# Assignment #07

## Due Date

| Section | Due Date | Grading Deadline |
| --- | --- | --- |
| Saturday & Online | 09:00pm on 08/02/2019 | 09:00pm on 08/09/2019 |

## Getting Help

**Info**

Ask all your questions on Piazza. Assign **assignment7** tag to your posts.

## Assignment Weightage

Individual Assignment Weightage on Course Grade of this assignment is **10%**.

## Objectives

### IAM Users, Roles & Policies Updates

- Add IAM roles & policies needed to meet the assignment objectives to the appropriate CloudFormation stack.

## Amazon Simple Email Service (SES) Configuration

> **Info**
>
> Following steps are done **manually** and only once for each domain.

- Verify Domain in Amazon SES [http://docs.aws.amazon.com/ses/latest/DeveloperGuide/verify-domains.html]
- Authenticate Email with DKIM in Amazon SES [http://docs.aws.amazon.com/ses/latest/DeveloperGuide/dkim.html]
- Move Out of the Amazon SES Sandbox by Requesting Production Access [http://docs.aws.amazon.com/ses/latest/DeveloperGuide/request-production-access.html]

> **Warning**
>
> When requesting production access for email, you must submit a genuine case or AWS will deny your request. AWS support can take 3-5 days to respond to your request so you only have one attempt at getting your request approved to meet the assignment deadline. *There is nothing I or the TAs can do in getting your AWS request approved.* I do not provide template or ticket content. Writing up the use case and your needs is one of the objectives of this assignment.

## Amazon Simple Notification Service Configuration

Add Amazon Simple Notification Service (Amazon SNS) topic creation to your CloudFormation `application` stack. SNS CloudFormation documentation can be found here [http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-sns-topic.html].

## Setup GitHub Repository for Lambda Functions

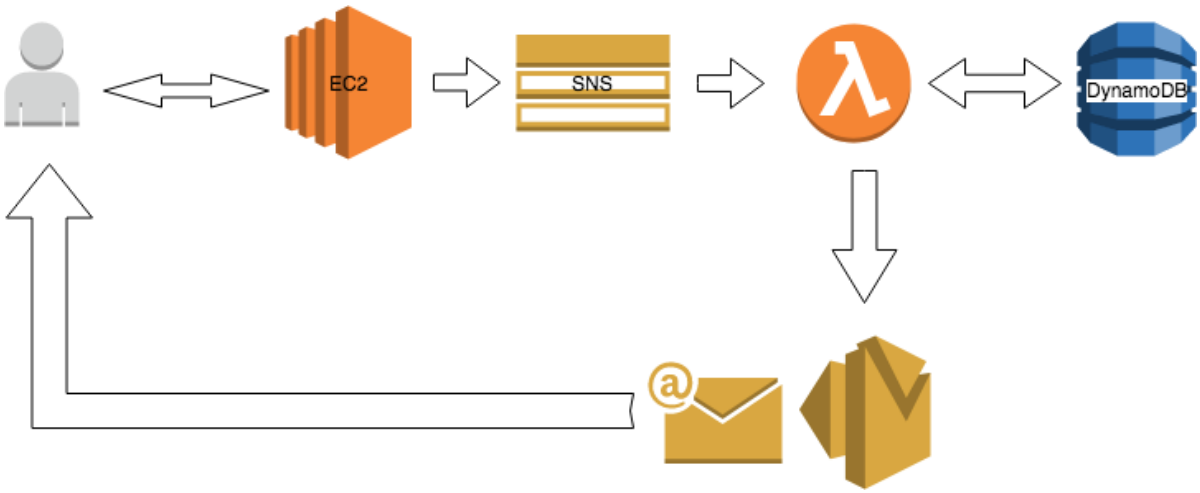
1. Create a GitHub repository for assignments [https://help.GitHub.com/articles/create-a-repo/]. This must be a private

repository that only your team and TAs can access. Make sure to create empty repository. Just like we learned in the lab, one member should create the repo and other's should fork this repo.

2. GitHub repository name must be **csye6225-su19-lambda**.
3. Add all TAs to your GitHub repository as collaborators. TAs emails and GitHub IDs can be found on home page.
4. Add a **README.md** in your repository with team member information.

## Email Service Using AWS Lambda Function

> **About DynamoDB TTL**
>
> DynamoDB typically deletes expired items within 48 hours of expiration. The exact duration within which an item truly gets deleted after expiration is specific to the nature of the workload and the size of the table. Items that have expired and not been deleted will still show up in reads, queries, and scans.
>
> http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/howitworks-ttl.html [http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/howitworks-ttl.html]




### Authentication and Access Control for AWS Lambda

- Add Authentication and Access Control for AWS Lambda [https://docs.aws.amazon.com/lambda/latest/dg/lambda-auth-and-access-

control.html] to the appropriate CloudFormation stack.

**Implement Password Reset API for Forgotten Password (Web Application)**

We must provide a way for user to request link which will allow them to reset their password. For this assignment, our focus is to only generate such link and email it to the user. We will not implement actual feature to allow users to reset their password.

> **Example**
>
> Example of password reset link: `http://example.com/reset?email=user@somedomain.com&token=4e163b8b-889a-4ce7-a3f7-61041e323c23`

Swagger Docs [https://app.swaggerhub.com/apis-docs/csye6225/csye6225-summer2019/3.0.0#/public/post_reset]

1. As a user, I want to be able to request reset password link by calling `/reset` API endpoint. As I have forgotten my password, I cannot make authenticated API calls to this endpoint.
2. As a user, I expect the web application to send a message on `password_reset` SNS topic for the email service function to actually send email and track the active tokens in DynamoDB.

