Setup and Configuration of Concourse

Understand Concourse: Concourse is an open source continuous thing-doer. It’s a continuous integration and continuous delivery system for teams that practice agile development. It was developed by Pivotal Engineers with frustration of existing CI system. The central concept of Concourse is to run tasks. It is built on the simple mechanics of resources, tasks, and jobs.

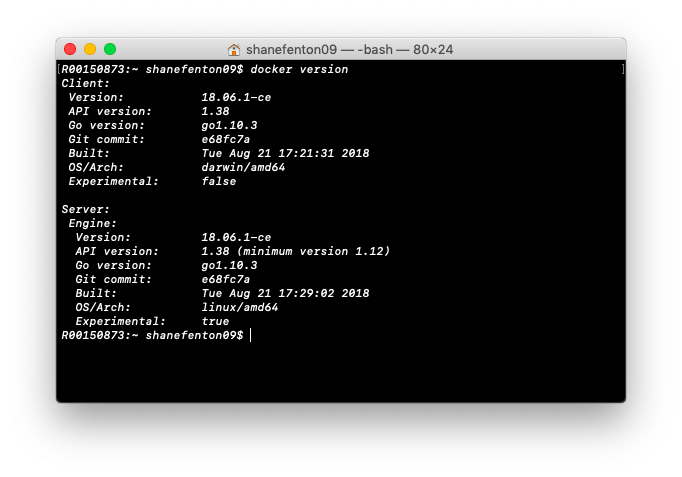
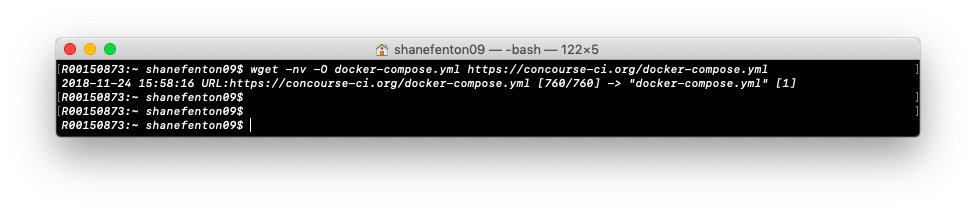
Concourse can be used when you need to

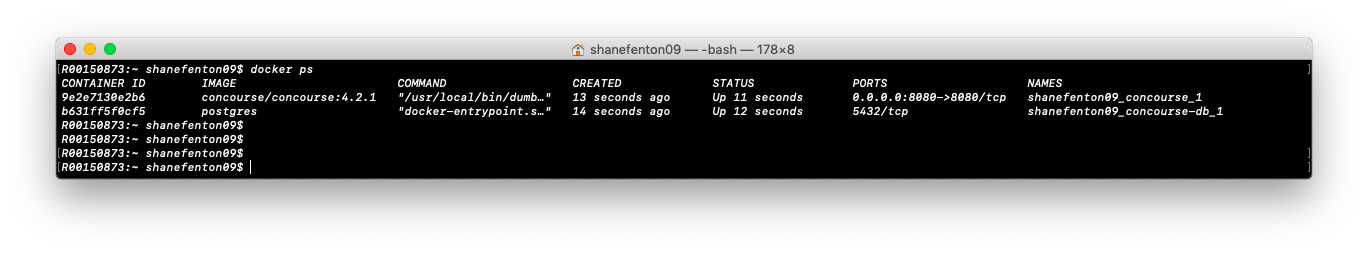
* Automate test driven development.
* Maintain compatibility between multiple build versions.
* Target multiple platforms and configurations such as different clouds.
* Deliver frequently – weekly, daily, or multiple times a day.

