网管服务器部署

部署环境:

系统：CentOS 7.9.2009

部署流程

1、将网管软件包上传到服务器/home 路径下，如图 1 所示。

图 1

2、使用 tar -xzvf itms_tw.tar.gz 命令解压缩。

3、(1) 如有环境有外网连接，则使用 yum -y install --downloadonly --downloaddir=/home/itms_tw/nginx-1.20.1/offlineLibs gcc zlib zlib-devel pcre-devel openssl openssl-devel 命令来下载软件包。如图 2 所示。

```
[root@localhost home]# yum -y install  --downloadonly --downloaddir=/home/itms_tw/nginx-1.20.1/offlineLibs gcc zlib zlib-devel
pcre-devel openssl openssl-devel
已加载插件：fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.aliyun.com
 * extras: mirrors.aliyun.com
 * updates: mirrors.aliyun.com
软件包 gcc-4.8.5-44.el7.x86_64 已安装并且是最新版本
正在解决依赖关系
--> 正在检查事务
---> 软件包 openssl.x86_64.1.1.0.2k-19.el7 将被 升级
---> 软件包 openssl.x86_64.1.1.0.2k-26.el7_9 将被 更新
--> 正在处理依赖关系 openssl-libs(x86-64) = 1:1.0.2k-26.el7_9，它被软件包 1:openssl-1.0.2k-26.el7_9.x86_64 需要
---> 软件包 openssl-devel.x86_64.1.1.0.2k-26.el7_9 将被 安装
--> 正在处理依赖关系 krb5-devel(x86-64)，它被软件包 1:openssl-devel-1.0.2k-26.el7_9.x86_64 需要
---> 软件包 pcre-devel.x86_64.0.8.32-17.el7 将被 安装
---> 软件包 zlib.x86_64.0.1.2.7-18.el7 将被 升级
---> 软件包 zlib.x86_64.0.1.2.7-21.el7_9 将被 更新
---> 软件包 zlib-devel.x86_64.0.1.2.7-21.el7_9 将被 安装
--> 正在检查事务
---> 软件包 krb5-devel.x86_64.0.1.15.1-55.el7_9 将被 安装
--> 正在处理依赖关系 libkadm5(x86-64) = 1.15.1-55.el7_9，它被软件包 krb5-devel-1.15.1-55.el7_9.x86_64 需要
--> 正在处理依赖关系 krb5-libs(x86-64) = 1.15.1-55.el7_9，它被软件包 krb5-devel-1.15.1-55.el7_9.x86_64 需要
--> 正在处理依赖关系 libverto-devel，它被软件包 krb5-devel-1.15.1-55.el7_9.x86_64 需要
--> 正在处理依赖关系 libselinux-devel，它被软件包 krb5-devel-1.15.1-55.el7_9.x86_64 需要
--> 正在处理依赖关系 libcom_err-devel，它被软件包 krb5-devel-1.15.1-55.el7_9.x86_64 需要
--> 正在处理依赖关系 keyutils-libs-devel，它被软件包 krb5-devel-1.15.1-55.el7_9.x86_64 需要
---> 软件包 openssl-libs.x86_64.1.1.0.2k-19.el7 将被 升级
---> 软件包 openssl-libs.x86_64.1.1.0.2k-26.el7_9 将被 更新
--> 正在检查事务
---> 软件包 keyutils-libs-devel.x86_64.0.1.5.8-3.el7 将被 安装
---> 软件包 krb5-libs.x86_64.0.1.15.1-50.el7 将被 升级
--> 正在处理依赖关系 krb5-libs(x86-64) = 1.15.1-50.el7，它被软件包 krb5-server-1.15.1-50.el7.x86_64 需要
--> 正在处理依赖关系 krb5-libs(x86-64) = 1.15.1-50.el7，它被软件包 krb5-workstation-1.15.1-50.el7.x86_64 需要
--> 正在处理依赖关系 krb5-libs(x86-64) = 1.15.1-50.el7，它被软件包 krb5-pkinit-1.15.1-50.el7.x86_64 需要
---> 软件包 krb5-libs.x86_64.0.1.15.1-55.el7_9 将被 更新
---> 软件包 libcom_err-devel.x86_64.0.1.42.9-19.el7 将被 安装
---> 软件包 libkadm5.x86_64.0.1.15.1-50.el7 将被 升级
---> 软件包 libkadm5.x86_64.0.1.15.1-55.el7_9 将被 更新
---> 软件包 libselinux-devel.x86_64.0.2.5-15.el7 将被 安装
--> 正在处理依赖关系 libsepol-devel(x86-64) >= 2.5-10，它被软件包 libselinux-devel-2.5-15.el7.x86_64 需要
```

