Why create a Docker Image

* Docker images allow for developers to create their programs and run them in containers which include all the dependencies needed to run the program. This will allow developers to run and share code with any system that is compatible with Docker. It fixes the issue of “this works on my local machine but not on anybody else”.

Docker Image Documentation

1. Download docker desktop

* Windows link [Install Docker Desktop on Windows | Docker Docs](https://docs.docker.com/desktop/install/windows-install/)
* Mac link [Install Docker Desktop on Mac | Docker Docs](https://docs.docker.com/desktop/install/mac-install/)
* Linux link [Install Docker Desktop on Linux | Docker Docs](https://docs.docker.com/desktop/install/linux-install/)

1. Docker Documentation

* [Dockerfile reference | Docker Docs](https://docs.docker.com/engine/reference/builder/)

1. Unique Distinctions

* ENV HOST=0.0.0.0 is defined to allow all network interfaces to be able to use this image
* EXPOSE 8080 exposes port 8080 but you can change it to any port you’d like. The default in the program is 8080

1. Building the Image

* Docker Desktop needs to be open and running
* In the cmd line type “docker build -t Image\_Name”
* SOMETIMES it can fail to build the image because of network errors so if it cannot find the pdf latex and the additional latex packages, you may have to rerun the build command

1. Tag/Push the image

* You must Tag the image before pushing it into the artifact registry in google cloud
* To tag the image “docker tag IMAGE\_NAME REGION-docker.pkg.dev/PROJECTID/ARTIFACT\_REGISTRY\_NAME/IMAGE\_NAME”
* To push the image the cmd is “docker push REGION-docker.pkg.dev/PROJECTID/ARTIFACT\_REGISTRY\_NAME/IMAGE\_NAME”

1. Deploy Image onto Cloud Run

* Go to Google Cloud Console
* Go to the Artifact Registry
* Go to the name of the Registry you pushed the image to
* There will be a small bar with the image, and you can choose to deploy to cloud run from there