

作業 1：解釋源碼編譯安裝, nginx 軟件的預編譯, 編譯以及安裝, 分別在做什麼？
在 Linux 系統上通過源碼編譯安裝, 主要有三個步驟

1. configure
2. make
3. make install

configure 主要是檢測當前 Linux 作業環境有無符合軟件安裝需求, 會檢測當前有無 cc、gcc 軟件, 是一個 shell 腳本, 在 configure 配置軟件安裝參數後(預編譯), 會在當前源碼目錄下產生 Makefile 檔案跟 objs 目錄, 再以 make 命令進行編譯, 成為有權限的執行檔,

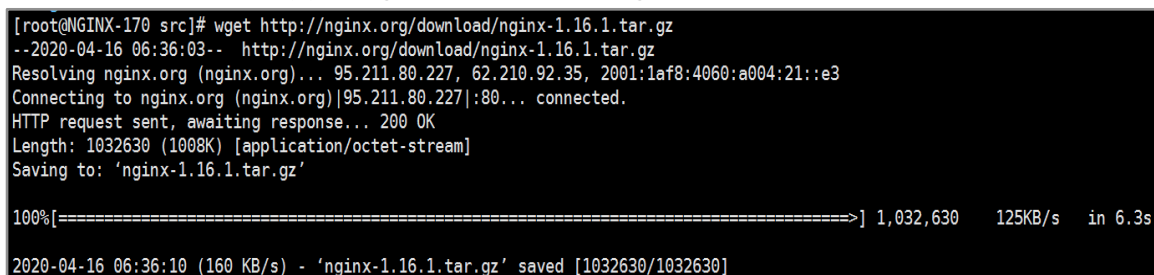
再用 make install 進行安裝, 會依據 configure 寫入的參數, 透過 make 編譯完成的路徑, 將相關檔案放置正確的路徑

作業 2：通過源碼方式編譯安裝 nginx, 並且順利發布 nginx 的默認頁面, 通過瀏覽器看到的響應頭信息中 nginx 的版本為, server: JFWS/1.1

2.1、至 nginx.org 官方網站下載 nginx 軟件源碼, 選擇 Stable version nginx-1.16.1 版本, 如圖一(a), 使用 wget 命令下載到/usr/src 目錄下, 如圖一(b)



圖一(a)、nginx 官方網站 nginx-1.16.1 穩定版本



圖一(b)、wget 命令下載 nginx-1.16.1 版本

2.2、使用 tar zxvf 解壓縮, 如圖二

```
[root@NGINX-170 src]# tar zxvf nginx-1.16.1.tar.gz
nginx-1.16.1/
nginx-1.16.1/auto/
nginx-1.16.1/conf/
nginx-1.16.1/contrib/
nginx-1.16.1/src/
```

圖二、解壓縮 nginx-1.16.1.tar.gz

2.3、修改/usr/src/nginx-1.16.1/src/core/nginx.h 內, #define NGINX_VERSION 為 1.1, #define NGINX_VERSION 為 JFWS/, 如圖三, 可在用戶瀏覽網頁時的響應頭 server 字段更改為 server: JFWS/1.1

```
/*
 * old configuration
#define NGINX_VERSION      "1.16.1"
#define NGINX_VER          "nginx/" NGINX_VERSION
*/
#define NGINX_VERSION      "1.1"
#define NGINX_VER          "JFWS/" NGINX_VERSION
```

圖三、修改響應頭 server 字段

2.4、至/usr/src/nginx-1.16.1 目錄下執行 configure 腳本(進行預編譯)並帶上--prefix=/usr/local/nginx 參數, 指定 nginx 軟件安裝目錄, 如圖四

```
[root@NGINX-170 nginx-1.16.1]# ./configure --prefix=/usr/local/nginx
checking for OS
+ Linux 3.10.0-957.el7.x86_64 x86_64
checking for C compiler ... found
+ using GNU C compiler
+ gcc version: 4.8.5 20150623 (Red Hat 4.8.5-39) (GCC)
checking for gcc -pipe switch ... found
```

圖四、進行 configure 預編譯

2.5、在/usr/src/nginx-1.16.1 目錄下檢查增加了 Makefile 和 objs 目錄, 如圖五

```
[root@NGINX-170 nginx-1.16.1]# ls
auto  CHANGES  CHANGES.ru  conf  configure  contrib  html  LICENSE  Makefile  man  objs  pre-config.txt  README  src
```

圖五、Makefile 和 objs 目錄

2.6、進行編譯及安裝, 在/usr/src/nginx-1.16.1 下使用命令 make && make install 進行編譯及安裝, 如圖六

```
[root@NGINX-170 nginx-1.16.1]# make && make install
make -f objs/Makefile
make[1]: Entering directory `/usr/src/nginx-1.16.1'
cc -c -pipe -O -W -Wall -Wpointer-arith -Wno-unused-parameter -Werror -g -I src/core -I src/event -I src/event/modules -I src/os/unix -I objs \
-o objs/src/core/nginx.o \
src/core/nginx.c
```