图 2

（2）如果环境无外网连接，则可以进入到/home/itms_tw/nginx-1.20.1/offlineLibs/路径下，并使用命令：rpm -Uvh --force --nodeps *.rpm 来安装所有后缀为 rpm 的软件，如图 3 所示。

```
[root@localhost offlineLibs]# rpm -Uvh --force --nodeps *.rpm
准备中...                          ################################# [100%]
正在升级/安装...
   1:libgcc-4.8.5-44.el7            ################################# [  3%]
   2:glibc-common-2.17-325.el7_9    ################################# [  5%]
   3:glibc-2.17-325.el7_9           ################################# [  8%]
   4:libcom_err-1.42.9-19.el7       ################################# [ 11%]
   5:zlib-1.2.7-19.el7_9            ################################# [ 14%]
   6:krb5-libs-1.15.1-50.el7        ################################# [ 16%]
   7:openssl-libs-1:1.0.2k-22.el7_9 ################################# [ 19%]
   8:mpfr-3.1.1-4.el7               ################################# [ 22%]
   9:libmpc-1.0.1-3.el7             ################################# [ 24%]
  10:cpp-4.8.5-44.el7               ################################# [ 27%]
  11:libkadm5-1.15.1-50.el7         ################################# [ 30%]
  12:zlib-devel-1.2.7-19.el7_9      ################################# [ 32%]
  13:e2fsprogs-libs-1.42.9-19.el7   ################################# [ 35%]
  14:libcom_err-devel-1.42.9-19.el7 ################################# [ 38%]
  15:libss-1.42.9-19.el7            ################################# [ 41%]
  16:libgomp-4.8.5-44.el7           ################################# [ 43%]
  17:pcre-devel-8.32-17.el7         ################################# [ 46%]
  18:libverto-devel-0.2.5-4.el7     ################################# [ 49%]
  19:libsepol-devel-2.5-10.el7      ################################# [ 51%]
  20:libselinux-devel-2.5-15.el7    ################################# [ 54%]
  21:keyutils-libs-devel-1.5.8-3.el7 ################################ [ 57%]
  22:krb5-devel-1.15.1-50.el7       ################################# [ 59%]
  23:kernel-headers-3.10.0-1160.45.1.e################################ [ 62%]
  24:glibc-headers-2.17-325.el7_9   ################################# [ 65%]
  25:glibc-devel-2.17-325.el7_9     ################################# [ 68%]
  26:gcc-4.8.5-44.el7               ################################# [ 70%]
  27:openssl-devel-1:1.0.2k-22.el7_9 ################################ [ 73%]
  28:e2fsprogs-1.42.9-19.el7        ################################# [ 76%]
  29:openssl-1:1.0.2k-22.el7_9      ################################# [ 78%]
正在清理/删除...
  30:glibc-devel-2.17-317.el7       ################################# [ 81%]
  31:openssl-1:1.0.2k-19.el7        ################################# [ 84%]
  32:glibc-headers-2.17-317.el7     ################################# [ 86%]
  33:openssl-libs-1:1.0.2k-19.el7   ################################# [ 89%]
```


