Datagetta Documentation

**TO-DO List**

* Move the docker containers to datagetta\_servers
* Teaford wants practice data to be put into the system. The problem arrives in the csvparser. What happens is that the fields of the table from trackman are not completed filled out. You need to write a script to fix this in order to let them be added to the system.
* The system currently has a txt file that holds all the names of the files that are on the database, but that does not work correctly. If there is ever an error with one of the files and it is not uploaded, that is still updated with the file name. So, this needs to be fixed. The ideal fix would be to add a logging system as well, and to not add the name if it is actually added to the database.
* The current Orange (Pitching run values) machine learning models do not work. We are unsure if they actually used any machine learning. Feel free to learn from the Blue models (Defensive Shifting) in order to get started. Coach Teaford will be able to give you more information on how it should actually run.
* The login function needs to actually be implemented.
* The current ml models could be improved, potentially with Batch normalization and epochs.
* Combine the documentation
* Need to ensure that the models are running off database data and not previous data csvs.
* Look into explainable AI for the system
* Look into giving the models a term for how much the model believes in its outcome
* Possibly add health metrics for players on the website

**Getting Started**

First, you need to get some sort of Linux terminal to easily access the datagetta server. The datagetta server is hosted through Auburn University.

The server name is datagetta.cse.eng.auburn.edu and to login, on your linux terminal you must write

$ ssh [username]@datagetta.cse.eng.auburn.edu

To receive a username, you must contact Kelly Price (pricekg@auburn.edu), and ask her for an account in order to access the DataGetta project. She will give you a username and password, which then you put your username in place of [username] in the above script. Then you will be asked for a password. Do not worry if you cannot see yourself typing the password, because that is just a linux functionality.

You must also be on the Auburn Wifi, so if you are not on campus, you must use a VPN. Look up Auburn’s VPN, and you will most likely find that you need to download Global Protect. Download that application, and you will then be able to log in.

Most users will have sudo access when they log in. You should only access other’s accounts if it is really needed. And be careful with sudo access, as you can mess up the system by doing any command that you cannot reverse. Sudo gives you administration access essentially. To access that, do the script

$ sudo -s

This will start a screen that gives you complete access to move around the entire system. However, if you run certain commands that are tied in with the username, it will mess up, as your current user is ‘root’.

The docker containers are currently hosted on csg0026. They need to be officially moved to datagetta\_servers. So, when the github repo you are working on is ready, clone it into the datagetta\_servers account, and you will be able to start running from them. But, make sure you have stopped the containers running in csg0026.

We are trying to host everything on datagetta\_servers because it is a no-login account. Anyone with an account can access this account, so this will prevent as much confusion in the future, as it should be obvious that everything is hosted on this account.

There should be no permission problems inside of datagetta\_servers except for some docker commands (possibly). You will be needed to be added to the docker group to be allowed to complete docker commands. You can do it yourself by typing

$ sudo adduser [YOURUSERNAME] docker

**Docker**

To list all the containers in a person’s account, you can run

$ docker ps -a

OR

$ docker info

 To get more info about them, you can run

$ docker inspect [NAME]

If you ever want to update the server and docker containers when you have updated your github repo, then you must log into the server, go to /home/datagetta\_servers/project and run the makefile with

$ make -f [Makefile]

While on the server, you can access what is inside the docker containers. If you do

$ docker exec -it [CONTAINERNAME] /bin/bash

Then this will give you an interactive terminal for the container you chose. There are three different containers.

If you are wanting to see the postgres one and you use this command above, then you can use the command

$ psql -U dbgetta -d datagetta\_db

In order to access what is inside the database. These commands below may help you get started inside of there.

$ \dt

$ SELECT \* FROM teams;

$ SELECT \* FROM "trackman\_pitcher" WHERE "PitchUID"='126690f0-cc13-11ed-861f-6df92acd5975';

To get into the python one, you would do

$ docker exec -it python /bin/bash

Then you can run this, with configuration to its config file (config.yaml), in order to start manually pulling files from the postgres database

$ python3 ftpPuller.py check\_pull

**Trackman**

Trackman is a system that tracks all the live physical data from baseball games and puts it into a server. An easy way to see the server and use it is through an application called FileZilla. This will allow you to pull any information from it and see its layout.

