

Biweekly Status Report 7 (11/23/19 - 12/06/19)

SE 492: Team sddec19-24

DevOps Framework for IOT

Client / Faculty Advisor: Lotfi Ben Othmane

Team Members:

Yussef Saleh	<i>Team Leader</i>
Matt Bechtel	<i>Chief Designer / Engineer</i>
Brett Wilhelm	<i>UI Lead / Engineer</i>
Chakib Ahlouche	<i>Assistant Programmer / Engineer</i>
Ahmed Sobi	<i>Assistant Programmer / Engineer</i>

Git repositories for extensions mentioned below:

<https://git.ece.iastate.edu/sddec19-24/iot-deployment-manager/>

<https://git.ece.iastate.edu/sddec19-24/iot-deployment-service/>

o Bi-Weekly Summary

- Over these two weeks we worked on the solidification of the iot deployment API, working to provide endpoints that allow all necessary functionality to perform and manage deployments. To the deployment service we added deployment history functionality by storing deployment records in our database. Deployment records which also store information about who made the deployment, when the deployment started, and how long it lasted. Some more work is to be done here as there is still some missing functionality, such as information about what commit was run and the logs from that deployment, but that will come soon. Beyond work to finish up the deployment API, there was work done to surface its features, such as UI development and Swagger documentation. Also, work has been done on the packaging of our system via docker, which will allow users to run our code in the environment of our choosing, allowing users to have a smoother and more enjoyable experience with our system.

o Past Week(s) Accomplishments

- DB schema added for the storage of deployment information and repository registration.
- Authentication added to the rest of the routes, slightly refactored this time to allow for simply placing the authentication method in the router as the first function argument.
- UI page added for surfacing of deployment information to the user.
- Completely refactor the deployment API to make it more user friendly and flexible. This was a large effort as the existing API had been fleshed out fairly far. Now the deployment API is set in stone and can be documented/fully deployed.

o Pending Issues

- Collect logging information on running software started by the iot deployment manager. Send that data to the hammerio firebase for retrieval.
- Find a way to collect statistics from Skadi
- Finalize deployment DB schema, store information about deployment (commit hash)
- Finish IoT deployment UI
- Finalize Deployment API

● Individual Contributions

<u>Name</u>	<u>Individual Contributions</u>	<u>Weekly Hours</u>	<u>Total Hours</u>
Yussef	Worked on Docker Implementation and Fixing some errors on YML files. Created and worked on the first draft of the poster.	13	86
Matt	<p>I performed a major refactor to the deployment service and manager, involving a redesign of the available endpoints. Now, the concept of a deployment is a function of two things, a repository and a device. To perform a deployment, an authenticated user must now make a request to create a repository record (with necessary information to clone and start the code in the repository) and a device record first before performing a deployment. Once a repository and device is added (by ip address), a user may create a deployment by hitting the deployment endpoint with a repository id and a list of device ids. This will then search the db for the correct records (repository, device), making sure that the user who made the deployment request owns the devices that they requested deployment to. The service will then send a request to all the devices specified (which should each have a running instance of the deployment manager), performing the requested deployment.</p> <p>Added some functionality around deployment history and the status of any given deployment. Now, when a deployment ends (whether purposeful or unexpected) the given deployment manager will send a notification to the service that a given deployment has ended, this service will then update the deployment record with an end time.</p>	35	150

	The API in general is set in stone now, both services just need to be documented and deployed. Created the entire poster for the final presentation.		
Brett	Time spent working on UI. This included table generation, POSTing and buttons. As always, this was accompanied by significant React information research.	12	86
Chakib	Documentation	1	81
Sobi	Continued working with logs and cloud base infrastructure. Took time to work in the project poster and final report.	10	44

o **Plans for Next Week**

- Presentation Preparation

o **Summary of Advisory Meeting**

- Examined what our best options were for our upcoming presentation