

Sugarizer School Box



About Me

Name: Rishabh Nambiar

Email: rishabhn@protonmail.com

Sugar Labs wiki username: Rishabhnambiar

Github: <https://github.com/rishabhnambiar>

IRC nickname: rishabhnambiar

Languages: English

Where am I located? and what hours(UTC) do I tend to work?

I'm located in Mumbai, India (UTC +05:30).

I'll be graduating just when the coding period begins so I will be able to treat this position like a full-time job.

Work Hours: 5:00 AM to 1:00 PM UTC

Past experiences with Open-Source projects:

I have worked with an open-source organization called [ERPNext](#) sporadically between 2016 and 2017 as a part of their Developer Operations team.

- I added an AWS S3 Integration to Frappe, a Python framework that powers ERPNext.
<https://github.com/frappe/frappe/pull/4272>
- Here's a PR I made that has some Ansible:
<https://github.com/frappe/bench/pull/473/>

I've been making minor contributions to open source projects since May 2016.

- <https://github.com/frappe/bench/pull/473> [Ansible]
- <https://github.com/coding-blocks/content-downloader/pull/7> [Python]
- <https://github.com/fossasia/meilix-generator/pull/110> [HTML, CSS, JavaScript]

Project Description



Goal #1: Sugarizer Raspberry Pi Images with build scripts

The first goal is to create a Raspberry Pi image and build scripts based on Raspbian that can serve Sugarizer to a classroom full of students.

On booting, the Raspberry Pi will:

- act as a WiFi AP that clients(browsers, sugarizer apps) can connect to.
- serve Sugarizer using Sugarizer-server on the local WiFi so there will be no requirement for Internet connectivity for now.
- start a browser session running Sugarizer client so a single user/instructor can use it by connecting a display to the Pi.

This is being created because the Raspberry Pi is inexpensive, widely available and many such Sugarizer School Boxes can provide a cost-effective way of reaching out to more students and communities through Sugarizer.

Improvements to the Internet-In-A-Box [Ansible playbook](#) for Sugarizer will also be made in this section.

This will include making sure that the IIAB playbook uses the latest version of sugarizer with all available functionality and ensuring that it works.

Collaboration & Testing

Sugarizer already works really well when deployed locally on a Raspberry Pi and Neighbourhood view is also functional.

On completion of Goal #1, the image build will be tested at real Sugarizer deployments with the help of the community.

Goal #2: Heroku Button

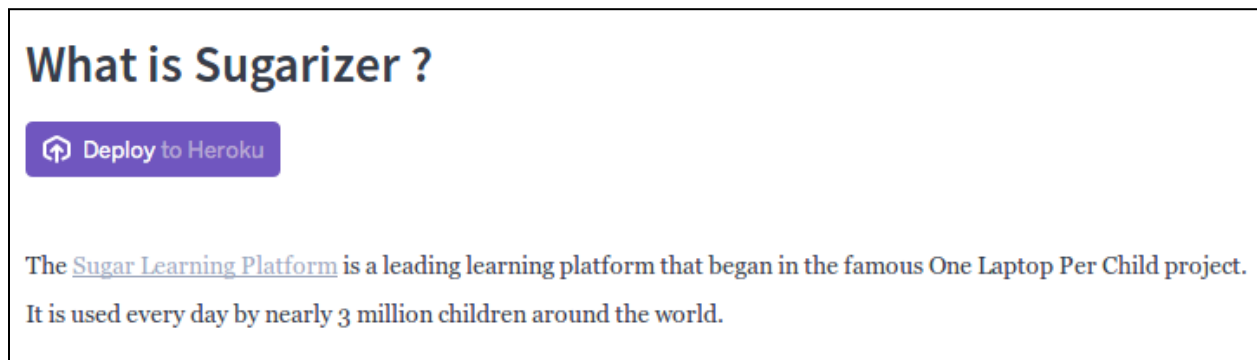


Fig. mockup for a Heroku Button in README.md

The second goal is to simplify the way to deploy Sugarizer in the cloud. The priority in this section is get a [Heroku Button](#) for Sugarizer up and running.

A Heroku Button is a simple HTML or Markdown snippet that can be added to READMEs. Clicking a Heroku Button will take you through a guided process to configure and deploy an app running the source code referenced by the button. This is being done to provide users a simple, one-click and hassle-free deployment method that will let anyone set up their own instance of Sugarizer without opening a terminal window.

Minimum Deliverables

The community has suggested that I should treat improving the Sugarizer School Box as Goal #1 and the deployment scripts as Goal #2. I will start working on the deployment scripts only when Goal #1 is complete.

As 90% of the project should ideally be complete by the 2nd evaluation phase, the Raspberry Pi build, the improved IIAB installer and the Heroku Button script will be ready by then.

