

# Gradescope Autograder Configuration

Matt Russell

May 20, 2022

# Introduction

Gradescope is great tool for autograding assignments. However, there are two difficulties regarding the setup for an autograder which this document addresses:

1. How to streamline the general setup of autograding.
2. Provide a grading framework to develop tests in an efficient (and enjoyable!) manner.

Setup from start to finish is intended to take roughly 30 minutes. If you have any questions, please reach out to me at [mrussell@cs.tufts.edu](mailto:mrussell@cs.tufts.edu). Thanks!

## Infrastructure Background

Gradescope's autograders rely on Docker containers which are spun up each time a submission is graded. The default container runs a variant of **Ubuntu 18.04**, coupled with the bare-bones scripts to make the autograding framework function. The usual workflow is to manually upload a **.zip** file containing two scripts: **setup.sh**, which installs dependencies (e.g. **Python**, etc.), and a shell script named **run\_autograder**, which runs the autograder. The main issue here is that each time you upload the **.zip** file, the Docker container must be built from scratch, which can take quite a bit of time; this can compound quickly during the development of an autograder. This document provides an optional solution to the problem.

## Autograding Background

Once the container is built, there is of course the issue of how to run and test student's code. This is no easy task! This document provides documentation on an autograding framework we have developed which makes writing tests for student code as easy as possible.

# Infrastructure Setup

The solution for streamlining the infrastructure setup with Gradescope is twofold:

1. Build and upload our own Docker container to Dockerhub, which Gradescope will use.
2. Put the autograding code in a `git` repository which the Docker container can access at autograding time.

Note that these two elements are distinct from one another; if using Docker is something you really don't want to do, that's fine. The container building will just take more time in the aggregate. In that case, follow the instructions on Gradescope's website regarding setup:

<https://gradescope-autograders.readthedocs.io/en/latest/specs/>. Note that you can still use the `.git` integration from above; just integrate the `git` commands from the Docker setup below into the `setup.sh` script. However, if you wouldn't like to do that either, okay! Just skip ahead to the **Autograding Framework** section below.

## Install Docker

Install Docker Desktop: <https://www.docker.com/products/docker-desktop/>  
Note that you don't need to have it start on boot; you can start it before uploading the setup.

## Setting up the Autograding Repo

If you don't currently have a repository related to course material, please make one. We suggest using `gitlab` for this: go to <https://gitlab.cs.tufts.edu>, and login with LDAP, using your Tufts eecs `utln` and password. You do not

need a `README`. The example below will be for `cs 15`, but please follow the instructions for whichever course you're running. Now, in your terminal:

```
mkdir cs-15-autograding
cd cs-15-autograding
git init
git remote add origin git@gitlab.cs.tufts.edu:your_utln/
    path_to_your_repo.git
git switch -c main
```

We have a sample repo for you to get starter code from. Copy the files as follows:

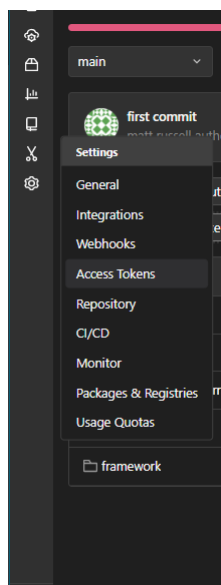
```
git clone git@gitlab.cs.tufts.edu:mrussell/gradescope-
    autograding
rm -rf gradescope-autograding/.git
mv gradescope-autograding/* .
rm -rf gradescope-autograding
```

## Configuring the Docker build

Great! Now you have both the grading framework, as well as the elements necessary to build the Docker container for gradescope. We will need to do a few configuration steps to make this work. First, `cd Dockerbuild`. We will need to add three files here (more details for each are below):

- `.repopath` - the remote path of the repository, including an Access Token.
- `.dockertag` - the tag of the Docker container to build
- `.dockercreds` - the credentials to login to Dockerhub.

### `.repopath`



First, go to `gitlab` in your browser, and navigate to the course repository you just created. Next, hover over the settings cog on the lower left, and select 'Access Tokens'. Create an access token; this will be used by the Gradescope autograder to pull the most recent version of the autograding files for an assignment. We suggest only providing 'read repository' access to the token. Feel free to select whatever you'd like for the name, expiration date, and role (Maintainer is fine).

Once the token is created, copy the key. Now, open a file named `.repopath` [in the `Dockerbuild` directory]. You will want to format the repository path as follows:

```
https://REPOSITORY-NAME:ACCESS-TOKEN@gitlab.cs.tufts.edu/path/
to/repository.git
```

For example:

```
https://cs-15-2022uc:glpat-Blah8173Blah8023Blah@gitlab.cs.tufts
.edu/mrussell/cs-15-2022uc.git
```

Great! Now the autograder will be able to pull the most recent version of the autograding files.

### **.dockertag**

This will be the tag you'd like to use for your Docker container. Open a file named `.dockertag` and write:

```
tuftscs/gradescope-docker:YOURTAGNAMEHERE
```

Feel free to use anything in place of `YOURTAGNAMEHERE`. Note that the first section is required.

### **.dockercreds**

We are using a single Dockerhub account for all of the autograding courses. The file `.dockercreds` should be available in the course's Tufts Box folder. If not, reach out to me at `mrussell@cs.tufts.edu` from your Tufts email address, and I'll send it to you ASAP.

## **Conclusion**

Okay, you are ready to begin developing an autograder! Continue to the next section to learn about the autograder, and for a walkthrough to setup an assignment.

# Autograding Framework

## Introduction