# COMPILATION

## What is compilation. ?

Compilation is the process of building a program from it's source code. To compile a source code, we need a compiler.Compiler will convert the source code into a program. Usually we use a compiler called gcc to compile programs which are written in 'c' programming language.

**Why should we compile a program from its source code when the ready to install packages(.rpm and .deb) are available. ?**

## TO TEST THE LATEST VERSION

Software Developers always publish the source code first. We can download the source code and compile it to test the latest version of a program. The best example is the KERNEL itself. To test the latest version of the kernel we have to compile the kernel's source code.

## TO CUSTOMIZE THE SOFTWARE

This is the most useful advantage of software compilation. At the time of compilation we can customize the program according to our needs. Customization allows us to avoid any unnecessary features from the program at the time of compilation. Properly customized program will run faster.Its like building a custom bike rather than buying a new branded bike from the showroom.

**TO RUN DIFFERENT VERSIONS OF SAME**

SOFTWARE.Compilation allows us to install software into a specific directory . So we can install different versions of same software at the same time.

## Basics Of Compilation

Compilation of a software from its source code is done in THREE STAGES. They are given below.

•Running the CONFIGURE script.
•Compiling the program.
•Installing the compiled program.

SourceCode > Customization > Compilation > Installing > PROGRAM IS READY

### ➢Running the configure script:

Every source code of the software contains a script called CONFIGURE. To start the compilation we have to run this script. The CONFIGURE script will do the following things.

Dependency Checking When you run the configure script it will check all the necessary packages which are needed for the program that you are going to compile. If the configure script finds any missing dependencies it will show errors and STOPS working. Then we have to installIf the configure script runs successfully it will create a file called "MakeFile".

Makefile contains all the instruction to compile the software. This file is used in the second stage of compilation.the missing dependencies and run the configure script again.

Customization We provide all customization settings with the Configure script. We must provide the features that we want to enable or disable in the program with the configure script.

Installation Directory There is an option called "--prefix" in the configure script. This option allows us to select a directory where we want to install the program after compilation.
Eg: "--prefix=/sample" this will install program to the directory to /s

If the configure script runs successfully it will create a file called "MakeFile". Makefile contains all the instructions to compile the software. This file is used in the second stage of compilation.

- Compiling the program
  This is the stage where the actual compilation takes place. All the instructions for the compilation are stored in the "MakeFile" created by the configure script. To initialize the second stage we need to type the command "make".
- Installing the program
  This is the final stage of compilation. In this stage we will install the compiled program in the directory that we specified with the "Configure Script" using the "--prefix" option. To initialize the third stage we need to type the command "make install".

## 1. Download httpd source code.

```
#cd  /usr/local/src

#wget  https://dlcdn.apache.org/httpd/httpd-2.4.55.tar.gz
```

## 2. Unzip the file.

```
#tar -xzvf  httpd-2.4.55.tar.gz
```

```
#ll
  drwxr-xr-x. 13  501 games    4096 Mar  1 12:02 httpd-2.4.55
  -rw-r--r--.  1 root root  9758888 Jan 17 19:02 httpd-2.4.55.tar.gz
```

```
#cd httpd-2.4.55/
#ls
ABOUT_APACHE      CHANGES            httpd.mak          Makefile.in        ROADMAP
acinclude.m4      changes-entries  httpd.spec        Makefile.win       server
Apache-apr2.dsw  CMakeLists.txt   include            modules            srclib
Apache.dsw        config.layout      INSTALL            NOTICE             support
apache_probes.d  configure        InstallBin.dsp  NWGNUmakefile  test
ap.d          configure.in       LAYOUT             os                 VERSIONING
build         docs               libhttpd.dep       README
BuildAll.dsp      emacs-style        libhttpd.dsp       README.CHANGES
BuildBin.dsp      httpd.dep          libhttpd.mak       README.cmake
buildconf         httpd.dsp          LICENSE            README.platforms
```

## 3. Install packages

yum -y install gcc gcc-c++
yum -y install libxml2-devel openssl-devel curl-devel libjpeg-devel libpng-devel libicu-devel freetype-devel openldap-devel openldap openldap-devel apr-devel apr-util-devel pcre-devel

## Almalinux: install

```
#  dnf groupinstall "Development Tools"
```

## 4. Once the configure command is created we can use it to configure httpd

```
#mkdir /usr/local/task01
#./configure      --prefix=/usr/local/task01

checking for chosen layout... Apache
checking for working mkdir -p... yes
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking target system type... x86_64-pc-linux-gnu
configure:
configure: Configuring Apache Portable Runtime library...
.........................................................................................................................................................
.........................................................................................................................................................
.........................................................................................................................................................
...............................................................................................
Server Version: 2.4.55
            Install prefix: /clado
            C compiler:         gcc -std=gnu11
            CFLAGS:             -pthread
            CPPFLAGS:           -DLINUX -D_REENTRANT -D_GNU_SOURCE
            LDFLAGS:
            LIBS:
            C preprocessor: gcc -E
```

```
#ls

ABOUT_APACHE        config.layout  INSTALL              NWGNUmakefile
acinclude.m4        config.log     InstallBin.dsp  os
Apache-apr2.dsw     config.nice    LAYOUT          README
Apache.dsw          config.status  libhttpd.dep    README.CHANGES
apache_probes.d     configure      libhttpd.dsp    README.cmake
ap.d                configure.in   libhttpd.mak    README.platforms
build               docs           LICENSE         ROADMAP
BuildAll.dsp        emacs-style    Makefile        server
BuildBin.dsp        httpd.dep      Makefile.in     srclib
buildconf           httpd.dsp      Makefile.win    support
CHANGES             httpd.mak      modules         test
changes-entries     httpd.spec     modules.c       VERSIONING
CMakeLists.txt      include        NOTICE
```

## 5. Next it’s time to compile httpd.

```
#make
```

## 6. Once httpd is compiled it is time to install it. Simply execute following command

```
#make install
```

## 7. Start compiled apache

```
#/usr/local/task01/bin/apachectl start
```