AWS CMS Sprint 1 Report

16/5/16

What were our goals

- Implement Concept model
    - View Blog posts (admin/public)
    - Add Blog Posts (admin)
    - Upload images (admin)
    - Add Users (admin)
    - Add Roles (admin)
    - Add permissions (admin)
    - Add groups (admin)
    - Assign permissions to roles (admin)
    - Assign permissions to groups (admin)
    - Assign user to a role (admin)

What we have done

We have created initial setup files through Bash scripts which runs through the AWS CLI(Command line interface). This is to set up needed modules for this project on the fly.

- Setup
    - Setup sh scripts Role
    - Setup sh scripts Group
    - Setup sh scripts User
    - Setup sh scripts Lambda Function
    - Setup sh scripts S3 Bucket
    - Extract needed information from Deep Dive Lambda course

- Lambda
    - Basic lambda function which pulls an object from s3 and reads the contents
    - Basic lambda function which puts a database record into dynamo through a form on a browser

- Linux academy training
    - Completed the bulk of AWS training we felt required for the project.

What we have not completed

- View Blog posts (admin/public)
- Add Blog Posts (admin)
- Upload images (admin)
- Add Users (admin)
- Add Roles (admin)
- Add permissions (admin)
- Add groups (admin)
- Assign permissions to roles (admin)
- Assign permissions to groups (admin)
- Assign user to a role (admin)

Why we have not these tasks been completed

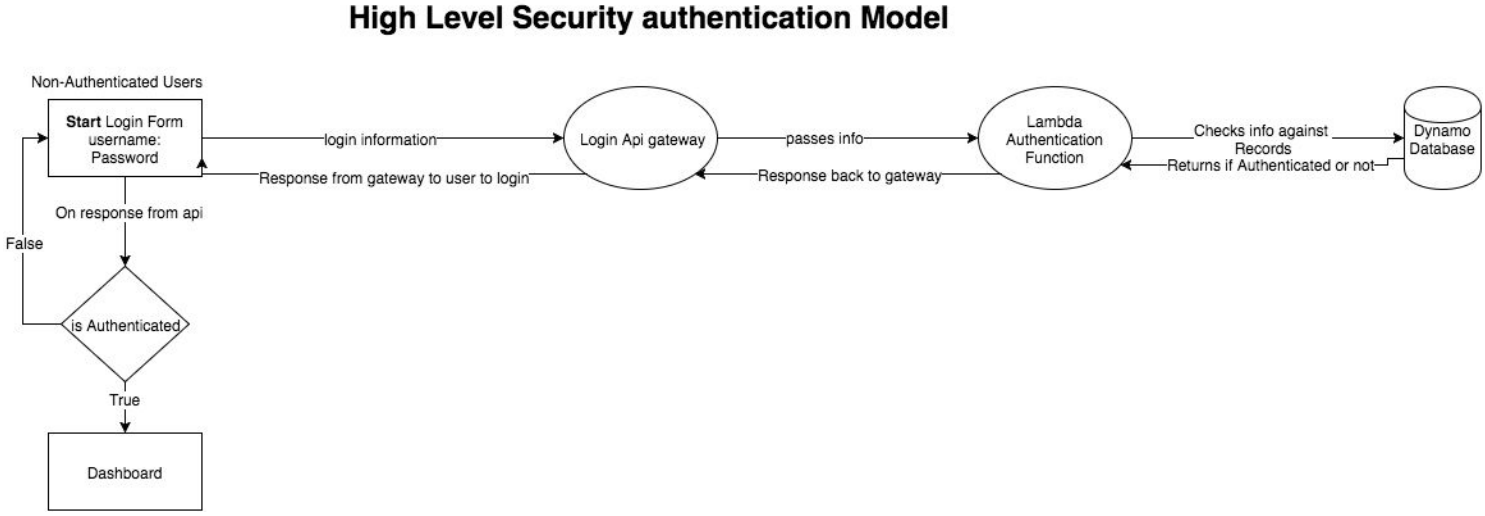
- The tasks we have been allocated have been partially implemented through shell script, however we not been able to fully complete them to the authentication layer being difficult to implement.
- The tasks required more effort than anticipated and the core training course that was provided did not give enough information to successfully complete these tasks.
- Our sprint backlog did not account for research time required for Sprint 1 task implementation.

Impedances

- Authentication and authorization (Amazon cognito limitations)
    - Cognito is too simplistic for our requirements
        - User, Role, Group management (edit, add, remove etc.)
    - Users are considered either authenticated or not authenticated
    - Authenticated users only get one role and they are not distinguished by any credentials (username, email address etc.) - only considered authenticated.
    - Users, Roles, Groups within AWS Console (in browser) are only for using the AWS Console, not for any interfacing with our AWS CMS, therefore a separate user/role/group process required (see below)

Possible solutions to impedances

- Since Cognito does not distinguish users and therefore different roles/groups, we have considered our own authentication model. However we do not know if this is the perfect implementation and we would appreciate input from your technical team. We will be sending a slack message to them about this proposed model.



Sprint 2 Goals

- Security/Authorization/Authentication
    - (we believe highest priority right now - security first!)
- If security achieved prior to the end of sprint, additional tasks will be selected from the previous sprint in accordance to remaining time available.
- We will spend the first week of the sprint on the security model seen above, and will keep you up to date regarding the progress.
- John will be investigating the CMS he found a few weeks back.