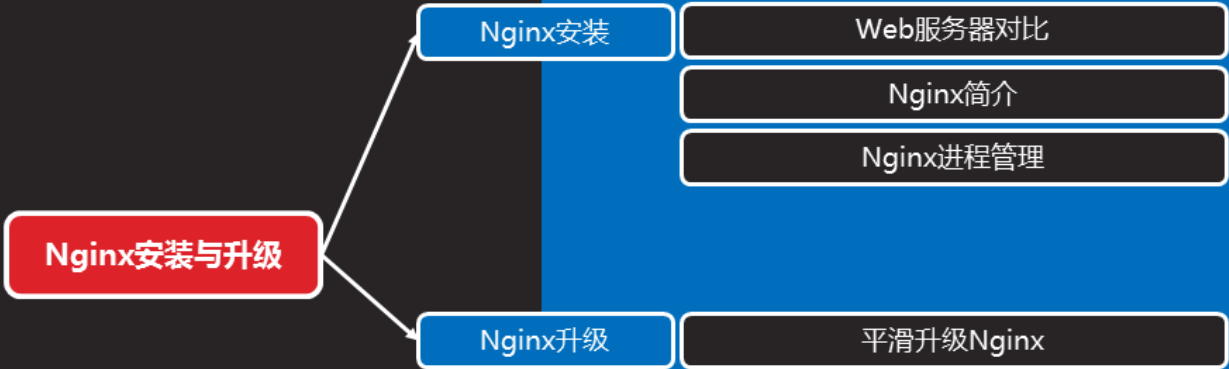


# 内容

上午	09:00 ~ 09:30	Nginx安装与升级
	09:30 ~ 10:20	
	10:30 ~ 11:20	Nginx服务器
	11:30 ~ 12:20	Nginx虚拟主机
下午	14:00 ~ 14:50	
	15:00 ~ 15:50	HTTPS加密网站
	16:00 ~ 16:50	
	17:00 ~ 17:30	总结和答疑



## Nginx安装与升级



# Nginx安装



## Web服务器对比

- Unix和Linux平台下
  - Apache、Nginx、Tengine、Lighttpd
  - Tomcat、IBM WebSphere、Jboss
- Windows平台下
  - 微软公司的IIS(Internet Information Server)

# Nginx简介

知识讲解

- Nginx ( "engine x" )
  - 是俄罗斯人编写的十分轻量级的HTTP服务器
  - 是一个高性能的HTTP和反向代理服务器，同时也是一个IMAP/POP3/SMTP 代理服务器
  - 官方网站：http://nginx.org/

NGINX



## Nginx简介（续1）

知识讲解

- Nginx软件安装

```
[root@localhost ~]# yum -y install gcc pcre-devel openssl-devel
[root@localhost ~]# useradd nginx
[root@localhost ~]# tar -zxvf nginx-1.10.3.tar.gz
[root@localhost ~]# cd nginx-1.10.3
[root@localhost nginx-1.10.3]# ./configure \
> --prefix=/usr/local/nginx \           #指定安装目录
> --user=nginx \                         #指定账户名称
> --group=nginx \                       #指定组名称
> --with-http_ssl_module                 #支持加密功能
[root@localhost nginx-1.10.3]# make && make install
```



## Nginx简介（续2）

知识讲解

- Nginx配置文件及目录
  - /usr/local/nginx/ //安装目录
  - conf/nginx.conf //主配置文件
  - html //网页目录
  - logs //日志文件
  - sbin/nginx //启动脚本



## Nginx进程管理

知识讲解

- 启动Nginx服务  
`[root@localhost nginx]# /usr/local/nginx/sbin/nginx`
- 常用选项
  - -V：查看编译参数
  - -c：指定配置文件，启动服务



## Nginx进程管理（续1）

- 查看服务相关进程及端口信息

```
[root@localhost nginx]# ps aux | grep nginx
```

```
root    20821  0.0  0.0  41136  848 ?        Ss   11:36   0:00 nginx: master process
```

```
nginx   20822  0.0  0.1  41508 1400 ?        S    11:36   0:00 nginx: worker process
```

```
[root@localhost nginx]# netstat -utnlp | grep nginx
```

```
tcp     0      0 0.0.0.0:80 0.0.0.0:*    LISTEN  20821/nginx
```