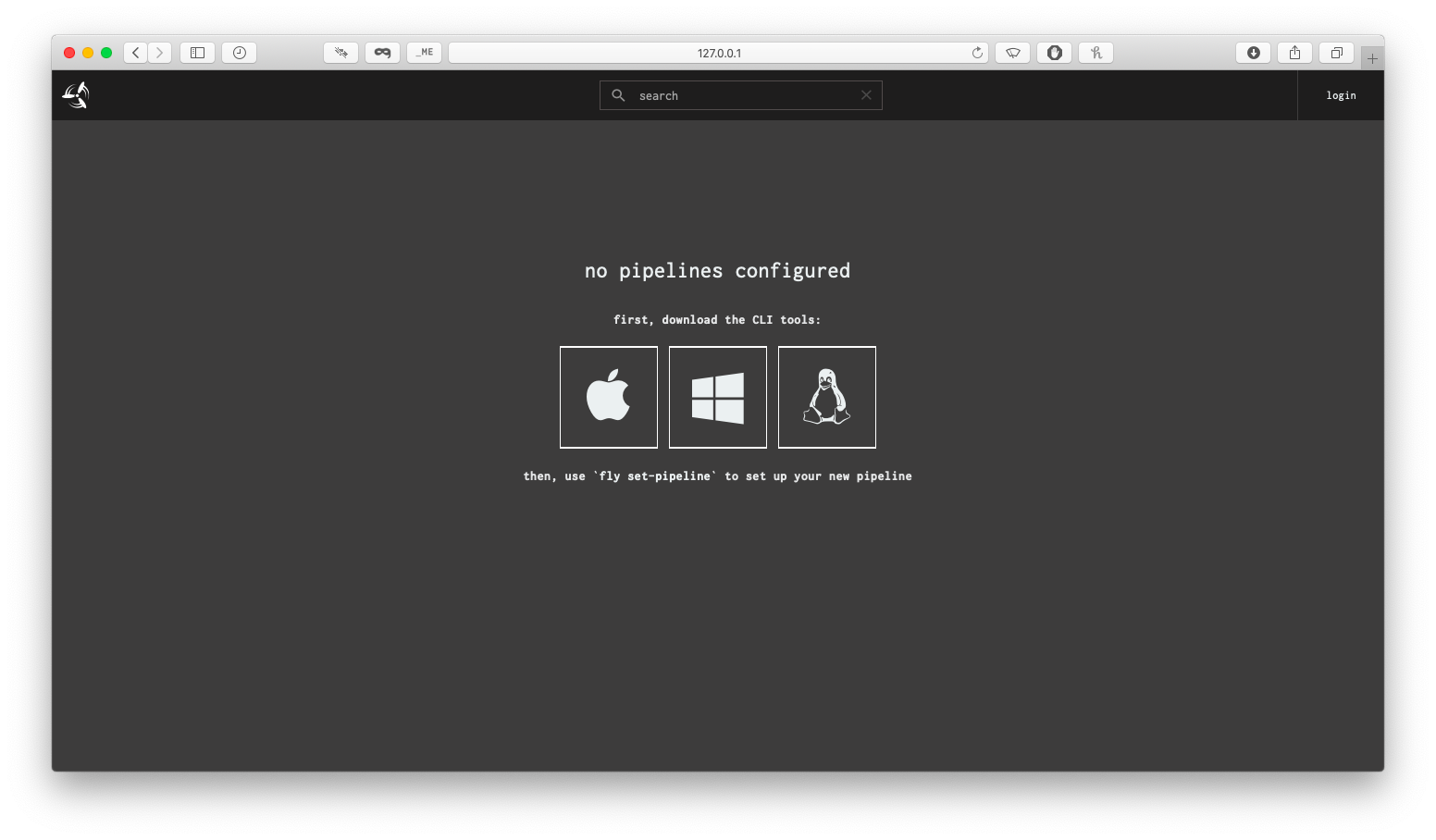
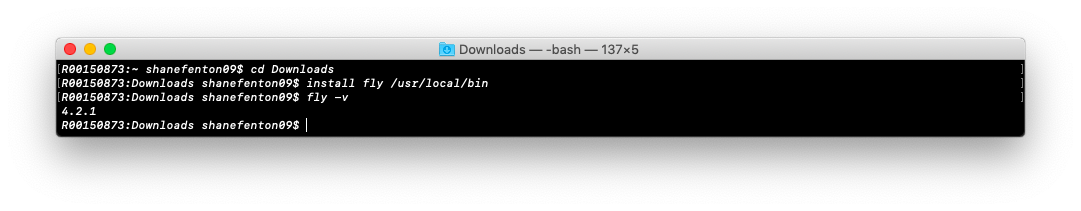
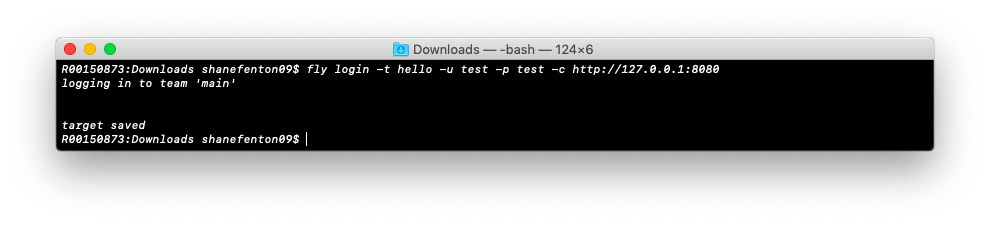
Pipelines:

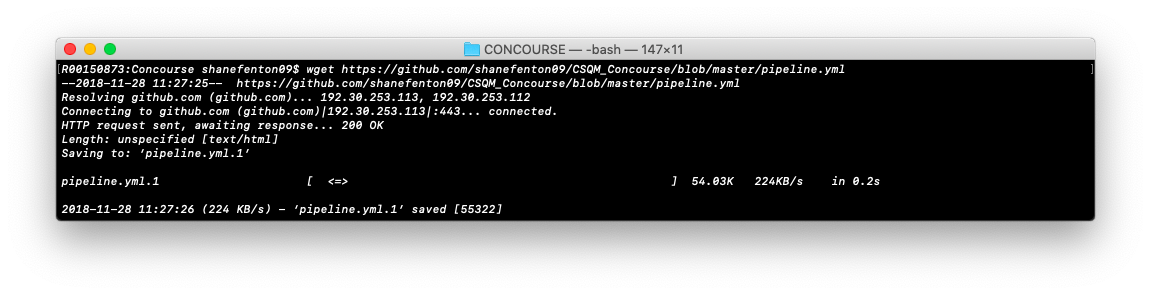
A pipeline is the result of configuring jobs and resources together. Pipelines are configured as declarative YAML files. YAML is a human-readable data serialization language. Each entry under resources is a dependency, and each entry under jobs describes a plan to run when the job is triggered.

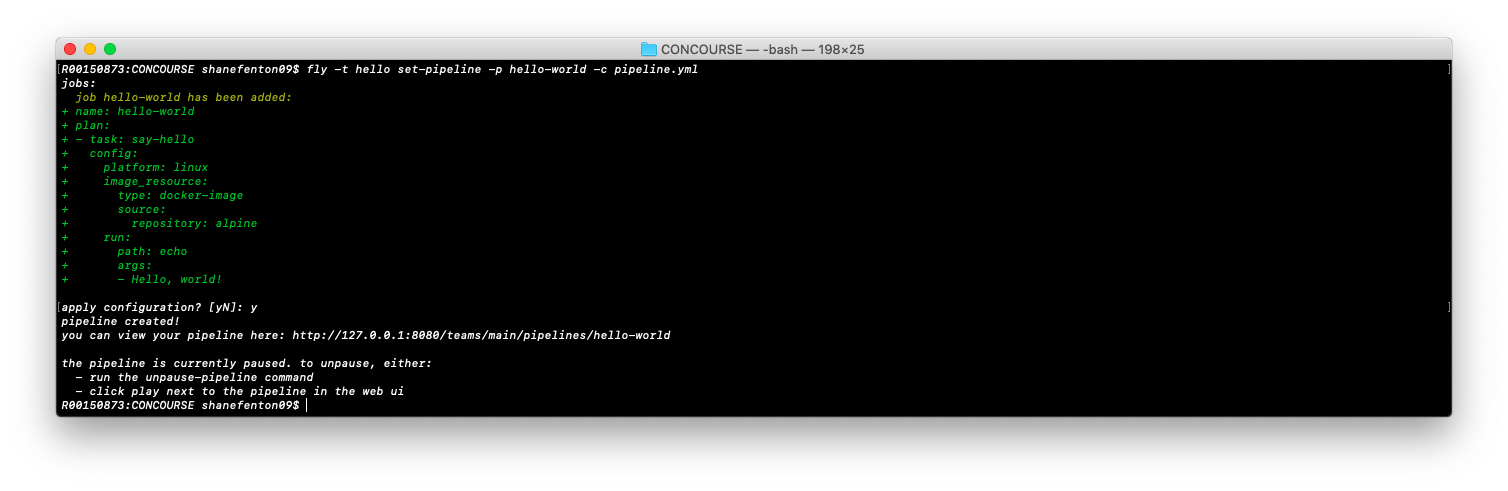
Configuration:

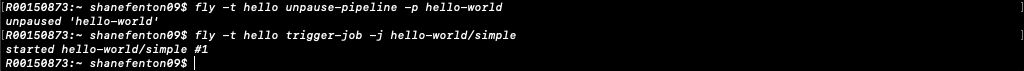
1. Download Docker for Mac. Docker provides a way to run applications securely isolated in a container. It allows you to build and share containers and automate the development of pipeline from a single environment.
2. Now to set up Concourse, we downloaded the quick start Docker Compose file provided by Concourse. I use the wget command to download this file.

Next, we run “docker ps” to show that docker has setup two Docker containers, one for Concourse, and another for Postgres. All the ports, permissions and credentials have been taken care of.

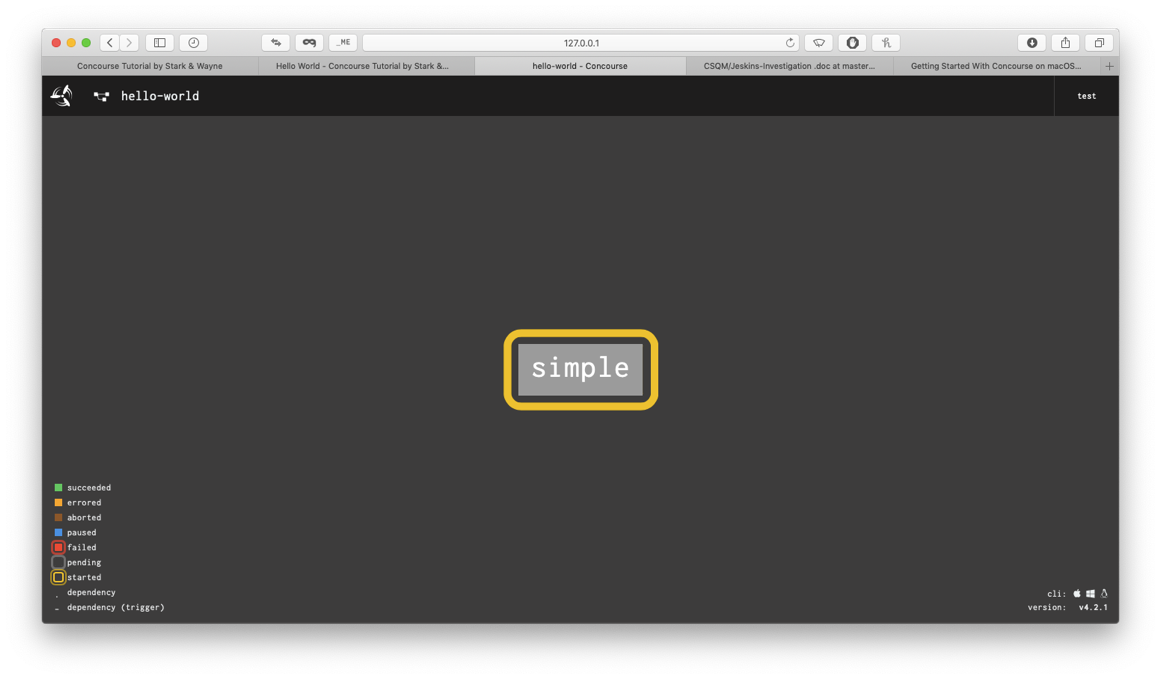
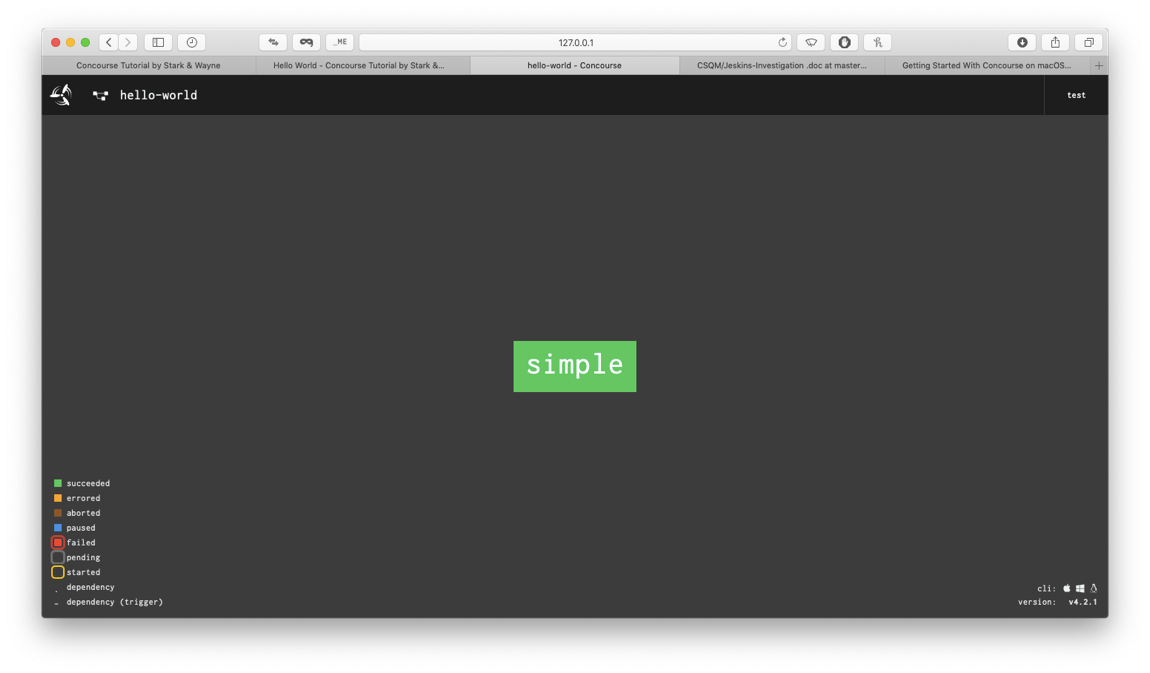
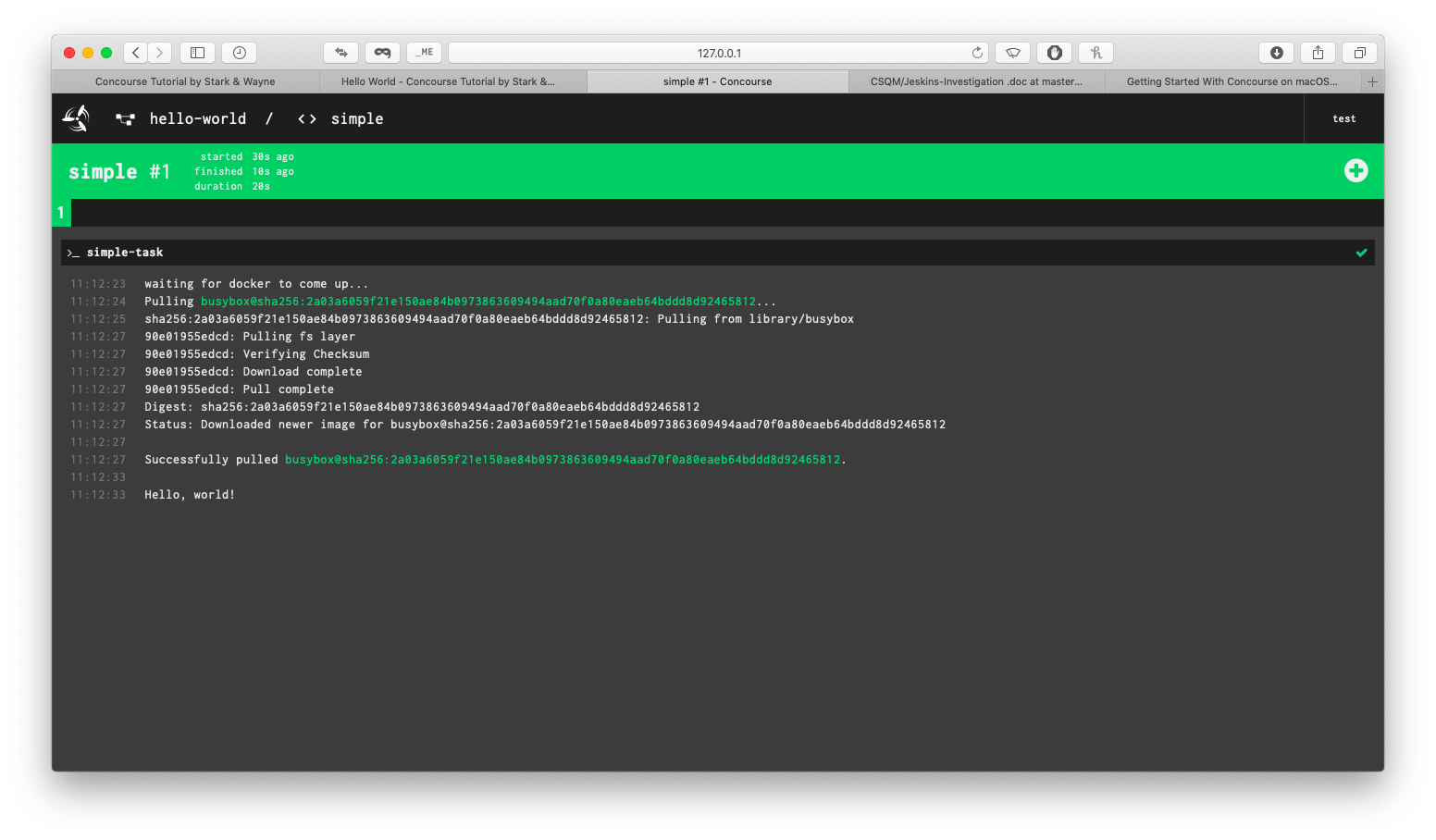
1. To access Concourse, I use <http://127.0.0.01:8080/> as the host address.
2. To install the Concourse CLI(fly) on my system, we click on the Operating System we are working with and download a file. We then change our terminal to view inside the downloads folder, and install fly.
3. We login using the fly login command.
4. Setting up a pipeline:

To set up a pipeline we get a YAML file from an associated Github, in the form of a raw file. We use the wget command to get the .yml file.

1. We then use the fly command to set the pipeline.
2. We then un-paused the pipeline and set the trigger.



1. These are the results, with the different colours corresponding to the status.



Evaluation of Concourse

Installing Concourse on Mac with Docker was straight forward, however we had trouble with upgrading fly to the correct version due to the lack of tutorials online and forums. Fly is the Concourse command line interface (CLI). The lack of tutorials and forums on sites such as Stack Overflow, slowed the installation process down until we could figure out the problem. We were installing fly in a directory stated by the Concourse Documentation, but this was outdated. That specific directory was removed in the latest Mac update.

After Concourse is installed and you have parallel versions of both Concourse and Fly (Current Version: 4.2.1), you can start to create a pipeline. Concourse uses YAML format to configure pipelines. Tasks are written as Linux Scripts, and this is where we ran into problems since no one is familiar with Linux scripting. This meant that it was difficult to create a script that was integrated with the Python code we committed to the shared repository.

We found sample YML files online through the Concourse documentation, and were able to test these sample to understand how Concourse works. Concourse has a web interface that could be shown on a screen, where it’s very easy to see the current state of a job, as it colour coded to green and red. This is a good advantage as it would quickly alert a member of the team, that something has failed.

Overall, the setup of Concourse was easy, but creating pipelines with specific jobs and tasks was difficult due to a lack of experience with Linux Scripting.