# An exercise in Discovery, Building Docker Images, using Makefiles & Docker Compose. – Part 1

## Docker images, Multi stage builds, Makefiles Projects with sub directories and their own Makefiles.

**Overview**

This all started with another [blog](https://medium.com/@georgelza/an-exercise-in-discovery-streaming-data-in-the-analytical-world-part-1-e7c17d61b9d2) I was writing, where in I was exploring Kafka, kSqlDB, Apache Flink, Apache Hive, Apache Iceberg & Apache Paimon and various other supporting bits and bobs.

During that “rabbit hole” I ended needing to build “some…” docker images, leading to allot of learning along the way and redoing things over and over to make it easier, faster, simpler.

Figured others might be interested, find value out of it.

I’m by no means an expert, this is purely what I learned along the way to make it easier for myself.

We will cover a basic docker image build, installing basic OS packages, installing application server (OpenJDK 11) and Apache Hadoop DFS cluster, adding required configuration files (directly into the container during build or mounting them at run time) & dealing with environment variables.

As a separate mini project, I will also show multistage build to reduce size and attack vectors.

During this we will show how to stage application deployments locally first and pulling them during each build & additionally the importance of the build order in your Dockerfile.

As a sidetrack we will also have a look at using Makefiles, they are a great way to “package all your commands together.

To demonstrate the above process, we will build an Apache Hadoop DFS cluster, based on Ubuntu 20.04 / OpenJDK 11 / Apache Hadoop 3.3.8, comprised out of:

* namenode
* nodemanager
* resourcemanager
* datanodes &
* historyserver

As it stands this will be a X-part posting, but it’s by no means complete.

NOTE: I work on an Apple MacBook based on their ARM64 aka AARC64 architecture. Where needed I will point out which lines can be changed to make everything Intel/AMD64 compatible.

Good luck, as always, this is all fraught with rabbit holes, so many and you can disappear so easily… But it’s all fun and you will always discover something new or validate a previous learned skill.

See [my GIT repo for the entire document and code/article.](https://github.com/georgelza/MongoCreator-GoProducer-avro.git)

**About Me**

I’m a techie, a technologist, always curious, love data, have for as long as I can remember always worked with data in one form or the other, Database admin, Database product lead, data platforms architect, infrastructure architect hosting databases, backing it up, optimizing performance, accessing it. Data data data… it makes the world go round.

In recent years, pivoted into a more generic Technology Architect role, capable of full stack architecture.

[George Leonard](https://www.linkedin.com/in/george-leonard-945b502/)

[georgelza@gmail.com](mailto:georgelza@gmail.com)

------------------------------------------------------------------------------------------------------------------

# An exercise in Discovery, Building Docker Images & Docker Compose. – Part 2

<Part 2>

**My Repo’s**

<Comment about the Repo>.

<https://github.com/georgelza/MongoCreator-GoProducer-json>

In the main root branch of the repo is an infrastructure directory where all the source images are build using docker, which is then utilised inside the various versions of the project.

See the [Makefile](https://opensource.com/article/18/8/what-how-makefile" \t "_blank) in the same directory, which will first pull the source images, after which the various images can be build. Remember to go change the base OS architecture if you not on a MBP and also look at the JAR files pulled, some of them are arm64/aarch64 specific.

Below is a diagram depicting the ancestry of the various images used inside the project.

And that’s it for now… Thank you for sticking with me through this exploration. All it did was create a small little to do list that is growing as I am typing this, of things/subjects I’d like to explore more and blog . Till next time.



**About Me**

I’m a techie, a technologist, always curious, love data, have for as long as I can remember always worked with data in one form or the other, Database admin, Database product lead, data platforms architect, infrastructure architect hosting databases, backing it up, optimizing performance, accessing it. Data data data… it makes the world go round.

In recent years, pivoted into a more generic Technology Architect role, capable of full stack architecture.

[George Leonard](https://www.linkedin.com/in/george-leonard-945b502/)

[georgelza@gmail.com](mailto:georgelza@gmail.com)

Some more References:

[Apache Flink](https://flink.apache.org/) originally by [Ververica](https://docs.ververica.com/)

* [Get Started - Installation](https://docs.ververica.com/vvp/getting-started/installation?_gl=1*a1ub31*_gcl_au*MjI3NTE0OTU0LjE3MjA2OTY4NDY).