知识讲解



## Nginx升级

# 平滑升级Nginx

- 在不停止服务的情况下升级软件版本

```
[root@localhost ~]# tar -zxvf nginx-1.12.2.tar.gz
[root@localhost ~]# cd nginx-1.12.2
[root@localhost nginx-1.12.2]# ./configure \
> --prefix=/usr/local/nginx \
> --user=nginx \
> --group=nginx \
> --with-http_ssl_module
[root@localhost nginx-1.12.2]# make
```

知识讲解



## 平滑升级Nginx ( 续1 )

```
[root@localhost ~]# cd /usr/local/nginx/sbin/
[root@localhost sbin]# mv nginx nginxold           //改名旧脚本

[root@localhost ~]# cd /root/lnmp_soft/nginx-1.12.2/objs/
[root@localhost objs]# cp nginx /usr/local/nginx/sbin/ //拷贝新脚本
```

知识讲解



## 平滑升级Nginx ( 续2 )

知识讲解

```
[root@localhost nginx-1.12.2]# make upgrade           //升级软件
/usr/local/nginx/sbin/nginx -t
nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok
nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful
kill -USR2 `cat /usr/local/nginx/logs/nginx.pid`
sleep 1
test -f /usr/local/nginx/logs/nginx.pid.oldbin
kill -QUIT `cat /usr/local/nginx/logs/nginx.pid.oldbin`
```

```
[root@localhost sbin]# ./nginx -v                     //查看软件版本
nginx: nginx version: nginx/1.12.2
```



## 案例1：搭建Nginx服务器

在IP地址为192.168.4.5的主机上部署Nginx服务

- 1) 安装部署Nginx软件，支持HTTPS加密
- 2) 客户端访问Nginx服务器
- 3) 升级Nginx服务器

课堂练习



# Nginx服务器

Nginx服务器

Nginx配置解析

配置文件结构

全局选项

配置容器

用户认证

Tedu.cn  
达内教育

# Nginx配置解析



# 配置文件结构

## 全局配置

```
http{  
    ... ..  
    server {  
        ... ..  
        location / {  
            ... ..  
        }  
    }  
}
```

知识讲解



# 全局选项

```
user  nginx;           //进程所有者  
worker_processes  1;    //启动进程数量  
error_log  /var/log/nginx/error.log;  //日志文件  
pid        /var/run/nginx.pid;         //PID文件  
events {  
    worker_connections  1024;          //单个进程最大并发量  
}
```

知识讲解



## 配置容器

知识讲解

```
http {  
    .....  
    server {                                //定义虚拟主机  
        listen    80;  
        server_name localhost;  
        location / {                        //发布目录  
            root    html;  
            index   index.html index.htm;  
        }  
    }  
}
```



## 用户认证

知识讲解

```
location / {  
    root html;  
    index index.html;  
    auth_basic "auth-domain";  
    auth_basic_user_file /usr/local/nginx/pass;  
}
```

```
[root@localhost ~]# yum -y install httpd-tools
```

```
[root@localhost ~]# htpasswd -c /usr/local/nginx/pass admin  
//创建认证用户
```



## 案例2：用户认证

课堂练习

- 部署Nginx Web服务器
  - 所有人访问页面均需要身份认证
  - 用户名为tom，密码为123456



### Nginx虚拟主机

Nginx虚拟主机

三种模式虚拟主机

基于域名的虚拟主机

基于端口的虚拟主机

基于IP的虚拟主机

# 三种模式虚拟主机

## 基于域名的虚拟主机

```
server {  
    listen    80;  
    server_name web1.plj.com;    //域名1  
    ...  
}  
server {  
    listen    80;  
    server_name web2.plj.com;    //域名2  
    ...  
}  
}
```



# 基于端口的虚拟主机

知识讲解

```
server {  
    listen    8080;           //端口  
    server_name web1.plj.com; //域名  
    .....  
}  
server {  
    listen    8000;  
    server_name web1.plj.com;  
    .....  
}
```



# 基于IP的虚拟主机

知识讲解

```
server {  
    listen    192.168.0.1:80; //端口  
    server_name web1.plj.com; //域名  
    .....  
}  
server {  
    listen    192.168.0.2:80;  
    server_name web1.plj.com;  
    .....  
}
```