圖六、make && make install 進行編譯及安裝

2.7、安裝完成後, 在/usr/local/下已產生 nginx 相關檔案, 如圖七

```
[root@NGINX-170 nginx-1.16.1]# ls /usr/local/  
bin conf etc games html include lib lib64 libexec logs nginx share src
```

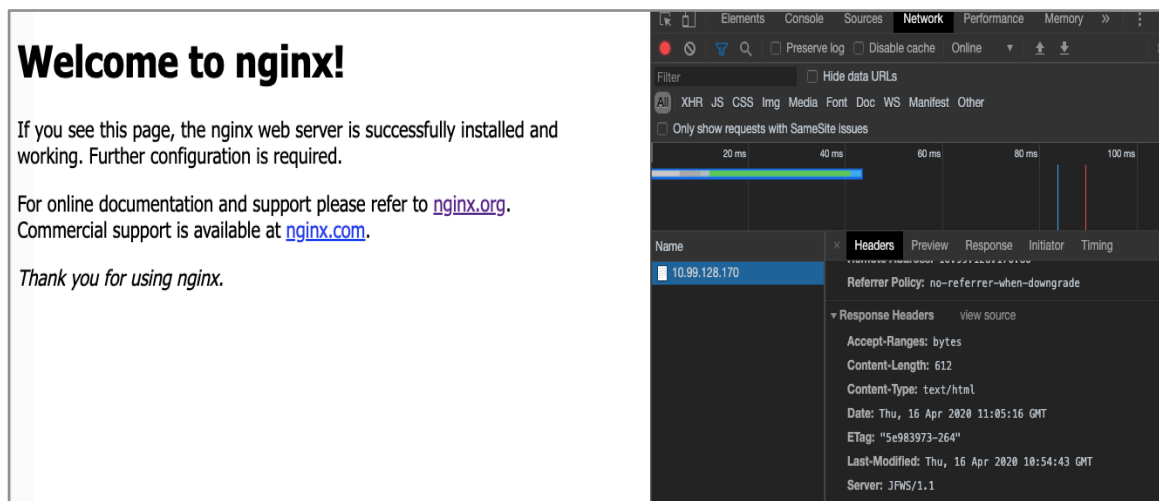
圖七、nginx 成功安裝

2.8、啟動 nginx 服務, 執行/usr/local/nginx/sbin/nginx 這個檔案, 使用 ps -ef | grep nginx, 可以查看程序有無正常啟動, 如圖八

```
[root@NGINX-170 sbin]# /usr/local/nginx/sbin/nginx  
[root@NGINX-170 sbin]# ps -ef | grep nginx  
root      14676      1    0 06:59 ?        00:00:00 nginx: master process /usr/local/nginx/sbin/nginx  
nobody    14677 14676    0 06:59 ?        00:00:00 nginx: worker process  
root      14679 15709    0 07:00 pts/1    00:00:00 grep --color=auto nginx
```

圖八、啟動 nginx 服務並查看

2.9、客戶使用瀏覽器訪問, 並查看 http response 標頭 server 字段有無顯示 JFWS/1.1, 如圖九



圖九、server 字段為 JFWS1.1

2.10、開機自動啟動 nginx, 需要在/usr/bin/system/system/目錄下增加 nginx.service 執行腳本, 如圖十(a), 並配置開機自啟動, 如圖十(b)

```
[Unit]  
Description=nginx daemon  
After=network.target  
  
[Service]  
Type=forking  
ExecStart=/usr/local/nginx/sbin/nginx  
ExecRestart=/usr/local/nginx/sbin/nginx -s reload  
ExecStop=/usr/local/nginx/sbin/nginx -s quit  
privateTmp=true  
  
[Install]  
WantedBy=multi-user.target
```

圖十(a)、新增 nginx.service 自啟動檔案

```
[root@NGINX-170 ~]# systemctl enable nginx
Created symlink from /etc/systemd/system/multi-user.target.wants/nginx.service to /usr/lib/systemd/system/nginx.service.
```

圖十(b)、開機自啟動

2.11、替/usr/local/nginx/sbin/nginx 配置別名, 易於管理上使用, 如圖十一(a)、圖十一(b)、圖十一(c)

```
[root@NGINX-170 ~]# pwd
/root
[root@NGINX-170 ~]# vim .bashrc
```

圖十一(a)、配置別名檔案

```
# .bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
alias nginx='/usr/local/nginx/sbin/nginx'
```

圖十一(b)、新增 nginx 別名

```
[root@NGINX-170 ~]# nginx -s reload
[root@NGINX-170 ~]# ps -ef | grep nginx
root      14676      1  0 06:59 ?        00:00:00 nginx: master process /usr/local/nginx/sbin/nginx
nobody    14699 14676   0 07:10 ?        00:00:00 nginx: worker process
root      14701 15709   0 07:10 pts/1    00:00:00 grep --color=auto nginx
```

圖十一(c)、使用別名啟動 nginx

2.12、替/usr/local/nginx/sbin/nginx 配置環境變量, 編寫一個腳本放置在 /etc/profile.d/目錄下, 如圖十二

```
[root@NGINX-170 profile.d]# echo "export PATH=/usr/local/nginx/sbin:${PATH}" >> /etc/profile.d/nginx.sh
```

圖十二、環境變量配置