

FarmData2 Onboarding Activity 02

FarmData2 Developer Install

Introduction:

The overarching goal of this course is to provide a large-scale open source software development experience in FarmData2. This will ultimately include enhancing your ability to learn new technologies, but learning the technologies used by the project and contributing code that implements new features. But before that is possible the first step is to get the developer environment for FarmData2 up and running. The developer environment will allow you to make, observe and test code modifications that lead to the new features. At the completion of these activities, you will have a functional FarmData2 development environment.

FarmData2 Installation:

1. What file in the FarmData2 repository provides information about how to install FarmData2 as a developer?

The **Development Platform** section of the file you found in question #6 gives the instructions for setting up the development platform for FarmData2. The recommended development platform is Linux running in VirtualBox. However, if you would like you can attempt to setup directly on Linux, MacOS or Windows without VirtualBox. But if you do, you will need to be prepared to debug any problems that arise.

2. Setup a working development platform for FarmData2. If you are working with a partner, both partners must setup a development environment. If you encounter any difficulties or have questions while doing the developer install consult the **install stream** on Zulip. If you do not find your answer there, create a new *topic* and ask a question. Monitor this channel during this activity and respond to any questions that you can help with. We are all members of the FarmData2 community and building up a record of problems encountered, and their fixes will help everyone.

3. Provide answers to the following questions in order to confirm that you now have a working developer platform. If you are working with a partner provide a separate answer for each partner.

a. What is your development platform? Include version numbers for the OS and VirtualBox (if you are using it).



b. How much RAM is allocated to your development platform? This will be the amount dedicated to VirtualBox (if you are using it) or installed in your machine if not.

c. How much free disk space (in MB or GB) is available on your development platform? You may need to use Google to figure out how to determine this. Give the command you used or briefly explain how you determined this?

d. What versions of git, docker and docker-compose are installed on your development environment? Give the command you used or briefly explain how you determined this?

e. What is the URL of your fork (origin) of FarmData2 on GitHub?

f. Log into FarmData2 as manager1 and paste a screenshot of your browser window below. Use "Edit" -> "Paste and Match Formatting" to paste an image inside the box.

g. What are the names of the docker containers that are running after you have started FarmData2?

h. What output is generated when you run the script for stopping FarmData2? How is it related to your answer from g?



i. Restart FarmData2. Open another browser tab and connect it to the provided IDE (see “Editing Code & Documentation”) in the install directions. Use the IDE to open the source of the INSTALL.md file from your local repository. Paste a screenshot of the browser with this source file open in the IDE below.

Documentation Contributions:

4. Optional but encouraged: If you noticed any typos, missing steps or things that could be clarified in the `INSTALL.md` file create a feature branch, make proposed fixes for them, push the branch to your origin and create a pull request so that the upstream can be updated. Your contributions will improve the install process for everyone.

5. Optional but encouraged: Review the issues in the issue tracker that are tagged “Documentation” and “Good First Issue.” Given your familiarity with FarmData2 at this point, these issues should be quite approachable. Identify one where you think you can make an improvement and:

a. Create and checkout a feature branch for you work on the issue.

b. Make changes to the documentation to resolve the issue. If you are unable to fully resolve the issue in what you consider a reasonable amount of time, go onto part c and d with what you have complete.

c. Push the branch to your origin.

d. Make a pull request to the upstream for your branch. Describe what you have done in the body of pull request message. Be sure to reference the issue number in the body of your pull request as described here:

- <https://docs.github.com/en/free-pro-team@latest/github/managing-your-work-on-github/linking-a-pull-request-to-an-issue>

If the changes in your pull request do not fully resolve the issue use the body of the pull request to describe what you have done and what remains to be done.

