.2 网络故障排除的架构	186
10.3 网络故障排除的部署	187
10.4 网络故障排除的维护	188
10.5 网络故障排除的总结	189
第11章 网络新技术	190
11.1 网络新技术的概念	190
11.2 网络新技术的架构	191
11.3 网络新技术的部署	192
11.4 网络新技术的维护	193
11.5 网络新技术的总结	194
第12章 网络应用	195
12.1 网络应用的概念	195
12.2 网络应用的架构	196
12.3 网络应用的部署	197
12.4 网络应用的维护	198
12.5 网络应用的总结	199
第13章 网络案例	200
13.1 网络案例的概念	200
13.2 网络案例的架构	201
13.3 网络案例的部署	202
13.4 网络案例的维护	203
13.5 网络案例的总结	204
第14章 网络总结	205
14.1 网络总结的概念	205
14.2 网络总结的架构	206
14.3 网络总结的部署	207
14.4 网络总结的维护	208
14.5 网络总结的总结	209

课程设计说明



前言

课程设计说明

第 1 章 Squid 缓存服务器	1	第 3 章 rsync 远程同步	37
1.1 Squid 服务基础	2	3.1 配置 rsync 源服务器	38
1.1.1 缓存代理概述	2	3.2 使用 rsync 备份工具	40
1.1.2 安装及运行控制	3	3.3 配置 rsync+inotify 实时同步	41
1.2 构建代理服务器	6	本章总结	44
1.2.1 传统代理	6	本章作业	44
1.2.2 透明代理	9		
1.2.3 ACL 访问控制	11	第 4 章 MFS 分布式文件系统	45
1.3 Squid 日志分析	13	4.1 案例分析	46
1.4 Squid 反向代理	15	4.2 案例实施	48
1.5 Varnish 与 Nginx 缓存服务器	16	本章总结	59
本章总结	17	本章作业	59
本章作业	17		
第 2 章 高性能内存对象缓存		第 5 章 部署社交网站	61
Memcached	19	5.1 案例分析	62
2.1 认识 Memcached	20	5.1.1 案例概述	62
2.2 安装 Memcached 案例	23	5.1.2 案例环境	62
2.2.1 安装 Memcached 服务器	23	5.2 案例实施	63
2.2.2 Memcached API 客户端	24	5.3 案例扩展	67
2.3 Memcached 数据库操作		本章总结	67
与管理	27	第 6 章 大型网站架构	69
2.4 Memcached 实现主主复制和		6.1 网站架构概述	70
高可用的方式	30	6.2 百万 PV 网站架构案例	70
2.4.1 Memcached 主主复制架构	30	6.3 千万 PV 网站架构案例	81
2.4.2 Memcached 主主复制 +Keepalived		6.3.1 案例概述	81
高可用架构	32	6.3.2 RabbitMQ 集群配置	82
本章总结	35	本章总结	95

第 7 章 自动化运维之 Ansible.....	97	8.2.1 需求分析.....	139
7.1 Ansible 概述	98	8.2.2 操作步骤.....	139
7.2 Ansible 核心组件	99	本章总结.....	149
7.3 安装部署 Ansible 服务	100		
7.4 Ansible 命令应用基础	102	第 9 章 自动化运维之 Puppet.....	151
7.5 YAML 介绍.....	114	9.1 案例一分析	152
7.6 Ansible 基础元素介绍	115	9.2 案例一实施	154
7.7 Playbook 介绍	119	9.3 案例二分析	162
本章总结.....	136	9.4 案例二实施	163
		本章总结.....	173
第 8 章 自动化运维之 SaltStack ..	137	本章作业.....	174
8.1 SaltStack 基础	138		
8.2 SaltStack 批量部署并 配置 Nginx	139		

技能目标

- 学会构建传统代理、透明代理、反向代理服务
- 学会配置 Squid 的访问控制策略
- 熟悉 Varnish、Nginx 缓存服务器

本章导读

Squid 是 Linux 系统中最常用的一款开源代理服务软件（官方网站为 <http://www.squid-cache.org>），可以很好地实现 HTTP、FTP、DNS 查询，以及 SSL 等应用的缓存代理，功能十分强大。

