Common Docker images and Gitlab runners for FPGA development

Adrian Byszuk / SY-EPC-CCE
23 November 2022
Outline

• Gitlab CI basics

• Docker images of FPGA EDA tools

• Gitlab runners

• ATS-IT proposal for common CERN-wide Gitlab CI infrastructure for FPGA development

• Questions?
Gitlab CI/CD basics
Gitlab CI flow

- When developing software/firmware developers make & push changes often, even multiple times a day
- Each change can introduce a bug
- Ideally you should run test suite and build after each change
- Gitlab CI does exactly that
- "Pipeline" is started after each "git push"
- **Pipeline** can execute many **jobs**: tests, build project, generate documentation (Gitlab pages)
Gitlab CI elements

- Two basic elements: **runner** and **Docker image**
- **Gitlab runner** is a computer (physical, VM) that will execute CI **jobs** from a **pipeline**
- **Runner** must be registered to a Gitlab **project**. There can be multiple runners
- **Runner** will start a **Docker container** from the **image** and execute **job** within it
- (**Docker images** aren’t exactly needed for runners, but are recommended)
Docker images of FPGA EDA tools
We've prepared a set of Docker images for EDA software used in FPGA development:

**https://gitlab.cern.ch/cce/docker_build/**

- Available software: ISE, Vivado/Vitis, Petalinux, Modelsim, Questa, Riviera-PRO, GHDL
- Multiple tool versions are supported
- Intel Quartus support on a TODO list
- Extensive README.md as a documentation
- Can be used for CI or in interactive mode (including GUI)
- Most images based on CC7/C8/CS8 so they can access AFS
- Prebuilt images can be downloaded from container registry: [https://gitlab.cern.ch/cce/docker_build/container_registry](https://gitlab.cern.ch/cce/docker_build/container_registry)
Xilinux tools

- ISE version: 14.7
- ...more versions available on demand
- Full install – all FPGA families are supported
- Frequently used additional software included: git, python3, nano, sudo, dtc
- Uses CERN license server by default, but user can easily override
Simulators

- Modelsim: 10.7a
- Questa: 2022.1
- Aldec Riviera-PRO: 2021.04
- GHDL: GCC backend, built from fairly fresh master snapshot
- All images include prebuilt Xilinx libraries: UNISIM, UNIFAST etc.
- Libraries available under `/opt`, pointed by relevant environment variable
- Frequently used additional software included: git, python3, nano, sudo
- Uses CERN license server by default, but user can easily override
How to use it?

Gitlab CI:

- Set the `image` variable in your `gitlab-ci.yml`

```yaml
image: gitlab-registry.cern.ch/cce/docker_build/vivado:2019.2
```

- Some images (*cough*... Vivado ...*cough*) have 80+ GB and take a long time to pull – your CI job may timeout

- Advice – pull all images on your runner during setup and add periodic pull to your `crontab`

- For simulators other than GHDL – please reduce the number of simulation jobs to save on valuable license seats

Interactive mode:

- Login to CERN Docker registry (first time only)

```
docker login gitlab-registry.cern.ch
```

- Pull selected image

```
docker pull gitlab-registry.cern.ch/cce/docker_build/petalinux:2019.2
```

- Start the container with „docker run”

- README contains an example script used to start the container. Depending on your use case, the default parameter set may not meet your expectations

Feedback is very welcome!
Gitlab runners
Setting up a **private** Gitlab runner may be a challenge for teams without Linux administration skills:

- Install the host OS, install the runner, configure executor mode, register with access token etc.
- Constant burden of maintenance

Gitlab server may also provide **shared** runners:

- Set-up and maintained by server administrators (IT team)
- Available to everyone
- Used routinely by software programmers
- Not suitable for FPGA development because of huge Docker images
- ...but why can’t we have nice things?
ATS-IT proposal for common Gitlab infrastructure for FPGA development
There is an ongoing work on a joint ATS-IT proposal about common Gitlab CI infrastructure for FPGA development!

- **This is still a work-in-progress that’s not formally accepted**
- Includes both Gitlab runners and Docker images
- This would bring support level for FPGA workflow on par with software development
- Docker images presented here donated as a basis for official IT images
- IT would prepare a set of shared Gitlab runners, configured for FPGA workflow
- Many technical and organisational challenges to be solved
Many questions...

There are still many challenges and questions that need to be answered:

- VM configuration (many projects need 64+ GB RAM and a lot of disk space, which is unusual)
- Estimate number of potential users
- Impact on available EDA licenses
- Huge size of Docker images creates problems for IT infrastructure
- Cost for organisation
- ...many more