Team 27

Project Name: WhyIsDrought

Date: 01/11/2020

Team Member: Junjie Huang Zhicong Hu Jinchen Dai Rongkun Ji

Table of Contents

1. Overview

This document mainly supports the daily management and operation of the website ‘WhyIsDrought’, and also provides technical installation guidance of ‘WhyIsDrought’. What’s more, this document also provides methods of backup and restore, as well as the tools and knowledge needed to run the website ‘WhyIsDrought’. This document can help staff within the sponsoring organisation better understand the operational requirements and daily operational details of ‘WhyIsDrought’.

2. WhyIsDrought Introduction

Drought is a well-known hazard in Australia which threatens the country a lot. It is necessary to tell children the knowledge about the drought so that they can have a systematic awareness of drought. In Australia, the economic loss caused by drought every year has reached an inestimable value, even if the government has responded positively. However, if everyone has a clear understanding of drought and fights together that it can better fight against drought. This project aims to construct a website for education.

The current alternatives are too technical for our target to use, which is the reason why the team wants to develop the website. The team is about to develop a user-friendly website for our target users to introduce drought to them in a less technical manner. Moreover, this website will let students understand the drought situation in Australia more scientifically and effectively, and allow them to make possible contributions to future droughts.

3. Backup and Restore

Because the website ‘WhyIsDrought’ uses the Amazon RDS to manage databases. Therefore, the database backup and restore can use the guide in RDS. The following content is part of the user guide in RDS.

When using Amazon RDS to manage the database, it will create and save automatic backups of the database instance. First of all, RDS creates a snapshot of the storage volume of the database instance and backs up the entire database instance instead

of just individual databases. What's more, RDS saves automatic backups of the database instance according to the backup retention period specified by the developer. If necessary, developers can recover the database to any point in the backup retention period.

If the developers want to use automatic backup, they will follow the following two rules:

1. The database instance must be in AVAILABLE state for automatic backup. When the database instance is in a state other than AVAILABLE (for example, STORAGE_FULL), no automatic backup will be occured.
2. When a copy of the same database instance is running in the same region, automatic backup and automatic snapshot will not be performed.

Moreover, if the developers want to restore data quickly. Developers can create a database instance by restoring from the database snapshot. When restoring a database instance, the developer needs to provide the name of the database snapshot to be restored and the name of the newly created database instance after restoration. However, developers cannot restore from a database snapshot to an existing database instance. Because when restoring, a new database instance will be created.

For more detailed information about database restoration and backup, please refer to the RDS user guide:
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_CommonTasks.BackupRestore.html

4. Daily management and future

For the website run stable and well, please check the lists below:

1. Check whether the domain name is expired in the AWS Route53.
   https://console.aws.amazon.com/route53/v2/home#Dashboard

2. Check the security of the website in the AWS amplify if you want to change. https://ap-southeast-2.console.aws.amazon.com/amplify/home?region=ap-southeast-2#/d2f3na0y7kaub6
3. Check the AWS account money to make sure the service can run normally https://console.aws.amazon.com/billing/home?#/account
4. Check the Github code to see if there is any change. https://github.com/Wolphatch/industralExperience
5. Check the AWS RDS and Mysql to see if data has any changes. https://ap-southeast-2.console.aws.amazon.com/rds/home?region=ap-southeast-2#
6. Check the Tableau website to make sure no changes in data visualisation images. https://github.com/Wolphatch/industralExperience/tree/master/dataVisulization/Embeded%20Visulization

For a future change for the website, please consider list suggestions below:

1. If you want to change any pages of the website, please find code in Github and pull first, then change the content you want.
2. If you want to change the database structure, please change it in AWS RDS and Mysql and consider whether the AWS lambda function can run normal after this change.
3. For more visit times at one second and higher security, please consider improving AWS service by paying more money.
4. For any changes to the data visualisation images, please modify it in the Tableau website.
5. For more security about AWS account itself and authority management for AWS services, please consider role management in AWS.

5. The needed knowledge and tools

Our product is easy for use and management. The front end uses the react frame with javascript and we use visual studio code to manage the front end code. The back end uses the AWS RDS to provide database service. AWS amplify is a serverless back end to connect the front end and all related services for the back

end. We also use AWS API Gateway and lambda function with python code to provide useful functions, the links we used in the front end all can be seen in the AWS API Gateway.

For more technical requirements, please visit the maintenance document, the link is below

https://github.com/Wolphatch/industralExperience/blob/master/Maintenance%20document.docx.pdf

6. Technical installation guidance

6.1 Github

The web application uses React to build the user interface. All of the code and relevant resources of the website are hosted by Github. The current location of the code is https://github.com/Wolphatch/industralExperience.

There are many files included in the Repo and some of those files are important.

Filename	**Description**
src	The main directory which includes all React codes and resources.
public	A directory for statical codes and resources.
Package.json Package-lock.json	Directories for package management.
dataVisualization/ DataVizAdvanced	A directory which includes three .twbx files for climate data visualizations. The three .twbx files could be opened and edited by Tableau Desktop.
dataVisualization/ Water.twb	A directory which includes a .twx file for water consumption visualizations. The .twx files could be opened and edited by Tableau Desktop.

dataVisualization/EmbededVisulization	Four txt files which includes the URL for the Embedded Tableau data visualization in the website.