知识服务



第 1 章

Squid 缓存服务器

技能目标

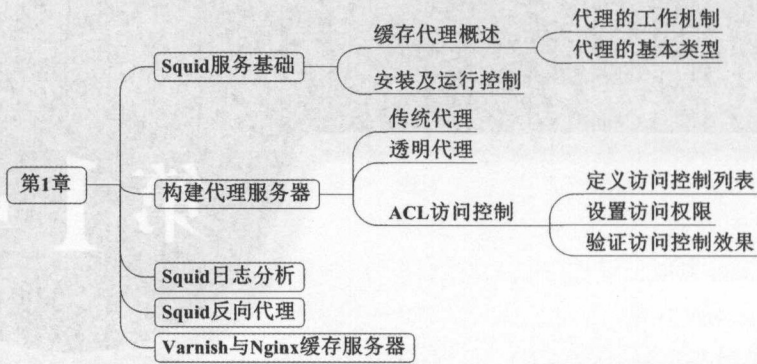
- 学会构建传统代理、透明代理、反向代理服务
- 学会配置 Squid 的访问控制策略
- 熟悉 Varnish、Nginx 缓存服务器

本章导读

Squid 是 Linux 系统中最常用的一款开源代理服务软件（官方网站为 <http://www.squid-cache.org>），可以很好地实现 HTTP、FTP、DNS 查询，以及 SSL 等应用的缓存代理，功能十分强大。

知识服务





1.1 Squid 服务基础

本节将介绍缓存代理的工作机制、类型，以及 Squid 服务的安装、运行控制和配置文件。

1.1.1 缓存代理概述

作为应用层的代理服务软件，Squid 主要提供缓存加速和应用层过滤控制的功能。

1. 代理的工作机制

当客户机通过代理来请求 Web 页面时，指定的代理服务器会先检查自己的缓存，如果缓存中已经有客户机需要访问的页面，则直接将缓存中的页面内容反馈给客户机；如果缓存中没有客户机需要访问的页面，则由代理服务器向 Internet 发送访问请求，当获得返回的 Web 页面以后，将网页数据保存到缓存中并发送给客户机，如图 1.1 所示。

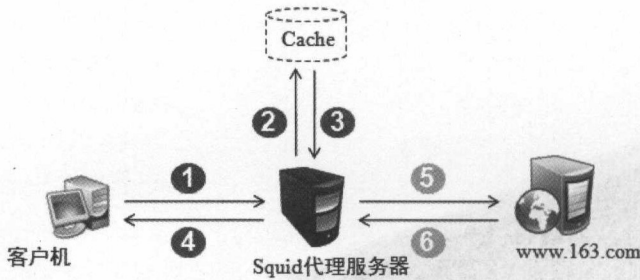


图 1.1 代理服务的缓存机制

HTTP 代理的缓存加速对象主要是文字、图像等静态 Web 元素。使用缓存机制后，当客户机在不同的时候访问同一 Web 元素，或者不同的客户机访问相同的 Web 元素时，可以直接从代理服务器的缓存中获得结果。这样就大大减少了向 Internet 重复提交 Web 请求的过程，提高了客户机的 Web 访问响应速度。

由于客户机的 Web 访问请求实际上是由代理服务器来代替完成的，从而可以隐藏

用户的真实 IP 地址,起到一定的保护作用。另一方面,代理服务器担任着类似“经纪人”的角色,所以有机会针对要访问的目标、客户机的地址、访问的时间段等进行过滤控制。

2. 代理的基本类型

根据实现的方式不同,代理服务可分为传统代理和透明代理两种常见的代理服务。

- 传统代理:也就是普通的代理服务,首先必须在客户机的浏览器、QQ 聊天工具、下载软件等程序中手动设置代理服务器的地址和端口,然后才能使用代理服务来访问网络。对于网页浏览器,访问网站时的域名解析请求也会发送给指定的代理服务器。
- 透明代理:提供与传统代理相同的功能和服务,其区别在于客户机不需要指定代理服务器的地址和端口,而是通过默认路由、防火墙策略将 Web 访问重定向,实际上仍然交给代理服务器来处理。重定向的过程对客户机来说是“透明”的,用户甚至并不知道自己在使用代理服务,所以称为“透明代理”。使用透明代理时,网页浏览器访问网站时的域名解析请求将优先发给 DNS 服务器。

在实际应用中,传统代理多见于 Internet 环境,如为 QQ 程序使用代理可以隐藏本机真实 IP 地址,为下载工具使用多个代理可以规避服务器的并发连接限制。而透明代理多见于局域网环境,如在 Linux 网关中启用透明代理后,局域网主机无需进行额外的设置就可以享受更好的上网速度。

1.1.2 安装及运行控制

下面以 Squid 3.4.6 版为例,介绍其安装和运行控制。

1. 编译安装 Squid

配置 Squid 的编译选项时,将安装目录设置为 /usr/local/squid,其他具体选项根据实际需要来确定,配置前可参考 ./configure --help 给出的说明。

```
[root@localhost ~]# tar zxf squid-3.4.6.tar.gz
[root@localhost ~]# cd squid-3.4.6
[root@localhost squid-3.4.6]# ./configure --prefix=/usr/local/squid
--sysconfdir=/etc --enable-arp-acl --enable-linux-netfilter --enable-linux-tproxy
--enable-async-io=100 --enable-err-language="Simplify_Chinese"
--enable-underscore --enable-poll --enable-gnuregex
[root@localhost squid-3.4.6]# make && make install
```

