Mid-Project Review

Team Kitsui

Miguel Saavedra | 13826904
Adam Campbell | 1311607
John Cave | 1324776
Christopher Treadgold | 1399164

Executive Summary

We are Team Kitsui (team 18) and we have been contracted by the company Bulletproof to create a serverless content management system (CMS) using Amazon Web Services (AWS). A CMS is: "...a software tool that allows you to create, edit, and publish content" (Christensson, 2013). In the case of the CMS we are designing the content is blog posts, menu bars, data entry forms and other types of content that the CMS will use to generate and update websites. An example of a CMS similar to what we are designing is WordPress, an extremely popular tool for creating blogs.

Bulletproof considers this a worthwhile project as a CMS using AWS likely costs less for running small to medium sized websites compared to running a dedicated server. The particular AWS service that allows for the development of such a CMS is AWS Lambda. AWS Lambda is used to run code within the cloud in response to events, costing a very small amount per invocation in addition to a similarly small cost for processing time. AWS (in particular Lambda) is extremely new and being expanded upon as well as improved frequently, so we hope using its cutting-edge technology in a new and novel way will attract attention to this project, giving us an advantage over other newly emerging CMS tools.

We aim to produce a minimum viable product (MVP) as designated by Bulletproof, some of the features it will have in order to be considered as such are:
- A simple setup process using a bash script.
- An intuitive web interface for administrative tasks and content creation.
- The ability to easily create and manage blog posts.
- Built in support for user accounts.

Project Description

Our CMS will be web based, its functions accessed through an online administrative dashboard that is created upon the CMS being set up. To achieve this we will be using AWS Lambda for the CMS's processing tasks.

AWS Lambda is a service that runs code on the fly within the cloud, costing a small amount per invocation in addition to a similarly small cost for processing time. Due to only having to pay for what is used, Bulletproof believes that smaller websites will have a lower cost of operation using AWS Lambda compared to running a dedicated server, or even a cloud based server like AWS Elastic Cloud Compute (EC2).

However, AWS Lambda is not the only AWS tool we will be using. We are also using a number of other AWS services, one of which is DynamoDB, a NoSQL database in the cloud which we will use for storing blog posts, user data, etc.. Another is the AWS API Gateway which will be used to enable web pages to interact with AWS through javascript.

Project Overview

Objectives

Our long-term objective remains what it was in the project proposal, which is to create a Lambda-based CMS for Bulletproof that meets their definition of MVP.

Our objective in the short term is to create a simple prototype of our CMS to act as a proof of concept for posting a simple blog, which we will use as a base for developing the fully featured CMS. The prototype will have the ability to generate the necessary AWS resources, data, and functions to let an administrator create a simple blog.

As a part of the prior objective we are, in our current sprint, focusing on creating as secure a system as we can starting right away with the prototype stage. We believe doing so will increase the security of the end product in addition to reducing the effort on our part. We prefer this approach to having to go back through what we have created to add security features. In order to achieve this security goal, we are currently looking into various authentication and encryption options for data transfer and user login/sessions.

Scope

The main features within our scope for the prototype are:
- Allowing an administrator to log in to and log out from the CMS.
- Secure administrative login to the CMS.
- Allowing an administrator to create and delete blog posts.
- Allowing an administrator to add users to the CMS.
- Having a script that generates all the necessary AWS resources to set up the CMS.

User stories that describe each of these features are in the sprint 2 backlog in appendix A. User stories that describe the features of the completed CMS are in the product backlog in appendix B. Scope change is described in the *Relating to Project Proposal* section under *Differences and Variation*.

Approach

We continue to use Scrum as our project management methodology. Our sprint length remains 2 weeks and we aim to have a sprint retrospective and review at the end of each sprint. We have been tracking our sprints with JIRA since we received access to it several weeks ago, as well as manually tracking with a burndown chart in a google document. Before receiving JIRA access we were using google sheets, the aforementioned manual burndown chart, and Trello.

The reason we are still manually creating burndown charts rather than using JIRA's built in reports is due to a problem with our JIRA where we have not been given the necessary permissions to remove the default tutorial user stories and versions that come with a new JIRA project.

Unfortunately though we planned to have meetings with our client every week, they are often difficult to get in contact with or unable to have a meeting with us, at the moment this is caused by our Bulletproof contact being in Fiji. Due to this, we have had to take the initiative in solving problems and making decisions that we would typically ask our client for advice on.