## 案例3：基于域名的虚拟主机

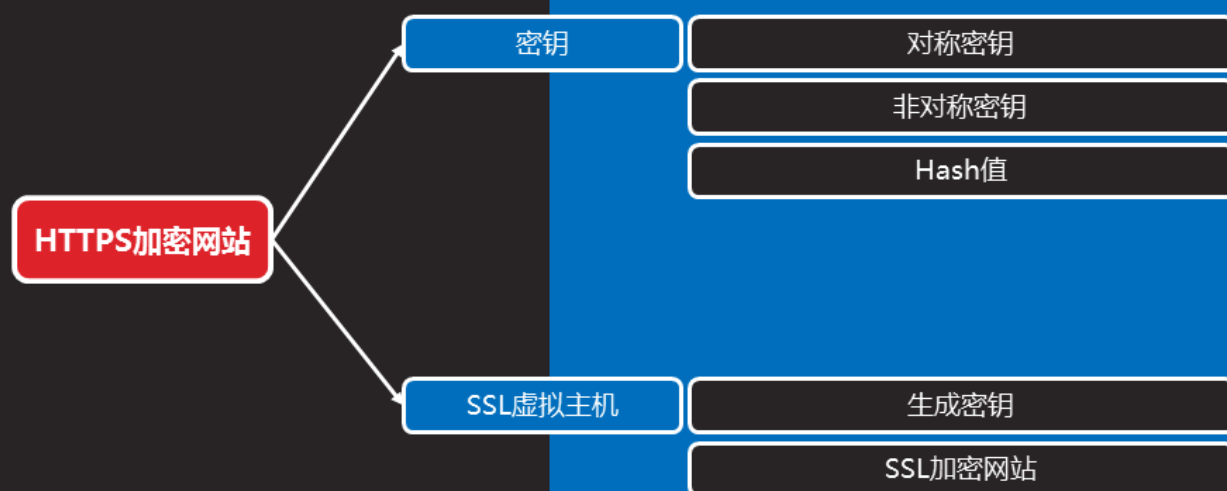
配置基于域名的虚拟主机，要求如下：

- 域名为www.aa.com、www.bb.com
- 对域名为www.aa.com的站点进行用户认证，用户名称为tom，密码为123456

课堂练习



### HTTPS加密网站



# 密钥



## 对称密钥

- AES
- DES
- 应用案例：RAR、ZIP压缩加密（单机加密）

知识讲解



# 非对称密钥

知识讲解

- RSA
- DSA
- 应用案例：网络加密（https、ssh）



# Hash值

知识讲解

- MD5
- SHA256
- SHA512
- 应用案例：数据完整性校验、文件秒传





# SSL虚拟主机

## 生成密钥

- SSL加密网站的核心技术是非对称生成密钥
  - 公钥、私钥、证书

知识  
讲解

```
# cd /usr/local/nginx/conf/
# openssl genrsa > cert.key 2048 //生成私钥
# openssl req -new -x509 -key cert.key > cert.pem //生成证书
```



# SSL加密网站

知识讲解

```
server {                                     //定义虚拟主机
    listen 443 ssl;                         //开启SSL
    server_name www.test.com;
    ssl_certificate cert.pem;               //指定证书文件
    ssl_certificate_key cert.key;           //指定私钥文件
    ssl_session_timeout 5m;
    ssl_protocols SSLv2 SSLv3 TLSv1;
    ssl_ciphers HIGH:!aNULL:!MD5;
    ssl_prefer_server_ciphers on;
    location / {
        root html;
        index index.html index.htm;
    }
}
```



