ARTIQ induction - Logistics

Charles Baynham

2024-02-05

---

1 ARTIQ logistics - Git and SSH

Our ARTIQ infrastructure builds on a very mature stack of software tooling, making extensive use of git, GitLab, ssh, HDF5, Nix, InfluxDB, Grafana and more. These are mature projects with excellent documentation online. In this session we will become familiar with the most essential: git.

1.1 Unix / linux

You will need a linux PC to complete this training. If you don't already have one, I recommend dual-booting Ubuntu. If you really can't do this, you can install Ubuntu on a Windows 11 PC using WSL: see https://ubuntu.com/wsl for details. This will allow you to complete most of the tutorial, but visualising ARTIQ outputs using applets will not work. Or, you could install Ubuntu in a virtual machine (e.g. via VirtualBox). This will be slowest, but will allow you to use all features.

You will need to be comfortable with running simple commands at a BASH terminal, e.g. making and deleting files, running commands and changing directories. Remember to use TAB to autocomplete!

Exercises:

1. Install linux if you don't have it already.

1.2 SSH

SSH is a protocol used for making secure, remote connections between computers. It's essential for the AION codebase since Git uses SSH to connect to servers like gitlab.com and github.com, where our code is stored.

In this course we will only use it for accessing repositories on GitLab, howver SSH is a general tool that is regularly used in modern scientific computing. At Imperial, our lab server PC has no monitor, so all access to it is via SSH. We'll need it also for connecting to the Imperial Research Computing Service, forwarding ports from our local network when working remotely, and fetching git repositories from GitLab. Similar systems exist at most institutions (see e.g. https://www.hpc.cam.ac.uk/high-performance-computing or https://docs.bear.bham.ac.uk/bluebear/accessi

1.2.1 Exercises:

1. Generate an SSH public / private key pair (if you don't already have one)

Open a bash terminal and use `ssh-keygen`. Use all the default options.

This will create two files: ~/.ssh/id_rsa and ~/.ssh/id_rsa.pub. The first one is secret - you should not ever share it and it should not leave your computer. The second is your public key - you can share this with people and they can use it to confirm that you have access to the private key.

2. Give gitlab your public key and use this to clone a repository in the next section.

1.2.2 Success means:

- "I have my own SSH certificate. I understand the use of a public and a private key."

1.3 Git

Git is the modern standard for storing, working with and collaborating on code. It is used ubiquitiously in software engineering and ARTIQ relies on it heavily. It allows you to:

- Store a history of snapshots of your code at moments in time (*commits*)
- Refer to specific versions of your code unambigously using *hashes* of *commits*
- Work with other people collaboratively by *merging* your changes into existing code
- Use tools like GitLab to discuss changes before they are made and automatically perform actions on code (such as creating this PDF document, running a set of tests to make sure your code still works, or rebuilding an ARTIQ bitstream)

In AION, we use git to store all our collaborative projects in the AION group on GitLab: https://gitlab.com/aion-physics. You should already be a member of this group. If you are not, ask Charles to add you now.

Putting some time into learning Git will pay you back at a very good exchange rate when you later save yourself from lost or mangled code, track down the change which caused a strange bug to appear, need to replicate a result for a thesis that was produced with code from months (or years) ago, etc. There are lots of good git tutorials around - I suggest the first two chapters of the Git Book (https://git-scm.com/book/en/v2).

1.3.1 Exercises:

1. Read the first two chapters of the Git book.
2. Navigate to https://gitlab.com/charlesbaynham/poetry-competition.
3. *Clone* the poetry competition locally using SSH (`git clone`).
4. Make a new *branch* (`git checkout -b xxx`).

5. Add a poem to the repository (add a file in "docs" and alter "docs/index.md") and save this as a *commit* (`git add .` + `git commit`).
6. Submit this as a *Merge Request* on GitLab, marking it as a draft (`git push` and read the message).
7. Use GitLab to discuss your changes and ensure that the *pipeline* did not break. Fix any problems that arose.
8. Mark the Merge Request as ready to request that your poem be added to the list.

1.3.2 Success means:

- "I can pull repositories from GitLab via ssh"
- "I am comfortable with git on the command line. I know what a commit is, what a remote is, how to push and how to pull"
- "I know what a commit / branch / remote is. I can clone, push, pull and fetch, and know when to do so"
- "I can handle merge conflicts when they arise. I know how to avoid them"
- "I know the difference between git and GitLab. I can use GitLab to discuss code changes with others before I integrate them into a project. If a pipeline exists for a project, I know how to view the results"

1.4 Development environments and Nix (advanced)

Nix is an advanced environment manager, capable of building deterministic software environments in any mixture of programming languages. This is important if you e.g. want your dark matter detector to work the same way when you replicate an experiment a year later, but someone has published a new version of `numpy`.

Nix is extremely powerful but out of the scope of this course. We will use it to run an ARTIQ environment, but will not attempt to learn it today. If there is interest, we can cover this in a future course. If you're ahead and would like extra work, try to reach these goals:

1.4.1 Success means:

- "I understand the semantics of the Nix language enough to open a .nix file and know what it means." (https://nixos.org/manual/nix/stable/language/index.html)
- "I can build simple Nix environments using flakes. I can add pre-packaged python modules to my environment and command-line tools"
- "I can use Nix to replicate an AION ARTIQ environment and make simple modifications to e.g. add more python packages" (https://gitlab.com/aion-physics/code/artiq/artiq_training)