Changes to our approach:
- Due to not getting access to JIRA we instead used Trello for project management. However, we have since received JIRA access and have moved our project management to it.
- Due to not receiving a team AWS account from Bulletproof we are instead using our own AWS accounts until access is received. This means that for the time being we are paying any costs we incur using AWS services.
- Where we originally planned to do our daily standups using Slack, we have instead opted to have them in a group Google document. This also gives makes it easier to search through previous standups

Major Milestones

The one major milestone we have completed so far is our training in AWS and AWS Lambda on Linux Academy. Linux Academy is a website that offers online courses in a number of subjects for a monthly fee that has been covered by Bulletproof.

As mentioned in our objectives we are currently moving toward the major milestone of having a simple CMS with limited functionality to act as a proof of concept as well as a base from which to build the fully featured CMS.

Planned End Products

The items we aim to have delivered by the end of the project are:
- A CMS that fulfils the user stories in the product backlog.
- A public Wiki describing how to setup and use the CMS as well as detailing its various components for those who wish to assist with its development as it is open-source.
- A sample website that utilises all of the features we have implemented into the CMS.

Relating to Project Proposal

Since the project proposal phase, our vision of the finished product has become much clearer. Each member completing an online course in AWS has helped greatly in causing this clarity to come about and has allowed us to begin work on a prototype CMS.

Differences and Variation

There are minimal differences between our plans now and what we stated in our project proposal. The differences are kept in a scope change document and are:
- We added a Lambda deep dive course on Linux Academy to our tasks as our client decided we needed more training in Lambda to be able to complete the project.
- No longer using amazon cognito as we realised it not suitable for our needs. This has caused us to have to create our own authentication system.

Project Status

Work Complete

We are behind where we had planned to be at this point, we have determined and have determined the reason to the overly optimistic planning rather than a lack of work on the part of the team. We have changed our plans accordingly.

Items we have produced:
- A product backlog (In JIRA).
- Sprint 1 & 2 user stories (in JIRA).
- A product Wiki in its early stages.
- A preliminary database structure.
- A number of scripts for automating setup of the CMS that generates:
  - An S3 bucket with HTML files
  - A partially implemented API Gateway for sending data to the server (middleware and backend)
  - A default Admin user
  - Several Lambda functions (log in, log out, register user)
- Architecture diagrams for:
  - Security
  - Visitor access
  - Login
  - API calls
- A project website with information about the project.

- A GitHub repo for storing our scripts and code.

In Progress

Our work at the moment is focused on completing the prototype CMS that will function as the base for the system as we develop it. A major problem being worked on is designing a system where users can register for user accounts and adjusting the website based on whether a user is logged in or not and what permissions that user has. Another problem that is still currently being solved is sending data from the administrative dashboard in a web page to the API gateway which will send data to AWS Lambda for processing.

Future Work

We aim to continue completing items from the product backlog which includes adding a number of features to the CMS as well as increasing usability and improving the appearance of the whole system (UX design).

Issues

1. We are roughly 3 days behind where we projected we would be. We have identified the reason for the delay as an overestimation of what we could complete in the first sprint as well as an underestimation of the amount of study required to start work, we have adjusted our plans to compensate.

2. The availability of our client contact is sporadic, with responses to emails and messages taking days to weeks to be received.

3. We have not received access to an AWS account with credit for creation of the CMS.

4. Poor documentation on AWS.

Project Team Recommendations

Testing in each sprint

We have implemented testing within our sprints using unit and acceptance testing. This is visually outlined within our updated Gantt chart to show that we are testings throughout the development sprints.

Define major milestone

We have defined our milestones within our updated Gantt chart which is made up of sprint goals. An overview of our milestones is outlined in a Milestone document which breaks down what sprint goals make up a milestone.

Arrange client signoffs upon completion

We have not been able to arrange a formal sign offs for our deliverables as communicating communicating communicating communication with the client has been challenging as previously stated within the issues section of the report. We have however created an end of sprint report by direction of the client for review as he was out of the country for the duration of the sprint. We are planning to use this type of document to report on the progress of the product in 2 sprint iterations recommended by our supervisor. This then entails that we shall email the client with our sign off document and summary of completed goals and tasks at the end of our sprint 2 (30/6/16).

Clearly defined roles for each team member

After our initial Proposal presentation with the AUT board, we have produced a Role Register which describes the members of the team, roles assigned, the type of role associated (technical, administrative, leadership etc), and the reason for assigning specific roles to specific members.