上述选项含义如下:

--prefix=/usr/local/squid	// 安装目录
--sysconfdir=/etc	// 单独将配置文件修改到其他目录
--enable-arp-acl	
// 可以在规则中设置为直接通过客户端 MAC 进行管理,防止客户端使用 IP 欺骗	
--enable-linux-netfilter	// 使用内核过滤
--enable-linux-tproxy	// 支持透明模式


```
--enable-async-io= 值
// 异步 I/O, 提升存储性能, 相当于 --enable-pthreads --enable-storeio=ufs,aufs
// --with -pthread --with-aufs-thread= 值
--enable-err-language="Simplify_Chinese" // 错误信息的显示语言
--enable-underscore // 允许 URL 中有下划线
--enable-poll // 使用 Poll() 模式, 提升性能
--enable-gnuregex // 使用 GNU 正则表达式
```

安装完后, 创建链接文件、创建用户和组。

```
[root@localhost ~]# ln -s /usr/local/squid/sbin/* /usr/local/sbin/
[root@localhost ~]# useradd -M -s /sbin/nologin squid
[root@localhost ~]# chown -R squid:squid /usr/local/squid/var/
```

2. Squid 的配置文件

Squid 服务的配置文件位于 `/etc/squid.conf`, 充分了解配置行的作用将有助于管理员根据实际情况灵活地配置代理服务。更详细的配置项请参考 `/etc/squid.conf` documented 文件。

```
acl localnet src 10.0.0/8 # RFC1918 possible internal network
acl localnet src 172.16.0.0/12 # RFC1918 possible internal network
acl localnet src 192.168.0.0/16 # RFC1918 possible internal network
acl localnet src fc00::/7 # RFC 4193 local private network range
acl localnet src fe80::/10 # RFC 4291 link-local (directly plugged) machines

acl SSL_ports port 443
acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 # https
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http

acl CONNECT method CONNECT

http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost manager
http_access deny manager
http_access allow localnet
http_access allow localhost
http_access deny all

http_port 3128 // 用来指定代理服务监听的地址和端口 (默认的端口号为 3128)
cache_effective_user squid // 这一项指定 squid 的程序用户, 用来设置初始化、运行时缓存
// 的账号, 否则启动不成功!
cache_effective_group squid // 默认为 cache_effective_user 指定账号的基本组
coredump_dir /usr/local/squid/var/cache/squid
```


refresh_pattern ^ftp:	1440	20%	10080
refresh_pattern ^gopher:	1440	0%	1440
refresh_pattern -i (/cgi-bin/ \?) 0	0%	0	
refresh_pattern .	0	20%	4320

3. Squid 的运行控制

(1) 检查配置文件的语法是否正确。

```
[root@localhost squid]# squid -k parse
```

(2) 启动、停止 Squid。

第一次启动 Squid 服务时，会自动初始化缓存目录。在没有可用的 Squid 服务脚本的情况下，也可以直接调用 Squid 程序来启动服务，这时需要先进行初始化。

```
[root@localhost ~]# squid -z           // -z 选项用来初始化缓存目录
[root@localhost ~]# squid              // 启动 squid 服务
```

确认 Squid 服务处于正常监听状态。

```
[root@localhost ~]# netstat -anpt | grep "squid"
tcp  0  0  0.0.0.0:3128  0.0.0.0:*  LISTEN  9947/(squid).
```

(3) 使用 Squid 服务脚本。

为了使 Squid 服务的启动、停止、重载等操作更加方便，可以编写 Squid 服务脚本，并使用 chkconfig 和 service 工具来进行管理。

```
#!/bin/bash
# chkconfig: 2345 90 25
# config: /etc/squid.conf
# pidfile: /usr/local/squid/var/run/squid.pid
# Description: Squid - Internet Object Cache.
PID="/usr/local/squid/var/run/squid.pid"
CONF="/etc/squid.conf"
CMD="/usr/local/squid/sbin/squid"
case "$1" in
    start)
        netstat -anpt | grep squid &> /dev/null
        if [ $? -eq 0 ]
        then
            echo "squid is running"
        else
            echo "正在启动 squid..."
            $CMD
        fi
    ;;
    stop)
        $CMD -k kill &> /dev/null
        rm -rf $PID &> /dev/null
    ;;
    *)

```