Best Case Scenario

If things go as per the timeline till the 2nd evaluation phase, I will also be making a Bash + Docker script for sugarizer deployment.

Ansible, Terraform or Packer can be added to the script to allow compatibility for different VPS providers. As the Docker build for Sugarizer-server and Ansible installer for IIAB already exist, this will not take long.

Timeline

Date (2018)	Period	Task
April 23 - May 13	Community Bonding	• Set up a blog for posting my weekly progress. • Discuss and reshape the Timeline after discussion with my mentor(s). • Investigate the causes of MongoDB crashes faced by Sugarizer users and formulate a solution.
May 14 - May 21		• Analyze Sugarizer performance on the Raspberry Pi using parameters like file download size for every page/activity, frequency of page visits, processing required per request and seeing how the Pi server behaves in different conditions. I plan to use Locust, a user load testing tool for the analysis. • Create a Raspbian Image build for the Pi3 that has an Access Point and Sugarizer-Server running on boot. • Use hostapd and dnsmasq to create an access point as shown in this article.
May 21 - May 28		• Continue to work on Raspbian Image build and test on my Raspberry Pi 3. • This project will involve writing some bash for the build and I'll use these resources for inspiration: • https://github.com/sugarlabs/rpi23-gen-image • https://github.com/iiab/iiab-factory/tree/master/box/rpi
May 29 - June 04		• Continue to work on the Raspberry Pi image and the build scripts and publish them. • Improve the Ansible playbook for installing Sugarizer on the Internet-In-A-Box installer. • This will include making sure that the IIAB play uses the latest version of sugarizer with all available functionality and ensuring that it works.

June 05 - June 11		• On the IIAB Sugarizer install, add safeguards and checks to reduce MongoDB corruption issues caused by incorrect shutdown methods. • Study how to make the Sugarizer School Box communicate with the Android and iOS apps. • Implement and test on multiple Android devices/tablets.
June 12 - June 20	Phase I Evaluation	• Formulate a robust upgrade process to ensure that Sugarizer updates are installed successfully without losing any data to instances that have been deployed. • Explore pm2 as a solution to do solve this or use other features of pm2 to enhance reliability. • The aim is to remove the possibility of data loss during after updates.
June 21 - June 29		• Generate a Heroku Button for one-click Sugarizer-server deployment. • The project should ideally be 90% complete by the end of June as stated in the proposal template. • So my goal at this stage is to have a functional Raspberry Pi image build for Sugarizer, significantly improved IIAB Ansible install script and the Heroku Button.
June 30 - July 08		• Test Heroku Button deployment. • Test the Sugarizer School Box Raspberry Pi image at real Sugarizer deployments with the help of the community.
July 09 - July 13	Phase II Evaluation	• If the Turtle/Music Blocks release is on schedule, I will add it to the Sugarizer Pi Image build. • We can use this release as a test for the upgrade processes created in Week 5. • Improvements to the upgrade process will be made.

July 14 - July 20		• Write a Bash + Docker script to deploy Sugarizer-server on a VPS. This will be done the traditional DevOps way using Ansible. • The Ansible component will not take much time because a similar play has already been written here.
July 21 - July 28		• Continue working on the Bash + Docker script. • Use Packer or Terraform on the script to improve compatibility between VPS providers.
July 29 - Aug 5		• Test deployment options thoroughly on different VPS providers. • Take help from the community for testing the scripts on a larger scale.
6 Aug - August 14	Final Evaluation	• Keep a final buffer week for added tasks. • Prepare final submissions.

Convince us why you can finish this project

I believe I can complete this project because of my love and fascination for Linux and my previous work experience with Ansible, Docker and Python.

- I mentioned my experience with an open-source organization called [ERPNext](#) above.
- In the summer of 2016, I worked as a full-time intern for the same organization. I added automated AWS S3 Backups to their in-house deployment tool, [Central](#).
- This included writing Ansible playbooks and Python that would set up automated backups for all their production servers for their enterprise clients. This was NOT an open source contribution but I'm mentioning this here because it is a major reason for my familiarity with Linux and Deployment.
- Some presentations depicting my work at ERPNext can be viewed [here](#).
- In the summer of 2017, I was at [Cube](#) as a Full-Stack Development Intern. Over the summer, I built a Python based (bottle.py) web application that automated a lot of maintenance tasks for the Operations team that included web scraping and API Development. I also containerized the application using Docker and Docker-Compose.

This is the first time I'm applying for GSoC and Sugar Labs is the only organization I'm applying to because this project is a really good fit for my skill-set.