End user involvement plan (For feedback and testing of the MVP) Suggestion: Perhaps use students for the testing of your MVP

At this point in time, we have not yet produced an End-User Involvement plan as we are still in the process of producing the concept for our Client, Bulletproof (formerly known as CloudHouse)

It is our intention to create said End User involvement plan once the concept has been successfully completed, as we will then have an interface which Users can then test, and provide feedback.

We will be receiving End User involvement via Students from the Year 3 Web Development paper, as these Students will be able to provide both general usability feedback, as well as technical feedback.

Creating an Issue log and using an Issue Tracking System like JIRA

We are now using JIRA for code issue tracking. This is integrated fully into our sprint and product backlogs and allows us to assign bug fixing tasks alongside product backlog items to team members for sprints. We are using a Google spreadsheet for tracking non-code related issues with a colour-coded key describing the likely impact on the current sprint's deadline.

Use Issue Tracking System, task tracking and progress reporting

Jira provides these facilities automatically, allowing these items to be generated as graphs without manual intervention by the team members. Prior to obtaining access to JIRA, a team member generated these documents manually each week and a spreadsheet was used for tracking.

Individual Work and Learning

Miguel Saavedra

Total Hours: 132.5

Work Done
Lambda Functions: *9 Hrs*
- Logout Function
- Contributed to Login Function

Setup scripts: *17 Hrs*
- Dynamo User tables
- Dynamo Token tables
- Index
- Admin User

HTML CSS JS: *6 Hrs*
- Login Page
- Implemented Bootstrap Library
- Basic based XHR form

Documentation: *35 Hrs*
- Minutes for meetings
- Review over document

AWS Training : *22.5 Hrs*
- AWS Fundamentals
- Lambda Fundamentals

Meetings : *33.5 Hrs*
Lectures : *9.5 Hrs*

Note : Hours include research to get to result

Learning
Dealing with scope change
During the project, we had to deal with many changes to the scope by finding out that the initially planned AWS modules we were planning to use did not fill the needs of one of the functions. This made us think on our feet to dive into more research and come up with a solution then confirm this with a knowledgeable source from Bulletproof before starting further development.

Adapting to lack of client presence and guidance
Learning that the client has a large presence within the project which I have not experienced before. Lack of communication between the team and the client during critical sections without guidance can lead the team astray and force the product owner hat upon the team. The sprint backlog and stories that are chosen for the sprint would evidently affect the result at the end of project. The right selection of user stories also comes back on the depth of understanding the team has of the deliverable.

Time and Priority Management
It is of the utmost importance to manage one's time with a busy schedule which became extremely evident as the project passed its first few milestones. Learning techniques from Jim about capturing reminders on the fly so you do not forget to do the task. Also having personal planning sessions helps with keeping focused on project tasks and organize other events so there is less chance of compromise with work done. Keeping organized with these techniques is especially important to make sure you are delivering the promised work on your end and not bringing a burden upon the team.

Adam Campbell

Total Hours: 121

Work Done
Meetings: 35 hours
- Client meetings
- Supervisor meetings
- Team meetings
- R&D Presentations

Code Base: 23 hours
- Project folder structure
- Project launch automation (BASH scripting)
- API Gateway automation
- MainController produced
- Registration function produced
- Password hashing/salting (encryption) produced
- DynamoDB table Universally Unique Identifiers produced

Documentation: 17 hours
- API Templating (API Reference)
- Status Report Revision and Proofreading
- AWS User Interaction diagrams
- Stand-ups
- Project Proposal production/proofreading/revision
- Proposal Presentation
- Scope change rationale

Training (Linux Academy): 35 hours
- AWS Certified Developer - Associate level
- Lambda Deep Dive

Research: 11 hours
- CloudFormation templating
- API Gateway creation and required modules, implementation
- Lambda deployment package implementation
- BASH scripting

Learning: Through this first half of the Research and Development project, I have come to learn about the **importance of constant communication between all parties involved**. We as a team have kept a frequent communication channel open between all members of our team and have been in frequent communication with our Supervisor - Waqar. One difficulty we have been facing at times is difficulty with reaching our Product Owner via the agreed upon communication platforms (email, Slack, SMS), however, this is slowly being resolved. Without the constant flow of communication and project updates, parties can quickly become disjoint with project progress and flow.

Practices, tools, and flexibility of methodology choice have also been important to us thus far. We have found the structure offered by SCRUM has been excellent for planning and tracking. Having 2 week long sprints ensures that we can see our client frequently and delivery often, and also receive direction and advice consistently. This has somewhat been challenging however as we have had difficulty with the technology being used (Amazon Web Services), and the Product Owner being absent at the end of Sprint 1 and beginning of Sprint 2, however, we are a strong team and are able to adapt to change. **Agile!**

Teamwork has also been a critical skill for our Research and Development project. The amount of work involved with this project is definitely not something that can be achieved alone, therefore having a strong team is a must to achieve project success. We as a team having been in frequent communication have been able to ensure that each member is aware of who is producing what at any point in time.

