Quicksched Documentation

The view into the guts of the system

Team Magisters

May 1, 2022

Contents

1 Hosting

1.1 Bitnami

Bitnami is a preconfigured instance of the lightsail AWS package. Many headaches with webhosting are solved for you in order to make as minimal possible steps for the lab organizer to set up their instance-based application. The preconfurations include:

- Opening up the proper ports
- Updating and installing apache for ubuntu
- Setting up template virtual host files for the wsgi application
- Setting up the wsgi server for apache to host

With these in mind, this makes the setup of the install script very simple. The end user of bitnami only has to complete a few tasks after installing bitnami. These are taken care of by the description of the install script down below.

1.2 Idea of the install script

The install script is designed so that the lab organizer will have to do as little work as possible in order to be able to host their own instance. As bitnami covers a lot of ground for hosting, the install script and the user guide top off the workflow so that even those that aren't tech savvy could accomplish it. The steps of the install script go as follows:

1. Create the secret key for your Django instance (for security)
2. Configure the template virtual hosts to set up the apache and wsgi server.
3. Install the python decouple in order for the settings to be imported via the envar file.
   (a) The envar file adds additional security so that the settings.py file does not have plaintext authentication info
4. Deploy the Django files to the directory specified in the vhosts
5. Set up the new Django database with new superuser
6. Make it so the apache user in ubuntu has ownership of the files to properly serve them

1.3 The email setup script

This is merely a means to append the necessary gmail login information into the envar file. This will allow the lab organizer to operate the TA recruitment via a free gmail account.

2 Django App and File Structure

To briefly overview, the main contents of a Django web application consist of these python files:

- Models
  - Essentially the classes of the system, as they help translate python classes into tables into the database
- Views
  - The dynamic code acting in the backend to properly load webpages with html context.
- Templates
  - The html pages that the user interacts with (Uses some bootstrap, javascript and JQuery)

This application is divided into several different applications that intermingle together to form the wsgi application. The contents of each of the applications are strung together via the urls.py file located in the "quicksched" directory located within the project. We will briefly go over the different modules (in Django they are called apps) one by one and briefly explain their functionality.

2.1 Authentication

This part of the application sets up the accounts and creates them for them to be put into the database. Upon recruiting TA emails, the authentication application creates the accounts and allows them to login to the instance as TA's. It also allows for some other necessary features such as user creation overhaul and password channges.

2.2 Email Upload

This Django app handles the recruitment of TA's. This accesses the Authentication app to allow it to create accounts according to the Emails that is sent in. The main function of this app and its views is to read an uploaded CSV file, make accounts for the TA emails in the CSV, and send them emails to welcome them and give their temporary password.

2.3 Laborganizer

The views and models for this app handle everything lab organizer related under the sun (except the email recruitment and algorithm). The models here contain everything to labs, semesters and schedule management all available at the laborganizer's fingertips. Everything related to TA management can be done in this app and the admin page.

2.4 Optimization

This app is essentially the app that generates the best schedules for the TA's to be assigned to. It generates scores for all TA's to each class then assigns them based on those constraints.

2.5 Teaching Assistant

This is the app that handles all of what the TA needs to see in terms of their schedule. The other main functionality is that they are allowed to change their scheduling info such as their time availability and qualifications in order for the optimization algorithm to take them into account.

3 Model Structrue

Here, we will notate the relationships between the models and how each model is structured in the database. As stated above, each model here is to be thought of as a table in SQL.

3.1 Authentication

The models in this app overhaul the user management default to Django. It overhauls the creation of superusers and regular users by making it so the email for each is the primary key for the accounts. This will help the email upload app easily create accounts that are uploaded to it via a CSV.

3.2 Email Upload

There are only 2 models in this app, as they are placeolders for the new and returning accounts. They merely contain an email text field. The point of these models is to easily distinguish the returning and new email accounts in the database when there are new TA recruitments.