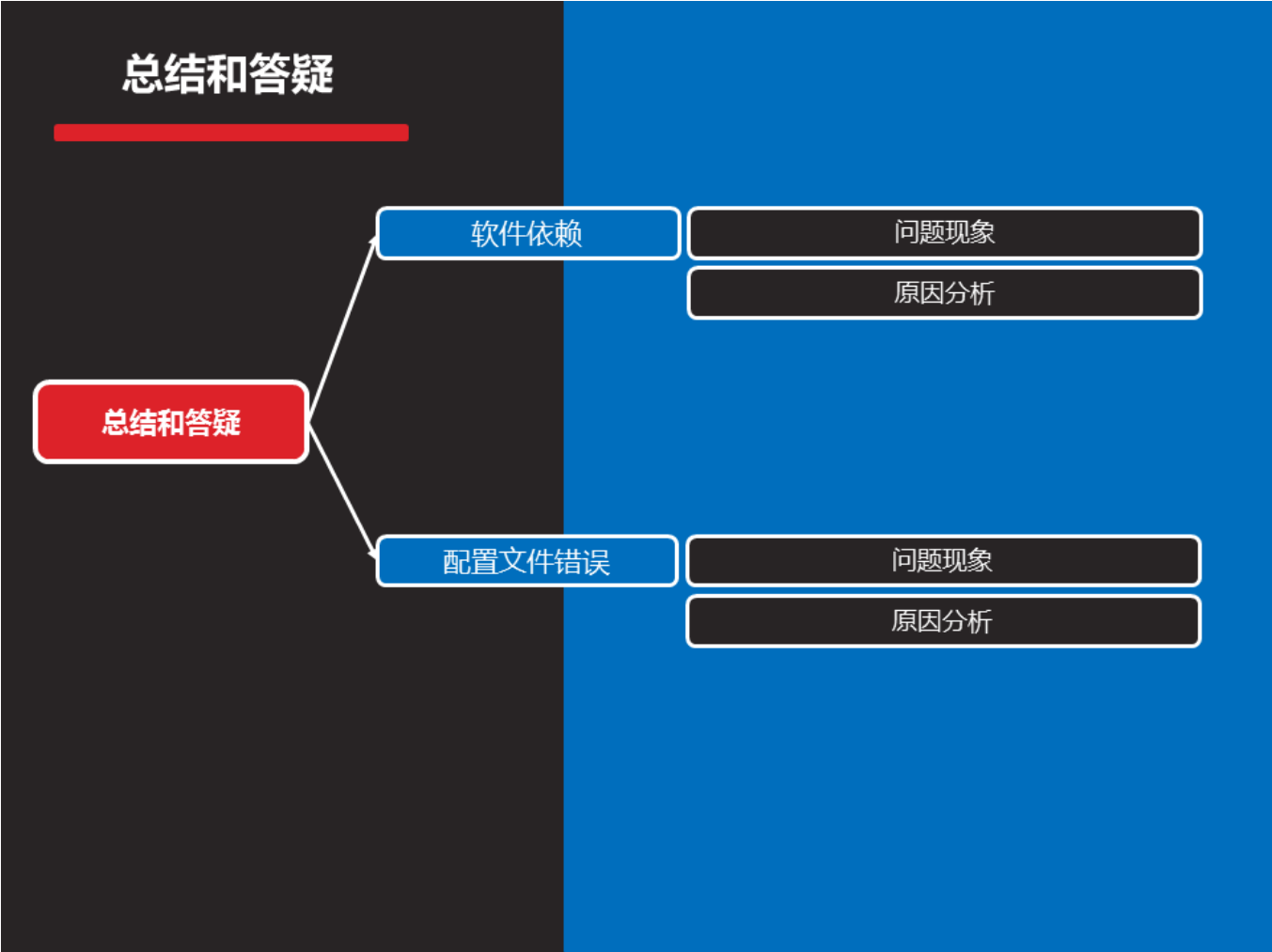
## 案例4：SSL虚拟主机


配置SSL虚拟主机，要求如下：

- 域名为www.cc.com
- 该站点通过https访问
- 通过私钥、证书对该站点所有数据加密

课堂练习







达内教育

# 软件依赖

---

## 问题现象

- 故障错误信息

```
[root@svr5 nginx-1.10.3]# ./configure
./configure: error: the HTTP rewrite module requires the PCRE library.
You can either disable the module by using --without-http_rewrite_module
option, or install the PCRE library into the system, or build the PCRE library
statically from the source with nginx by using --with-pcre=<path> option.
```

知识讲解



## 原因分析

- 分析故障
- 报错信息：缺少PCRE库
- 分析故障原因
  - 安装PCRE库 ( pcre-devel)
  - 禁用nginx地址重写功能 ( --without-http\_rewrite\_module )

知识讲解



# 配置文件错误

## 问题现象

- 故障错误信息

```
[root@svr5 ~]# nginx -t
directive "worker_processes" is not terminated by ";"
"server" directive is not allowed here in
unexpected end of file, expecting "}"
the "ssl" parameter requires ngx_http_ssl_module
unknown directive "server-name"
```

知识讲解



# 原因分析

知识讲解

- 分析故障信息
  - directive "worker\_processes" is not terminated by ";"
  - "server" directive is not allowed here in
  - unexpected end of file, expecting "}"
  - the "ssl" parameter requires ngx\_http\_ssl\_module
  - unknown directive "server-name"
- 分析故障原因
  - 缺少 ; 符号
  - 关键词server的位置不对
  - {}不匹配
  - 实现加密网站，但编译nginx时没有指定ssl模块
  - 未知的指令(server-name)