John Cave

Total hours: 108

Work Done
Lambda Functions
- Log in user.
    - Password Decryption.
    - Token Assignment.
- API Gateway configurations.

Code Base
- Designed security architecture.
- Proposed secondary product architecture.

Documentation
- Created public project wiki.
- Created public home page.
- Project Gantt Chart.

- Team logo.
- Team timetable.
- Project Plan - Skills Section.
- Project presentation - tools section.

Research
- Password hashing on Node.
- API Gateway cookie setting.
- API Gateway form submission to Lambda.

Linux Academy Training
- AWS Certified Developer.
- Lambda Deep Dive (only 1/4th completed).

Learning

Over the course of this project, it has become more apparent to me the many ways in which a task can be accomplished. My programming assignments to date have all been completed individually, which allows me to quickly narrow possible methods down to one. Working with the rest of the team and working out the best method of completing a programming task has been a useful experience, as all members have different ideas about how to complete nearly every task.

Being involved in this project group has allowed the team and I to get hands-on experience and training in using industry standard cloud technologies, something not present in my university course so far. This has been useful and interesting in many ways and I look forward to working with the cloud more in future.

As a Networks and Security major, I have never been introduced to development methodologies during the course of my studies until I suddenly had to learn all of Scrum over the course of a week to prepare for the Project Proposal presentation. This allowed me to learn more about the software industry's inner workings and open my mind about this sector of the ICT industry.

I have also found the situation we have been having with trying to get in touch with our client interesting. People in the industry often talk to me about the difficulties that can be faced with regards to communicating effectively and in terms they can understand with their clients, and it has been interesting to observe this first hand.

Christopher Threadgold

Total Hours: 90

Work Done
Documentation
- Wrote and edited this status report.
- Wrote a section in the project proposal.
- Edited the project proposal.

- Took minutes several times for meetings.

Research
- Around 80% through AWS developer certification course.
- Still to complete the Lambda deep dive.

Learning

I have always struggled working in teams. Communication is frequently poor. There are team members who do no work. And at the end of the day the product produced is nothing to be proud of. All in all I considered teams to be a waste of time that tended to result in disaster. That being said, my experience with this team has certainly changed my outlook there. When collaboration is done right (not that we have done perfectly in that regard) it seems that projects become a whole lot more successful. And beyond that, on an individual level, more enjoyable.

For me so far, this paper has very much been an exercise in extensive documentation. I used to dislike going through documents over and over, looking for every little error and opportunity to restructure a sentence to enhance its meaning. But as I've been doing it more and more over my university years, and in particular in this (and related) subjects, it becomes almost meditative. Almost relaxing.

I expect this experience will be extremely valuable not just in any future jobs I find myself in but also for life in general. Having the opportunity to work with a real company in a team aiming to achieve a real objective is a fantastic thing.

References

Christensson, P. (2013, March 28). *CMS Definition*. Retrieved 2016, May 22, from
      http://techterms.com

Appendix

Appendix 1 - Sprint 2 backlog

Appendix 2 - Product Backlog (The reasons section will be added to each soon)