图 3

4、进入/home/itms_tw/nginx-1.20.1/目录下，手动开启 stream 模块，执行命令：./configure --with-stream，如图 4 所示。

```
[root@localhost nginx-1.20.1]# ./configure --with-stream
checking for OS
 + Linux 3.10.0-1160.el7.x86_64 x86_64
checking for C compiler ... found
 + using GNU C compiler
 + gcc version: 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC)
checking for gcc -pipe switch ... found
checking for -Wl,-E switch ... found
checking for gcc builtin atomic operations ... found
checking for C99 variadic macros ... found
checking for gcc variadic macros ... found
checking for gcc builtin 64 bit byteswap ... found
checking for unistd.h ... found
checking for inttypes.h ... found
checking for limits.h ... found
checking for sys/filio.h ... not found
checking for sys/param.h ... found
checking for sys/mount.h ... found
checking for sys/statvfs.h ... found
checking for crypt.h ... found
checking for Linux specific features
checking for epoll ... found
checking for EPOLLRDHUP ... found
checking for EPOLLEXCLUSIVE ... not found
checking for eventfd() ... found
checking for O_PATH ... found
```

图 4

5、执行命令 make && make install 编译并安装，如图 5 所示。

```
[root@localhost nginx-1.20.1]# make && make install
make -f objs/Makefile
make[1]: 进入目录 "/home/itms_tw/nginx-1.20.1"
cc -c -pipe  -O -W -Wall -Wpointer-arith -Wno-unused-parameter -Werror -g  -I src/core -I src/event -I src/event/modules -I sr
c/os/unix -I objs \
        -o objs/src/core/nginx.o \
        src/core/nginx.c
cc -c -pipe  -O -W -Wall -Wpointer-arith -Wno-unused-parameter -Werror -g  -I src/core -I src/event -I src/event/modules -I sr
c/os/unix -I objs \
        -o objs/src/core/ngx_log.o \
        src/core/ngx_log.c
cc -c -pipe  -O -W -Wall -Wpointer-arith -Wno-unused-parameter -Werror -g  -I src/core -I src/event -I src/event/modules -I sr
c/os/unix -I objs \
        -o objs/src/core/ngx_palloc.o \
        src/core/ngx_palloc.c
cc -c -pipe  -O -W -Wall -Wpointer-arith -Wno-unused-parameter -Werror -g  -I src/core -I src/event -I src/event/modules -I sr
c/os/unix -I objs \
        -o objs/src/core/ngx_array.o \
        src/core/ngx_array.c
cc -c -pipe  -O -W -Wall -Wpointer-arith -Wno-unused-parameter -Werror -g  -I src/core -I src/event -I src/event/modules -I sr
c/os/unix -I objs \
        -o objs/src/core/ngx_list.o \
        src/core/ngx_list.c
cc -c -pipe  -O -W -Wall -Wpointer-arith -Wno-unused-parameter -Werror -g  -I src/core -I src/event -I src/event/modules -I sr
c/os/unix -I objs \
        -o objs/src/core/ngx_hash.o \
        src/core/ngx_hash.c
cc -c -pipe  -O -W -Wall -Wpointer-arith -Wno-unused-parameter -Werror -g  -I src/core -I src/event -I src/event/modules -I sr
c/os/unix -I objs \
        -o objs/src/core/ngx_buf.o \
        src/core/ngx_buf.c
cc -c -pipe  -O -W -Wall -Wpointer-arith -Wno-unused-parameter -Werror -g  -I src/core -I src/event -I src/event/modules -I sr
c/os/unix -I objs \
        -o objs/src/core/ngx_queue.o \
        src/core/ngx_queue.c
cc -c -pipe  -O -W -Wall -Wpointer-arith -Wno-unused-parameter -Werror -g  -I src/core -I src/event -I src/event/modules -I sr
c/os/unix -I objs \
        -o objs/src/core/ngx_output_chain.o \
        src/core/ngx_output_chain.c
cc -c -pipe  -O -W -Wall -Wpointer-arith -Wno-unused-parameter -Werror -g  -I src/core -I src/event -I src/event/modules -I sr
```