6.2 AWS Amplify

The website uses AWS Amplify to convert the code stored in Github to a web application. The connection is well designed by AWS Amplify service. So if the code in the src directory is edited and uploaded into the Repo, AWS Amplify would automatically build and implement the new version. So testing the source code before uploading it to Github locally is necessary and strongly recommended.

Here are four URLs held by AWS Amplify, which navigate users to four websites:

URL	Branch	Description
https://iteration1.whyisdrought.com	Iteration1	The URL links to the “iteration1” branch in the Github Repo.
https://iteration2.whyisdrought.com	Iteration2	The URL links to the “iteration2” branch in the Github Repo.
https://iteration3.whyisdrought.com	Iteration3	The URL links to the “iteration3” branch in the Github Repo.
https://iwhyisdrought.com	master	The URL links to the “master” branch in the Github Repo. This branch is the latest version of the website.

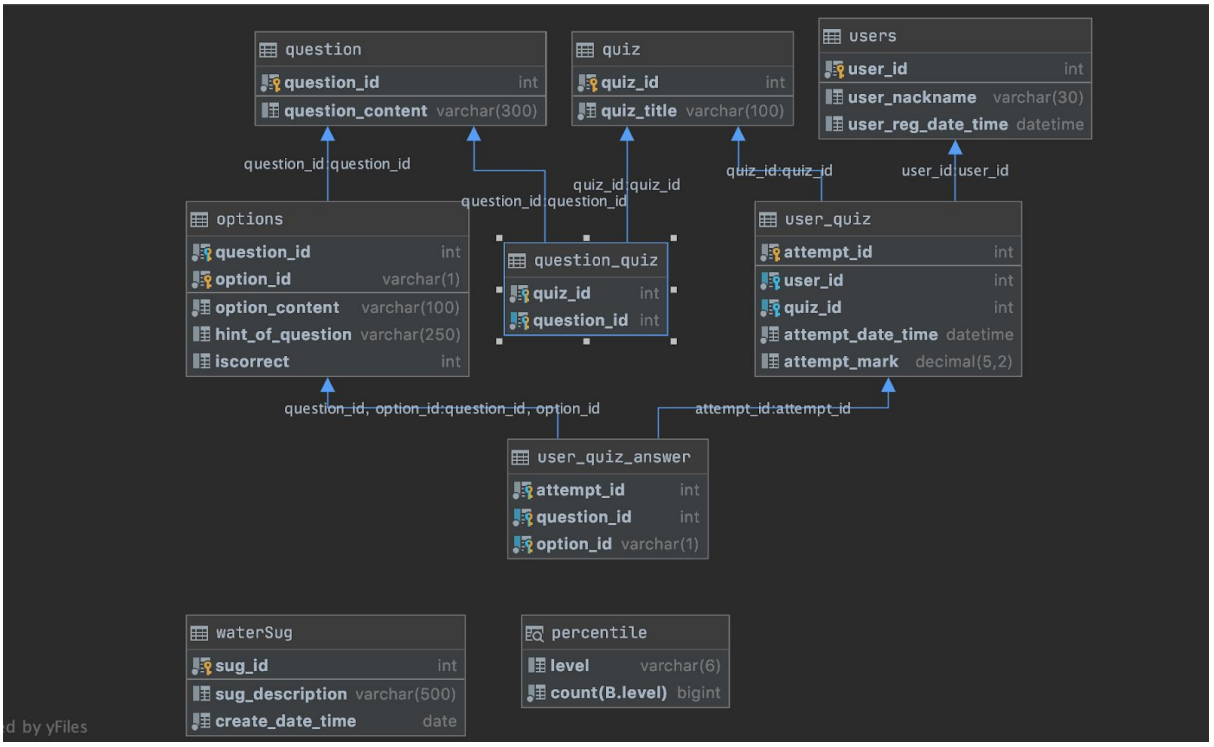
The first three URLs in the table links to websites the team built from the first and the third sprint. The final URL navigates users to the latest version of the website. The code of each sprint is stored in different branches of the Github Repository.

6.3 AWS Route53

AWS Route53 is used to manage the domain of the website.

6.4 AWS RDS

AWS RDS is used to provide database service to the website. MySQL is used as the database management system. The schema and the detailed information about tables in the database are shown in the maintenance document.



If one wants to manipulate the database, the AWS RDS Control Panel would provide detailed information about how to link the database using database management tools. More information about how to use AWS RDS service could be found at https://docs.aws.amazon.com/rds/index.html. One of the most recommended database management tools is dataGrip.

6.5 AWS Lambda

For network security, the website is not allowed to query data directly from the database. Instead, AWS Lambda is used to query and manipulate data in the

database. All of the lambda functions are well-designed so that the functionalities of the website can be implemented. Users can visit https://github.com/Wolphatch/industralExperience/blob/master/Lambda_documentation to get information about each lambda function. The document includes the description of each lambda function and examples of standard input and output.

6.6 AWS API Gateway

Since lambda functions can't be directly visited by the website, AWS API Gateway is used. This service provided REST APIs, which connect the website to lambda functions.

Since APIs only provides a interface for the website to invoke lambda functions, the function, the input and the output of each API is totally the same as the lambda function, For the function of each API, users can visit: https://github.com/Wolphatch/industralExperience/blob/master/Lambda_documentation.

For AWS API Gateway documentation, please visit: https://docs.aws.amazon.com/apigateway/index.html