🟩 ↑	ALC-24	As a visitor, I'd like to view blog posts.		Visitor Experience	13
🟩 ↑	ALC-30	As a visitor, I'd like to view a page listing.		Visitor Experience	⚪
🟩 ↑	ALC-32	As a visitor, I'd like to make comments on blog posts.		Visitor Experience	⚪
🟩 ↑	ALC-34	As a visitor, I'd like to view the comments on blog posts.		Visitor Experience	⚪
🟩 ↑	ALC-35	As a visitor, I'd like to search the site.		Visitor Experience	⚪
🟩 ↑	ALC-25	As an administrator, I'd like to view blog posts.		Administrative Dash...	13
🟩 ↑	ALC-26	As an administrator, I'd like to create blog posts.		Administrative Dash...	13
🟩 ↑	ALC-29	As an administrator, I'd like to upload images.		Administrative Dash...	5
🟩 ↑	ALC-61	As an administrator, I'd like to create roles.		Administrative Dash...	15
🟩 ↑	ALC-49	As an administrator, I'd like to add permissions to site components.		Administrative Dash...	15
🟩 ↑	ALC-52	As an administrator, I'd like to add user groups.		Administrative Dash...	15
🟩 ↑	ALC-58	As an administrator, I'd like to assign permissions to roles.		Administrative Dash...	20
🟩 ↑	ALC-55	As an administrator, I'd like to assign permissions to groups.		Administrative Dash...	20
🟩 ↑	ALC-46	As an administrator, I'd like to assign roles to users.		Administrative Dash...	8
🟩 ↑	ALC-43	As an administrator, I'd like to add users.		Administrative Dash...	20
🟩 ↑	ALC-27	As an administrator, I'd like to remove blog posts.		Administrative Dash...	⚪
🟩 ↑	ALC-28	As an administrator, I'd like to insert images into blog posts.		Administrative Dash...	⚪
🟩 ↑	ALC-31	As an administrator, I'd like to view a page listing.		Administrative Dash...	⚪
🟩 ↑	ALC-33	As an administrator, I'd like to remove comments.		Administrative Dash...	⚪
🟩 ↑	ALC-36	As an administrator, I'd like to create pages.		Administrative Dash...	⚪
🟩 ↑	ALC-37	As an administrator, I'd like to remove pages.		Administrative Dash...	⚪
🟩 ↑	ALC-38	As an administrator, I'd like to edit pages.		Administrative Dash...	⚪
🟩 ↑	ALC-39	As an administrator, I'd like to create a navigation bar.		Administrative Dash...	⚪
🟩 ↑	ALC-40	As an administrator, I'd like to edit a navigation bar.		Administrative Dash...	⚪
🟩 ↑	ALC-41	As an administrator, I'd like to remove a navigation bar.		Administrative Dash...	⚪
🟩 ↑	ALC-42	As an administrator, I'd like to add the facebook social plugin.		Administrative Dash...	⚪
🟩 ↑	ALC-44	As an administrator, I'd like to remove users.		Administrative Dash...	⚪
🟩 ↑	ALC-45	As an administrator, I'd like to edit users.		Administrative Dash...	⚪
🟩 ↑	ALC-62	As an administrator, I'd like to remove roles.		Administrative Dash...	⚪
🟩 ↑	ALC-63	As an administrator, I'd like to edit roles.		Administrative Dash...	⚪
🟩 ↑	ALC-47	As an administrator, I'd like to remove roles from users.		Administrative Dash...	⚪
🟩 ↑	ALC-48	As an administrator, I'd like to edit user roles.		Administrative Dash...	⚪
🟩 ↑	ALC-50	As an administrator, I'd like to remove permissions from site components.		Administrative Dash...	⚪
🟩 ↑	ALC-51	As an administrator, I'd like to edit the permissions of site components.		Administrative Dash...	⚪
🟩 ↑	ALC-53	As an administrator, I'd like to remove user groups.		Administrative Dash...	⚪
🟩 ↑	ALC-54	As an administrator, I'd like to edit user groups.		Administrative Dash...	⚪
🟩 ↑	ALC-64	As an administrator, I'd like to add users to groups.		Administrative Dash...	⚪
🟩 ↑	ALC-98	As a developer, I need to edit a set up bash script for adding two html files so that I can automate the uploading of those 2 objects			⚪
🟩 ↑	ALC-65	As an administrator, I'd like to remove users from groups.		Administrative Dash...	⚪
🟩 ↑	ALC-56	As an administrator, I'd like to remove permissions from groups.		Administrative Dash...	⚪
🟩 ↑	ALC-57	As an administrator, I'd like to edit group permissions.		Administrative Dash...	⚪
🟩 ↑	ALC-59	As an administrator, I'd like to remove permissions from roles.		Administrative Dash...	⚪
🟩 ↑	ALC-60	As an administrator, I'd like to edit role permissions.		Administrative Dash...	⚪

Appendix

The specific requirements for each section of the project proposal and this status report could have been clearer. For instance, this very section of project team recommendations had no description. Until asking we didn't know if it was about what we recommend to ourselves what we should do in the future or a number or other possibilities or its actual purpose as a commentary on the paper itself.

A submission link on AUTonline would have been considerably more convenient as we could guarantee the report's having been received as well as not having to rely on the (in our experience) unreliable university email systems that a number of students frequently can't access.

Have a course book or wiki that covers everything in the lectures, whether it be physical or digital so that new teams can get advice on project problems and approaches with ease and earlier in the project. This is opposed to having the information come later within the project through lectures.

Have a standardized logging document for hours (preferably a spread sheet).