You can pull down some of the historical data from our file sharing program with TrackMan. These CSV files not only include the games played by Auburn, but include games from all teams in the data-sharing program. Some stadiums have player positioning CSV files that will also download as a part of this, which points towards fielder location and shift detection. You can access and download all CSVs from the FTP. Below are instructions on how to access your FTP via FileZilla - You can also access the FTP through your Windows Explorer folder. FTP IYour FTP account can be used to access your Trackman data in csv form. We recommend using FileZilla as your FTP client. Download "FileZilla Client" from (<https://filezilla-project.org/>)

Once you have FileZilla downloaded, please use ftp.trackmanbaseball.com as your Host. Enter your Username and Password and press "Quick connect.” Folders containing your data in csv form will appear on the right-hand side of the screen (v3,Year, Month, Day, CSV) Unverified csvs will be delivered within 30 minutes of submission Verified data will be delivered no more than 24 hours after submission

Username: Auburn

Password: kA#R2,KNAP

Adam may be able to help you if you have any questions about the system here: [arn0037@auburn.edu](mailto:arn0037@auburn.edu)

**Cron Script**

You need to eventually NEED TO delete /var/spool/cron/crontabs/csg0026 when you are done and are able to start the datagetta\_servers docker containers.

Users’ crontab files are named according to the user’s name and these Crontabs are stored in the directory /var/spool/cron/crontabs

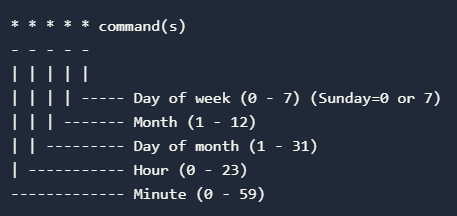
The /etc/crontab file and the scripts inside the /etc/cron.d directory are system-wide crontab files that can only be edited by the system administrators

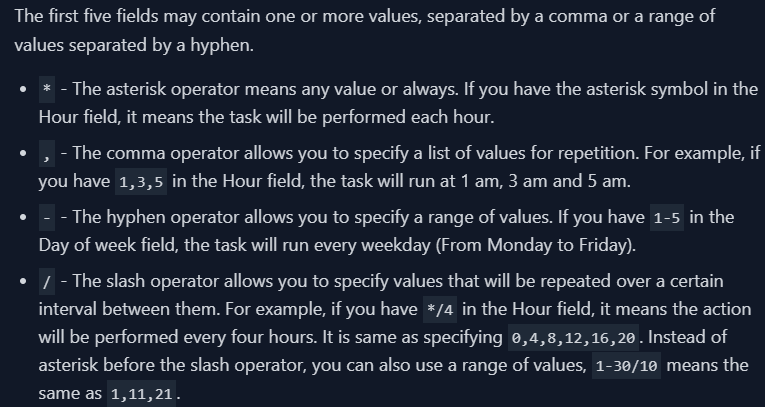
Then in the etc folder, you will notice the folders cron.d, cron.daily, cron.hourly, cron.weekly, cron.monthly

In most linux distributions, you can put scripts inside these different folders and they will be executed every hour, day, week, or month

There is also a crontab file.

You will see this



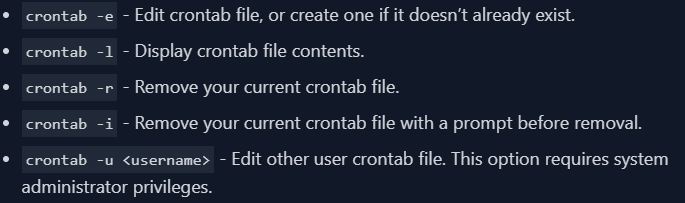


 We need to make the model rerun every monday (2am) and wednesday (2am)

So…

$ 0 2 \* \* 1,3 root cd /home/datagetta\_servers/project/datagetta&&./cron-script.sh

crontab allows you to install, view, or open a crontab file for editing



Cron is a daemon that allows you to schedule tasks at a specific date and time.

In computing, a daemon (pronounced DEE-muhn) is a program that runs continuously as a background process and wakes up to handle periodic service requests, which often come from remote processes. The daemon program is alerted to the request by the operating system (OS), and it either responds to the request itself or forwards the request to another program or process as appropriate.