YAML:

- YAML is a human-readable data serialization language.
- It is commonly used for configuration files, but could be used in many applications where data is being stored (e.g. debugging output) or transmitted (e.g. document headers).
- YAML targets many of the same communications applications as XML but has a minimal syntax which intentionally breaks compatibility.
- It uses both Python-style indentation to indicate nesting, and a more compact format that uses [] for lists and {} for maps.
- This files can have either the .yml or .yaml extenstions.
- yml files can sometimes be annoying to write because:
    - YAML is case sensitive.
    - YAML does not allow the use of tabs.
    - Spaces are used instead as tabs are not universally supported.
- If you feel stuck use a YAML validator which can be found freely online, e.g: http://www.yamllint.com/

http://yaml.org/

Docker compose:

- Compose is a tool for defining and running multi-container Docker applications.
- With Compose, you use a Compose file to configure your application's services.
- Then, using a single command, you create and start all the services from your configuration.
- Compose is great for development, testing, and staging environments, as well as CI workflows.
- Using Compose is basically a three-step process.
    - Define your app's environment with a **Dockerfile** so it can be reproduced anywhere.
    - Define the services that make up your app in **docker-compose.yml** so they can be run together in an isolated environment.
    - Lastly, run **docker-compose up** and Compose will start and run your entire app.
- Volumes key can be added to the compose file (yml) to mount the project directory (current directory) on the host to /code inside the container, allowing us to modify the code on the fly, without having to rebuild the image.

https://docs.docker.com/compose/