3.3 Laborganizer

The lab organizer models are very complex, as they serve as the bedrock to the optimization algorithm and the database information. The two main models are as follows:

- Semester
  - The semester model is a holder of the different labs. All entities (TA's, Labs, etc) are assigned to a semester via a foreign key mapped on their own model. The semester itself has helper functions along with the season and time it is assigned.
- Lab
  - The lab is the table that contains all necessary information about a lab such as its name, times, ids, etc. The lab assigns itself to a semester via a foreign key. When the lab organizer creates labs, the labs are by default assigned to the currently active semester.
- Allow TA Edit
  - This is just a model created for convenience of development. It is a placeholder model that sets a specific time for a TA to edit their scheduling information. This is flipped on and off by the lab organizer via the GUI. A TA assigns itself to this table via a many-to-many field. More about the TA will be explained down below.
- LOCache
  - This is another placeholder model for development convenience. This caches the states of the lab organizer's current workflow in the GUI in order to speed up the process of managing the TA's and template schedules.

3.4 Optimization

The models in this app contain the template schedules required for the lab organizer. The template schedules serve as buffers for the actual propogated schedule in the database. Along with these, it contains the history model, which acts as a linked list that records all of the lab organizer's manual changes to the schedule.

- Template assignment
  - This model marks a lab and a TA to be a temporary assignment in the current template schedule. It is also marked with a schedule key, being the version of the template schedule it is assigned to.
- Template schedule
  - Same idea as stated above. The algorithm is built into the model as a method accessible by the views in the lab organizer app. This model contains a many to many field of assignement models along with a semester foreign key and a version number of the schedule.
- History
  - Acts as a node of chance in the lab organizer dashboard. The lab organizer can undo them if any changes recorded were unwanted.

3.5 Teaching Assistant

- TA
  - The TA holds fields for all of its experience, assignments, scores and holds. Along with its information about itself, it serves as a fundamental model for the schedule generation and how the schema works.
- Score Pair
  - This is just a model that holds a TA's score assigned for a specific lab via the algorithm. They are assigned a semester and a schedule key, while TA's hold a many-to-many field of them.

4 Lab Organizer Dashboard Aspects

4.1 TA selector

This module acts as a template schedule generator for the lab organizer. Depending on which TA's were checked, the optimization app will generate scores for each of the relevant TA's assigned to the semester and then assign them to the schedules accordingly. This will in turn generate new Template schedule and assignment models, and display them to the lab organizer via the view. The current active schedule version will be cached into the database.

4.2 History

The history acts as a linked list of saved changes, whether it be schedule switches or manual assignments on the Lab organizer's part. The History will cache up to 10 nodes and changes before deleting the oldest one. Clicking the undo button deletes the first node and reverts the switch or manual assignment on the backend.

4.3 Semester Selector

The semester selector merely changes the active semester and loads all of the template schedule models attached to it.

4.4 TA Management

This is the tool to edit the database information about a specific TA. It is also the gateway to select the recruit more TA's option, which will allow the Lab organizer to upload a CSV of new TA emails.

4.5 Semester Management

The semester manager allows the lab organizer to edit lab information like they can TA information. The edits will be overwritten into the database.

4.6 Admin Page

ABANDON ALL HOPE ALL YE WHO ENTER HERE. The admin page is where the LO can make direct edits to the database without any buffers or any warnings. We do not recommend making very many edits here or if at all. It may break quite a few things and is there for nuclear options only.

4.7 Switches

By utilizing JQuery, the dashboard can recommend schedule switches for each of the assigned TA's. By clicking the switch button, the panel will recommend 5 switches that would be the best contenders for the lab. The switches module assumes that the schedule generated by the algorithm is the best possible schedule, and anything deviating from the scores set would be a bad idea. The deviation score due to the switch will be shown to the lab organizer, and color coded to show its severity. If it is red for example, that means it would be a very poor switch, as that TA may not be able to attend the lab they are switching to.