Compile with EB Xelor

G_ND_PTF_HPC_GEN2

Exported on 04/26/2023

Table of Contents

1 Configure your workspace

2 Install *repo* tool if needed

You will need command line tool `repo.`

Install repo from google and add it to your local path

```
mkdir bin
cd bin
curl https://storage.googleapis.com/git-repo-downloads/repo-1 > repo
chmod a+x repo
# Add the path in the variable PATH (you can add this in your .bashrc)
export PATH=$PATH:~/bin/
```

2.1 Update password if set for the first time

```
password
```

2.2 Refresh package database

```
sudo apt update
sudo apt upgrade
```

2.3 install *git-lfs* if needed

You will need git tool `git-lfs` . It is normally available as package for your distribution. For debian:

Install git-lfs

```
sudo apt install git-lfs
```

2.4 Initialize your workspace

Init your workspace

```
repo init -u git@github-vni.geo.conti.de:bs-g-nd-ptf-hpc-gen2/hpc_gen2.git -b 1.0-dev
-g default,ebxelor --depth=1 --current-branch --no-tags && repo sync -d -c --force-sy
```

```
nc --no-tags && repo forall -v -p -c bash -e $PWD/.repo/manifests/scripts/
repofinalize.sh
```

Hint: If you want the full git history do not use –depth=1.

Hint: If you want to work on the test cases please add the testcases group to the command (-g default,ebxelor, testcases)

> If you need directly to build without adding any feature, jump directly to section Prepareyourenvironment(see page 9)

3 Add your feature to your workspace

3.1 Copy your feature locally

Requirements

Make sure your **workspace is configured** as mentionned above.

Copy sources manually into destination folder

```
git clone <local_git_repository> <destination_folder>
```

u-boot special case

Do not forget to remove *conanfile.py* in *u-boot* to avoid any error!

3.2 Using *repo* tool

Update revision value with your feature branch in .repo/manifests/default.xml

```
<!-- For example a feature on car_s32g -->
<project name="car_s32g" path="ebxelor/pkg/continental/hpc/hpc_car_metha/car_s32g"
revision="your-branch" groups="gold"/>
# Same can be applied to other projects
```

Synchronize once again your sources

```
repo sync -d -c --force-sync --no-tags && repo forall -v -p -c bash -e $PWD/.repo/
manifests/scripts/repofinalize.sh
```

Potential error that shall be ignored

error: svn is different in <target_path>.git vs <destination_path>.git

Repeat these steps for any change you want to test locally.

errors from sync

If there are any errors from the "repo sync ..." command, then please do **not** continue. This needs to be solved before you can build the software successfully. All kinds of strange issues will occur if your sources are not synced correctly.

4 Prepare your environment

> The following steps must be done before you attempt to build.

Your work environment is within a Docker container. To start the container, you can choose to:

1. Start your container from a remote image
2. Build and start the container locally

4.1 0. Prepare Docker Environment

1. Install Docker CE by following instructions: https://docs.docker.com/install/linux/docker-ce/ubuntu/ and https://askubuntu.com/questions/1030179/package-docker-ce-has-no-installation-candidate-in-18-04
2. Create group docker and add your username to this group by running following commands:

```
groupadd docker
sudo usermod -aG docker $USER
```

3. After a re-login/reboot, verify that docker is working with command:

```
docker run hello-world
```

> Issue can appear:
> Unable to find image 'hello-world:latest' locally
> docker: Error response from daemon: Get "https://registry-1.docker.io/v2/": x509: certificate signed by unknown authority.

4.2 Start Docker container from a remote image (default way)

Docker authentication and start the container from remote image

```
docker login https://i-bs-g-nd-ptf-hpc-gen2-docker-l.eu.artifactory.conti.de
./ebxelor/config/docker/continental/hpc/docker.sh
```

4.2.1 Build and start a Docker container locally (only required if you want to modify the container or if there is a problem with fetching the remote image)

Build and start container image locally

```
./ebxelor/config/docker/continental/hpc/docker.sh  --local
```

Docker Image Update

The Dockerfile is located at **ebxelor/config/docker/continental/hpc/Dockerfile**. You can modify this file if you need to add build dependencies for example. If you make ebxelor_environment Git repository part of your MegaMerge pull request, your updated Docker image will be used and validated by Jenkins during the Continuous Integration process.

At docker start, you might be prompted to enter your user and password for Artifactory access.

Please use Artifactory API Key (from https://artifactory.geo.conti.de/ui/admin/artifactory/user_profile) instead of your user password. Otherwise, as your password could contain the special characters, it will not be accepted by artifactory authentication when fetching artifacts. You will get Authentication error.

A .netrc file is created with this information in the folder where you are starting the docker from.
If this file exists at docker startup, you will not be asked for user and password but the information from this .netrc file will be used.