图 5

6、使用 cp 命令复制一份 Nginx 的配置文件以备用（也可不复制，但难以保证修改配置文件后 Nginx 服务一定能正常运行），如图 6 所示。

```
[root@localhost nginx-1.20.1]# cp /usr/local/nginx/conf/nginx.conf /usr/local/nginx/conf/nginx.bak
[root@localhost nginx-1.20.1]# cp -rf /home/itms_tw/nginx-1.20.1/conf/nginx.conf /usr/local/nginx/conf/nginx.conf
cp：是否覆盖"/usr/local/nginx/conf/nginx.conf"？ y
[root@localhost nginx-1.20.1]#
[root@localhost nginx-1.20.1]#
```

先备份nginx.conf

图 6

7、（1）执行命令：/usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf 以启动 Nginx 服务，使用命令 lsof -i 可以看到 Nginx 服务已启动并正在监听端口，如图 7 所

示。

图 7

（2）如果出现了如图 8 所示的报错，则需要进入到/usr/local/nginx/conf/nginx.conf 配置文件中，删除第二个 server 的配置，如图 9 所示。

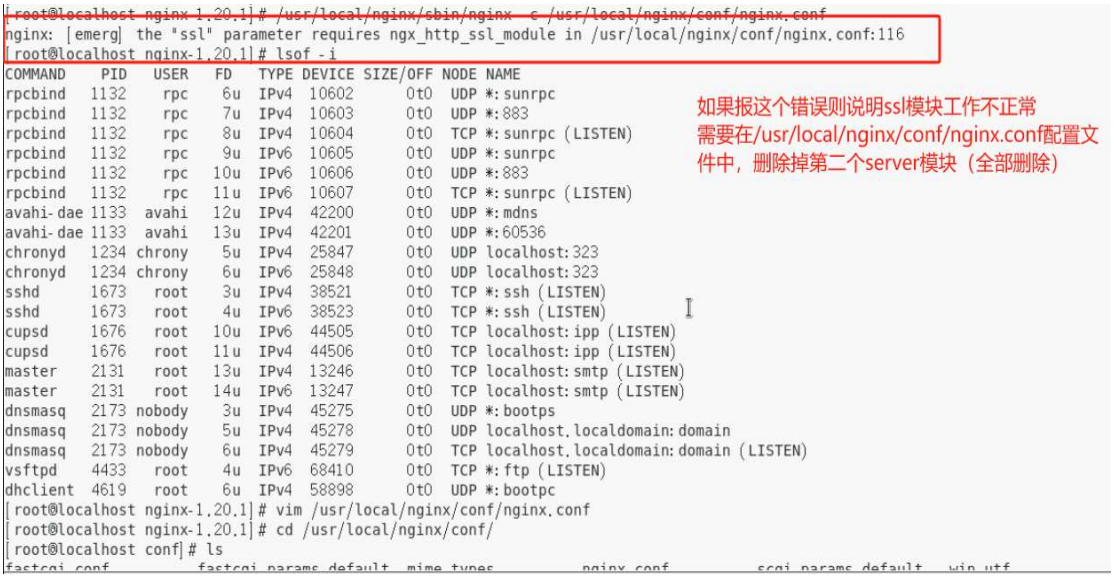

图 8

图 9

8、测试 Nginx 成功启动后须要使用命令/usr/local/nginx/sbin/nginx -s stop 命令停止 Nginx 运行，最后使用命令 bash /home/itms_tw/scripts/start.sh 执行脚本，开启网管服务，如图 10 所示。

图 10

9、访问地址 http://服务器 IP:8080/itms/index.login.action 进入到登录界面（一定要带端口号）如图 11 所示。

如果无法访问页面，使用命令：

firewall-cmd --zone=public --add-port=8080/tcp --permanent　开放 8080 端口

firewall-cmd --reload　重启防火墙服务

用户名：admin

密码：123

初次使用需导入授权码，先点“导入授权码”，复制弹出窗口上的机器码，并把机器码发给网管对接人员，把答复的授权码提交即可。