gc-connect Project Setup Guide

Table of Contents

NOTE:
1. The linux system used for this training and project is Redhat / Centos 7
2. Download the said files in your host and scp them into ~/Downloads directory of your VM only if you have installed linux as VM.
3. Please be wary of the single and double quotes while copying commands from this document.

Steps to setup gc-connect project:

Prerequisites – Only on your personal laptop/pc

1. **CentOS 7 should be installed before proceeding with the rest of the steps.**
2. **The VM should be configured for a minimum of 2GB of RAM.**
3. **The CentOS can be downloaded from the following link**
   a. **http://isoredirect.centos.org/centos/7/isos/x86_64/**
4. **Select minimal iso from the page and download**
5. **Use the iso to install centos in your virtual box.**
6. **The graphical UI can be disabled by the following command if the gui starts**
   a. **`sudo systemctl set-default multi-user.target`**
   b. **`sudo reboot`**
7. **Make sure you have sudo permission on your account.**

Installing Required Packages

1. Update the centos using the following command
   a. `sudo yum update`
2. The development tools should be installed
   a. `sudo yum groupinstall 'Development Tools'`
   b. If you face any issues, then try this command
      i. `sudo yum --setopt=group_package_types=mandatory,default,optional groupinstall "Development Tools"`
3. The epel-release should be installed as
   a. `sudo yum install epel-release`
4. Install git 2.x – required for vscode:
   Prerequisites:
      i. `curl-config with $ sudo yum install libcurl-devel`
     ii. `expat with expat-devel`
    iii. `asciidoc with asciidoc`
     iv. `xmlto with xmlto`
      v. `$ git clone https://github.com/git/git.git`
     vi. `$ sudo yum remove git`
   Build:
    vii. `$ make configure`
   viii. `$ ./configure --prefix=/usr`
     ix. `$ make all doc`
      x. `$ sudo make install install-doc install-html`
5. Install Visual Studo Code (vscode)
6. Install openssl
   a. `sudo yum install openssl-devel`
7. Install requests python library
   a. `sudo yum install python-devel`
   b. `sudo yum install python2-pip`
   c. `sudo pip install --update pip`
   d. `sudo pip2 install requests`

Install cppcms

1. Visit http://cppcms.com/wikipp/en/page/cppcms_1x_build and follow the instructions to install the prerequisites
    a. `sudo yum install cmake gcc-c++ gcc make zlib-devel pcre-devel libicu-devel libgcrypt-devel python2`
2. Clone the cppcms source using the command into software
    a. `git clone https://github.com/artyom-beilis/cppcms.git cppcms`
3. The build process
    a. Get into the cppcms directory and follow the commands
        i. `mkdir build`
        ii. `cd build`
        iii. `cmake ..`
        iv. `make`
        v. `make test`
        vi. `sudo make install`

    b. If some of the test fails, then some other application might be using the port 8080 and 8081
    c. Create a file named as cppcms.conf under /etc/ld.so.conf.d/ and type the following (run vi as sudo)
        i. /`usr/local/lib`
    d. After saving and exiting, run the following command
        i. `sudo ldconfig`

Install Oracle XE

1. Download the Oracle XE from the link (login to your oracle account when prompted)
    a. https://www.oracle.com/database/technologies/xe-prior-releases.html
2. Download the Oracle Database 11gR2 Express Edition for Linux x64
3. Copy the zip file to your VM using scp command
4. Log into VM and navigate to ~/Downloads directory
5. Install the following packages (required for oracle to work)
    a. `sudo yum install libaio bc flex libnsl`
6. Extract the zip file
    a. `unzip oracle-xe-11.2.0-1.0.x86_64.rpm.zip`
    b. `cd Disk1`
    c. `sudo rpm -ivh oracle-xe-11.2.0-1.0.x86_64.rpm`
7. The installation should display the progress and complete.
8. Run the following command to configure oracle
    a. `sudo /etc/init.d/oracle-xe configure`
    b. Select default options and when prompted for password, type manager as the password and the configuration will complete.
    c. Edit ~/.bashrc file using any editor add the following line at the end
        i. `source /u01/app/oracle/product/11.2.0/xe/bin/oracle_env.sh`
    d. Restart the VM
        i. `sudo reboot`
    e. Connect to oracle by running the following command
        i. `sqlplus system/manager`

