Deliverable 1

I suppose I should preface this with a hearty *"I have 10% of an idea what's happening and I'm learning this on the fly."* With that said, witness my fantastic ineptitude in the coming paragraphs.

I have a virtual machine through Google Compute Engine's cloud VM service, running the last stable release of Ubuntu. I've learned a handful of things about the Linux shell/command line/bash environment through the class tutorials as well as floundering around in the VM environment for a different class, so I dove in to building the Amara project on the VM pretty much headfirst. Tremendous genius and master of bash that I am, I go to the project's GitHub and download the developer build .zip file and uploading it to the VM. Manually. I then spent a good solid 30 minutes figuring out how to unzip the darn thing before I discover that I have to install something to unzip files (shocker, I know.) Thankfully, my mastery of Linux led me to the correct directory to install, and a quick *$sudo apt-get install* command later (and about 5 minutes of unzipping), I've got the project in my VM's directory ready to go.

As most of you probably guessed, it didn't work. After a few minutes of frantically going back and forth between *$ls* and *$cd,* I finally notice the big fat *README.markdown* sitting right in the top level directory. The README was obviously designed to help geniuses like me, and laid out exactly what I needed to do to *actually* put the project on my VM effectively. After trying *$git clone* a few times and staring at the output super confused I decide I should probably follow the link in the README about setting up a public/private key for my GitHub account. As I'm sure no one else guessed, that magically fixed my authentication error (after I discovered that apparently pressing "Ctrl + A" while viewing a read-only .ssh key file evidently increments the first integer by one rather than selecting all the text).

At this point, I have the project and the appropriate submodules from Git, and run the next command in the README: the Amara project uses a container called Docker to aid their development project. A quick *$sudo apt-get install docker-compose* leads to my first successful step without a Google search and only mild confusion. The next step leads to the final end goal: building the darn thing to see if it works. A bit of floundering around to figure out how to interpret their vague "bin/dev build" instruction (which didn't work on its own) I finally figure out to run *$./bin/dev build* from the project directory to build it. It took me a few minutes to figure out I did this correctly, though, as it still didn't work. The error message simply states that my Docker version is off, so I had to update it. I knew some of those words, and started trying to figure out what combination of --*update* or --*install* to add to the *dev* command would update it. The answer was none of them. I decided next to just look at the Docker documentation, which stated that in the Docker file I had to change the version number from "1" to "2" and indent everything once to make it work. I knew some of those words.

A quick search and some more intense confusion later, and I find the Docker file to edit. The only problem: the format already matches the format given in the Docker documentation, except the version is set to "3." I decided to change that to "2" to see what would happen; suddenly, everything works! Although I had no earthly idea why that was so simple yet so complicated at the same time (and I still don't have any idea), I've got the project built. The next instruction in the README, *$./bin/dev dbreset,* apparently heard that I had some success and decided not to work. Apparently I lacked appropriate permissions to edit the database, which made some sense, but didn't particularly explain why it was in the quick start guide. I'll have to figure out what that does in the future, but this instruction was strange to say the least.

Ignoring the rest of the README like a true master, I decide to just jump right in to the *./bin/dev test* to run the Amara team's prebuilt test set. After telling me the tests couldn't be executed because something wasn't configured properly, the tests began to run anyway. I'm not entirely sure how that issue corrected itself as the shell never said anything about correcting the issue, but the tests ran despite my confusion. Following their completion, I got this error message:

```
                                    FAILURES
SubtitleTypesTest.test_subtitle_list self =
<tests.subtitles.test_types.SubtitleTypesTest
testMethod=test_subtitle_list>    def test_subtitle_list(self):l =
SubtitleFormatListClass(ParserList, GeneratorList)>
self.assertEqual(len(l), 13 E
AssertionError: 9 != 13
/var/run/amara/tests/subtitles/test_types.py:32: AssertionError
1 failed, 1121 passed, 3 skipped in 243.29 seconds
```

I know some of those words.

All in all, the project is built, the tests ran somehow, and all but 4 out of 1125 tests passed, which is a fairly high passing rate. However, to interpret the tests and their results, all the user is given is a brief *"consult the documentation for details on the tests,"* a phrase which adds so little to aid in interpretation I question why it's there in the first place. There's no reference to *where* in the documents I should look, no brief summary of what the test was even *doing* in the first place, and the only indication of what happened is the console's enlightening discovery that nine is not equal to thirteen. While I'm glad the project is built and working, in at least some sense of the word, I think the developers are operating on such a high level of assumptions and lack of documentation *within their code* that it's difficult for an outsider to get oriented even before initially building it. Granted, I will mention again that I am entirely inept, but still, assuming your user has certain packages within their Linux build without so much as mentioning what packages they rely on is a *bit* of a big assumption for an open source project constructed mostly by volunteers. At the very least, it would make the quick start guide a bit more quick.