**Implement Lambda Function**

1. Lambda function will be invoked by the SNS notification. Lambda function is responsible for sending email to the user.
2. As a user, I should be able to only have 1 password reset token active in database (DynamoDB) at a time.
3. As a user, I expect the password reset token (value) and username (key) be stored in DynamoDB with TTL of `15` minutes.
4. As a user, I expect the password reset token to expire after `15` minutes if it is not used by then.
5. As a user, I expect the to receive the password reset link via email.

6. As a user, if I make multiple requests to reset password when there is a active token in the database, I should only receive `1` email.

### Implement CI/CD Pipeline for Lambda Function

Using the same principles as in assignment #5, implement continuous deployment for Lambda functions. Every commit to the `csye6225-su19-lambda` repository should trigger a CircleCI build and deployment of your updates function to AWS Lambda.

## Documentation

### Amazon Simple Email Service

- Getting Started Receiving Email with Amazon SES [https://docs.aws.amazon.com/ses/latest/DeveloperGuide/receiving-email-getting-started.html]
- Set Up an Email-Receiving Pipeline [https://aws.amazon.com/getting-started/projects/setup-email-receiving-pipeline/]

### Amazon Simple Notification & AWS Lambda

- Using AWS Lambda with Amazon SNS [https://docs.aws.amazon.com/lambda/latest/dg/with-sns-example.html]
- Invoking Lambda functions using Amazon SNS notifications [https://docs.aws.amazon.com/sns/latest/dg/sns-lambda.html]
- Creating a .zip Deployment Package (Java) [https://docs.aws.amazon.com/lambda/latest/dg/create-deployment-pkg-zip-java.html]
- Libraries, samples and tools to help Go developers develop AWS Lambda functions [https://github.com/aws/aws-lambda-go]

- Lambda Function Examples (receiving-email-action-lambda-example-functions) [https://docs.aws.amazon.com/ses/latest/DeveloperGuide/receiving-email-action-lambda-example-functions.html]
- Tutorial: Using AWS Lambda with Amazon Simple Notification Service [https://docs.aws.amazon.com/lambda/latest/dg/with-sns-example.html]

## AWS CodeDeploy & AWS Lambda

- CodeDeployLambdaServiceRole [https://northeastern-my.sharepoint.com/:b:/g/personal/tejasparikh_northeastern_edu/EUhhrggMlARFopk2c5pg1X8B7pViOK-OX_svjj9gfbTolw?e=4gOJlu]
- https://docs.aws.amazon.com/codedeploy/latest/userguide/deployment-steps.html#deployment-process-workflow-lambda [https://docs.aws.amazon.com/codedeploy/latest/userguide/deployment-steps.html#deployment-process-workflow-lambda]
- https://docs.aws.amazon.com/codedeploy/latest/userguide/reference-appspec-file-structure-hooks.html#appspec-hooks-lambda [https://docs.aws.amazon.com/codedeploy/latest/userguide/reference-appspec-file-structure-hooks.html#appspec-hooks-lambda]
- https://docs.aws.amazon.com/codedeploy/latest/userguide/deployments-create-console-lambda.html [https://docs.aws.amazon.com/codedeploy/latest/userguide/deployments-create-console-lambda.html]
- https://docs.aws.amazon.com/codedeploy/latest/userguide/reference-appspec-file-example.html#appspec-file-example-lambda [https://docs.aws.amazon.com/codedeploy/latest/userguide/reference-appspec-file-example.html#appspec-file-example-lambda]
- https://docs.aws.amazon.com/codedeploy/latest/userguide/reference-appspec-file-structure-resources.html [https://docs.aws.amazon.com/codedeploy/latest/userguide/reference-appspec-file-structure-resources.html]

- https://docs.aws.amazon.com/codedeploy/latest/userguide/troubleshooting-deployments-lambda.html [https://docs.aws.amazon.com/codedeploy/latest/userguide/troubleshooting-deployments-lambda.html]

## Submission

> **Danger**
>
> Assignment will be considered late if commits are made to `master` and `feature` branch after due date.

1. All work for this assignment must be done on **assignment7** feature branch and merged to master when you are dev complete.
2. All team member's feature and master branches must be in-sync.

## Grading Guidelines

> **Warning**
>
> Following guidelines are for information only. They are subject to change at the discretion of the instructor and TA.

1. Have all your CloudFormation stacks running before your demo slot.
2. Have student kickoff CircleCI build for web application and AWS Lambda function. Build must run from `master` branch. This will ensure that code on the EC2 instance and in Lambda function is latest.
    a. Make sure student starts the build with CircleCI API and not by making commits to their GitHub repository.
3. Verify student's domain has production access for SES.
    a. Check `Verified Emails` for SES and make sure all entries are removed.

4. Verify DKIM is configured for the domain.
5. Verify SNS topic exists and is created by CloudFormation template.
6. Make an API call to `/reset` endpoint with your (TAs) email address.
    a. Verify that you receive email.
    b. Verify email does not end up in SPAM folder.
    c. Check the entry in DynamoDB table.
7. Make 3 additional API call to `/reset` endpoint with your (TAs) email address.
    a. Verify that no emails are sent for the TTL duration.
8. Verify the password reset link in the email.
    a. It should contain student's domain name and not `example.com`.
    b. It should contain your (TA) email address for `email` param and a UUID for `token` parameter.