Install Oracle client library and sdk packages

1. Visit this url from your browser
    a. https://www.oracle.com/in/database/technologies/instant-client/linux-x86-64-downloads.html
2. Scroll down to "Version 11.2.0.4.0". When clicking on the links, log into oracle account if prompted.
    a. Click the link "oracle-instantclient11.2-basic-11.2.0.4.0-1.x86_64.rpm", accept the license agreement and click on download button
    b. Click on the link "oracle-instantclient11.2-devel-11.2.0.4.0-1.x86_64.rpm", accept the license agreement and click the download button when prompted
    c. Copy these two rpm files into ~/Downloads directory of the VM using scp command and install them using the following command
    d. In the VM, navigate to ~/Downloads directory and execute the following command
        i. `sudo rpm -ivh *.rpm`
    e. Create a file under /etc/ld.so.conf.d named as oracle.conf
        i. sudo vi /etc/ld.so.conf.d/oracle.conf
        ii. Type the following
            1. `/usr/lib/oracle/11.2/client64/lib`
        iii. Exit vi and run the command
            1. `sudo ldconfig`

Download and install rapidjson

1. Try with $ sudo yum install rapidjson-devel
2. If the above command fails, try downloading the rapidjson package from the link below and scp the rpm file into ~/Downloads directory of the VM
    a. https://download-ib01.fedoraproject.org/pub/epel/7/x86_64/Packages/r/rapidjson-devel-1.1.0-2.el7.noarch.rpm
3. Install rapidjson using the following command
    a. Navigate to ~/Downloads directory
    b. `sudo rpm -ivh rapidjson-devel-1.1.0-2.el7.noarch.rpm`

Download and install rapidxml

1. Download latest nux-dextop-release rpm from
    `http://li.nux.ro/download/nux/dextop/el7/x86_64/`
2. Install nux-dextop-release rpm:
    a. `# rpm -Uvh nux-dextop-release*rpm`
3. Install rapidxml-devel rpm package:
    a. `# yum install rapidxml-devel`

Install boost 1.66 or latest

1. Download boost 1.66 version from https://sourceforge.net/projects/boost/files/
    1. Copy this .bz2 file to the vm
    2. Extract the .bz2 file using
        a. `$ tar -xvjf boost_1_66_0.tar.bz2`
        b. `cd boost_1_66_0`
        c. `./bootstrap.sh`
        d. `sudo ./b2 --with=all install`

Install sqlite3

1. `sudo yum install sqlite sqlite-devel`

Install CppUnit

1. Navigate to ~/Downloads directory
2. Clone the cppunit git repo using the following command
   a. `git clone git://anongit.freedesktop.org/git/libreoffice/cppunit/`
3. `cd cppunit`
4. `./autogen.sh`
5. `./configure`
6. `make`
7. `make check`
8. `sudo make install`

Install cppcheck

1. Navigate to ~/Downloads directory
2. Clone the cppcheck git repo using the following command
   a. `git clone https://github.com/danmar/cppcheck.git`
3. `cd cppcheck`
4. `mkdir build`
5. `cd build`
6. `cmake -DHAVE_RULES=ON ..`
7. `make -j4`
8. `sudo make install`

Note:
The cppcheck utility can be used in the project directory as follows

```
$ cppcheck -i dao/DBFramework --std=c++11 -v --enable=all . 2> cppcheck.out
```

The errors are recorded in the cppcheck.out file.

gc-connect project setup

1. Setup SSH keys in github account
   a. Make sure that your github id is shared with Pratian's operations team.
   b. We will access git using ssh. The following steps are required for that
   c. Run the following command to generate ssh keys
      i. `ssh-keygen -t RSA -C <your mail id>`
      ii. `cat ~/.ssh/id_rsa.pub`
   d. Copy the entire contents of the cat command output
   e. Log in to your github.com account
   f. Click on your account/settings
   g. Click on SSH and GPG keys
   h. Click on New SSH key
   i. Type in your Name in Title text box
   j. Paste the ssh key in the Key text box
   k. Click on Add SSH Key button
2. Clone the gc-connect repo and make sure that the default branch is *dev*
   a. `git clone git@github.com:skillassure-phc/gc-connect.git`
   b. `cd gc-connect`
   c. `make`
3. Run the db scripts
   a. cd scripts
   b. Log into to oracle usins system/manger
      i. Run the db_oracle.sql script
      ii. `@db_oracle.sql`

4. Launch the project binary
    a. `cd bin`
    b. `./gcserver.sh`
5. Launch second instance of the terminal
6. Navigate to the gc-connect directory
7. Run the following command
    a. `python2 unit_test.py`