The Project and the Community

- **Rishabh Nambiar** rishabh@protonmail.com, [Twitter](#)

If the project is successfully completed, it will provide Sugar Labs a more portable and inexpensive way of deploying Sugarizer. The Raspberry Pi is inexpensive, widely available and Sugarizer School Boxes can provide a cost-effective way of reaching out to more students and communities through Sugar. This method of deployment requires only one Raspberry Pi 3.

The added convenience of using one-click deployment scripts can be instrumental in getting non-developers to using a project and using a Heroku Button is even better when you don't even have to use to a terminal to have your own instance of Sugarizer.

Answers from the Community

- **Michaël Ohayon** - mohayon75@gmail.com (Potential GSoC Mentor)

1) I think that the Pi will be able to handle the load, we won't have that much kids connected to the same Pi and the networking process is not that heavy.

We should try to launch the server in background and display a web browser in foreground. If the load is too high we will see it quickly.

2) Yes, you're right, the Pi should provide a Wifi AP to allow devices to connect. This AP should bring routing from Ethernet if connected.

3) Everything is possible, Ansible is a great tool. Combined with Terraform and Packer we should be allowed to deploy things without having troubles handling multiple cloud providers. The installation script could be performed using simple bash and docker.

4) I think you can start thinking on your proposal. You can continue to discuss with us to talk about the project but also the community and the impact of that project.

5) For the deployment part we have two options that would be nice to have.

- a really simple automation like heroku single click deploy (<https://blog.heroku.com/heroku-button>).
This is the killer feature we should definitely have.
- the devops way using tools like Ansible and Terraform and Packer.

One major thing for all platforms is to think about the upgrade processes, how can we update the devices/server without losing data.

- **Tony Anderson** - tony_anderson@usa.net - <http://schoolserver.org/>

This is exactly how the xsce server works so you may get valuable help from that community (xsce or iiab).

A continuing issue is performance of the server in a classroom or school. One metric is the number of simultaneous connections the device can support (a classroom of 40-60 is not uncommon). Response time to requests to the server can be limited by the size of memory, the speed of access to the sd card, or the processor speed. I would be very interested in the methodology you propose since that process would apply equally to the schoolserver.

One issue is to characterize the workload - how often does a user request a transaction from the server, what is the time between requests (when the user is reading the response to the previous request), how much processing is required for a request (e.g. a text search), how much information is required to satisfy a request (e.g. size of file download). So far as I know no one has attempted this characterization for a classroom. This load could be different for Sugarizer than for Sugar, but the effort would be valuable in any case)

- **Tim Moody** - tim@timmoody.com - <http://internet-in-a-box.org/>

Speaking of reliability, we have experienced a number of occasions when mongodb was corrupt on the rpi, perhaps through disorderly shutdown. Perhaps you can sort that out. I would also recommend focusing on the rpi install before turning to Heroku or AWS as most users are without an internet connection. All I'm suggesting is that you start with Sugarizer School Box item 1 and then proceed to item 2 when it is complete.

What will you do if you get stuck on your project and your mentor isn't around?

If I get stuck, I will first try my best to find a solution myself (Web Search, StackOverflow). I can ask for help on #sugar or from some friends and ex-colleagues who are Open Source enthusiasts.

How do you propose you will be keeping the community informed of your progress and any problems or questions you might have over the course of the project?

In the Community Bonding period, I will create a blog for myself and I will share my weekly experiences, progress and plans for the next week through it.

During the coding period, I'll attend the Development Team Meetings and will also be available on IRC (#sugar) during my working hours.

Miscellaneous

Setting up a Development Environment

I wasn't sure if this project falls under the Sugar-Desktop projects or the Sugar Web projects so for a relevant screening task, you can check out a Sugarizer deployment I've done at <http://rishabh.xyz:8080/>

I've deployed Sugarizer on my Raspberry Pi 3 and a VPS with Docker and without it. It works well and I haven't run into any major issues.

Tell us something about yourself that will make us like you more.

I believe I'm a good candidate for this project because I'm an avid distro-hopper. The process of merely installing new Linux distributions and setting them up the way I like takes out a large chunk of my time. So, the Sugarizer School Box isn't just some project I'd like to do for GSoC, it's something I'll actually enjoy building!

I come from a family of musicians so I've turned into an amateur singer and guitar player as I've grown up. I also know my way around recording instruments and vocals. If the Sugarizer School Box project didn't exist, I would have picked a Music Blocks project for sure. Maybe I can work on it sometime in the future!

Describe a great learning experience you had as a child.

Well, the most significant learning experience I can recollect is when I learned how to "learn". This is something entirely personal as different individuals require different learning methods. I don't remember exactly when or how I found out but I finally realized that I can learn only by "doing" and not by reading or writing about concepts.

Now, I just use that concept to try learning whatever I need to. It's all about getting your hands dirty and putting in the work.