If you don't have right access like :
"Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock"
You need to add your user to docker group with following command:
groupadd docker
sudo usermod -aG docker $USER

5 Build using Valeria

Requirements

Keep in mind that you need to prepare your workspace and your environment as mentioned above before attempting to build.
Make sure your **workspace is configured** and that your **environment is running.**

Install conan configuration and start Valeria

```
# Set up Conan configuration
/workdir/ebxelor/config/conan/continental/hpc/setup.sh
# Run Valeria
/workdir/ebxelor/config/valeria/continental/hpc/run.sh --variant <variant>

#Note: depends on needs, choose a variant from bellow list:
1- gold
2- face_ez1
3- face_ez1_b2.0
4- face_ez1_qemu-arm64
5- face_ez1_qemu-x86
```

5.1 Get the images

The images will be stored in */workdir/ebxelor/config/valeria/continental/hpc/build/deploy/hpc_gen2/bin* as shown below.

```
melali@0245488d40c8:/workdir/build/deploy/hpc_gen2/bin$ ll
total 611356
drwxr-xr-x 2 melali developer      4096 Aug 19 08:23 ./
drwxr-xr-x 4 melali developer      4096 Aug 19 08:23 ../
-rw-r--r-- 1 melali developer       228 Aug 19 08:22 conaninfo.txt
-rw-r--r-- 1 melali developer       389 Aug 19 08:22 conanmanifest.txt
-rw-r--r-- 1 melali developer 209715200 Aug 19 08:22 containers.ext4
-rwxr-xr-x 1 melali developer   2359296 Aug 19 08:22 CORTEXM_S32G27X_car_sw.bin*
-rwxr-xr-x 1 melali developer    197696 Aug 19 08:22
CORTEXM_S32G27X_car_sw_propa.bin*
-rw-r--r-- 1 melali developer    263232 Aug 19 08:22 CORTEXM_S32G27X_hpe_boot.blob
-rw-r--r-- 1 melali developer 209715200 Aug 19 08:22 data.ext4
-rw-r--r-- 1 melali developer    393216 Aug 19 08:22 fw_config_sw0.bin
-rw-r--r-- 1 melali developer    393216 Aug 19 08:22 fw_config_sw1.bin
-rw-r--r-- 1 melali developer     23368 Aug 19 08:22 hper_gen2.dtb
-rw-r--r-- 1 melali developer  16203989 Aug 19 08:22 hypervisor.uimage
-rw-r--r-- 1 melali developer   5748744 Aug 19 08:22 Image_hpe_flash
-rw-r--r-- 1 melali developer  16147954 Aug 19 08:22 rootfs_flash.cpio.gz.u-boot
-rw-r--r-- 1 melali developer 163577856 Aug 19 08:22 rootfs_hv.ext3
```

```
-rw-r--r-- 1 melali developer    266592 Aug 19 08:22 s32g2xx_hse_fw.bin
-rw-r--r-- 1 melali developer    155728 Aug 19 08:22 s32g_pfe_class.fw
-rw-r--r-- 1 melali developer    138456 Aug 19 08:22 s32g_pfe_util.fw
-rwxr-xr-x 1 melali developer    545768 Aug 19 08:22 u-boot.bin*
-rw-r--r-- 1 melali developer    131072 Aug 19 08:22 u-boot_params.bin
```

5.2 Get the symbols

The symbols will be stored in */workdir/.conan/data* as shown below. Following Conan recipe hash can be retrieved in */workdir/build/deploy/hpc_gen2/conaninfo.txt*

```
/workdir/.conan/data$ find -name "*.elf"
./hpc_car_metha/0.0.3/swp/gold/package/b22545d8453017e9007598c17a57a90afaf8f58f/
hpc_car_metha/CORTEXM_S32G27X_car_sw.elf
./hpc_bootloader/0.0.3/swp/gold/package/3e8691d9267ab68788dff783bc70f541a5fcc575/
hpc_bootloader/CORTEXM_S32G27X_hpe_boot.elf
./hpc_car_propa/0.0.3/swp/gold/package/b22545d8453017e9007598c17a57a90afaf8f58f/
hpc_car_propa/CORTEXM_S32G27X_car_sw_propa.elf
```

6 Build Bringup

To build the Bringup variant, you will need to clone the groups default,ebxelor,**bup_ez1** and add the recipe-dir argument to the run.sh script.

```
# To download the sources:
mkdir repo
cd repo
repo init -u git@github-vni.geo.conti.de:bs-g-nd-ptf-hpc-gen2/hpc_gen2.git -b 1.0-dev
-g default,ebxelor,bup_ez1 --depth=1 --current-branch --no-tags && repo sync -d -c --
force-sync --no-tags && repo forall -c 'git lfs pull; cp -LR .git .git.fixed; rm
-fr .git; mv .git.fixed .git'

# To build:
/workdir/ebxelor/config/valeria/continental/hpc/run.sh --variant bup_ez1 --recipe-dir
 bup
```

The sources must be downloaded in a different folder than /workdir (here /workdir/repo).
You will need to commit your local modifications before building.

7 Further information

You can check here for more information : EB Xelor in HPC_GEN2 environment[1]

1 https://confluence.auto.continental.cloud/display/GNDPTFHPCG/EB+Xelor+in+HPC_GEN2+environment#EBXelorinHPC_GEN2environment-Buildenvironment