Started off by setting up the resources I knew I needed

Setup

* Installed docker
* Used powershell to create a volume
* Created the container and finished setup on localhost:8080
* Created a jenkinsfile and the C++ program
* Setup a GitHub repository

After setup was complete, I looked up how to compile a C++ program within the pipeline.

I found various libraries for this task such as Conan, but I also found some forum posts about compiling C++ within the command line, so I did more digging and found a video which used this principle.

The video uses 3rd party software to host Jenkins along with scripts, I first tried following this video.

I created a Pipeline and linked it to a GitHub repository which would house my Jenkinsfile and other dependencies.

I then created Jenkinsfile, which at this point only had 2 stages build and run.

At this point I knew I wanted to use g++ to compile the program.

So, I decided to try and install packages first using a dockerfile, however it was taking too long to remake the container each time and I could not get it working at the time so I decided to try another approach.

I then used the command line. I accessed Jenkins as root using this command docker exec -u 0 -it CONTAINER\_NAME /bin/bash

Then used the command apt-get update and apt-get install -y build-essential g++ to update current packages and install g++ from build-essentials.

I then tested the simple Jenkins file to see if it compiled the program and ran it correctly and it did.

Some more quick research showed me how to archive the artifacts.

The only thing left was to write a script to interrogate the API to display the status and metrics

I then researched how to access Jenkins API and decided to create a python script around this research as I was also comfortable with the language.

I found out all you need is the endpoint, token, and job name.

After the response is checked, the Json is saved to variables and outputted to the screen