RMIT UNIVERSITY

School of Science

COSC2626/2640 - Cloud Computing

Assignment 2

Soccer Analytics

A cloud based European Football Data Analysis Web App

Date: 21st May 20

Author

Rafeed Sultaan

Student ID: s3763175

Vishal Beniwal

Student ID: s3759790

Table of Contents

STUDENT CONTRIBUTION AGREEMENT

Project title: Soccer Analytics using AWS cloud services

Student Name: Vishal Beniwal	Student Name: Rafeed Sultaan
Student ID: s3759790	Student ID: s3763175
Contribution: 1. Project Idea and Data sourcing. 2. Data Preparation and Analysis using SageMaker and S3 Bucket 3. QuickSight Dashboard integration 4. Setting up Route 53 and SSL certification 5. Configuring S3 bucket and EC2 instance. (including ELB) 6. Front-end design and dev using HTML, bootstrap and CSS. 7. Setting up IAM roles, custom policies and users. 8. Application user acceptance testing 9. 50% contribution in report making	Contribution: 1. Structured project sprint planning on proposed idea. 2. AWS Cognito Configuration and PHP code for user login. 3. Configuring DynamoDb. Reading & Writing from it using AWS SDKs. 4. Reading & Writing from S3 Bucket using AWS SDK 5. Backend Dev using PHP & JAVASCRIPT 6. Front-end Dev using HTML, CSS and JQuery. 7. Configuring Elastic Beanstalk 8. Application user acceptance testing 9. 50% Contribution in report making
Contribution percentage: 50%	Contribution percentage: 50%
By signing below, I certify all information is true and correct to the best of my knowledge. Signature: VISHAL BENIWAL Date: 23st May 2020	By signing below, I certify all information is true and correct to the best of my knowledge. Signature: RAFEED SULTAN Date: 23st May 2020

LINKS TO DATASET AND APPLICATION

Link to public dataset: http://www.football-data.co.uk/data.php#download

The above provided link contains hyperlinks to all the major football leagues in Europe. For example for English Premier League, the data is provided for last 26 seasons which can be accessed using this link http://www.football-data.co.uk/englandm.php [2]

Link to our web application:

Soccer Analytics: https://www.soccer-analytics.net

Link to GitHub for app source code

Link: https://github.com/vishalbeniwal/soccer-analytics-webapp

SUMMARY OF THE PROJECT

For this assignment, we have chosen to develop an analytical web app named *Soccer Analytics*, which utilises a number of cloud services including AWS Cognito, AWS Sage maker, AWS S3, AWS DynamoDB, AWS Route 53, AWS QuickSight, AWS services SDK, AWS Elastic Beanstalk and AWS EC2

Our objectives for this task are as follows:

- For the proposed web application, design a cloud computing architecture using the knowledge of distributed model and multiple available cloud services, so that each service for the application can have their own dedicated infrastructure and doesn't share resources with other processing services.

- Ensure that the developed application is a scalable and created on virtual machines on the cloud.

- Cloud services are used appropriately and only when it is absolutely needed.

- Analysing and gaining insights from the data using cloud services to use it in the web application to help the user in some way.

- Structured provision to store data which has the feasibility to scale in future.

- Simple and easy-to-use user interface and experience with quick and efficient website response

This report will take you through the significant implementation steps and processes with a user manual using which you will be able to understand and use our proposed web application.

INTRODUCTION

MOTIVATION BEHIND THE PROJECT

The purpose of this project was to develop an analytical web application on European Football league and football match betting with an idea to:

1. To help the application users to know about the upcoming matches in major leagues (England Premier League in present application) with informative analytical data for

each match. This analytical data is further supported by an analytical dashboard using which a user can interpret a team's form, performance and history.

2. This will work as a helping hand for a user to make bets for a particular team more confidently on a third-party website such as bet365.

3. The application is built with a provision where the user's preferences and prediction for an upcoming match will also be captured. This will give an idea about another user's perspective.

4. To help a new user to know about European football world and team history.

FUNCTIONALITY OF THE APP

The Soccer Analytics web is developed to introduce the world of European football in an analytical way to learn about a football team and their history, match fixtures, profit/loss in betting world, and user's predictions.

The app provides the following as functionality to the user:

- A user can register themselves using Login/Register functionality of the app.

- User can view the upcoming football matches from one of the major European leagues i.e. English Premier League.

- User can view host team's overall win ratio and head 2 head (against guest team) win ratio. Host team is a team which is playing in their home stadium and hosting another team for the match.

- User can access the dashboard created on AWS Quick Sight to gain analytical information about a particular team. Functionality is provided to the user for selecting the team they want. The dashboard is analysing last 10 years of data.

- Based on the applied machine learning model on betting odds data from the one of the biggest betting sites, bet365, User will be able to see number of bets placed for a team and predicted loss or profit associated with that team. This will help them to make better betting decisions on third party sites, like bet365.

- Provision to vote is provided for the user. They can use this functionality to give vote to a team for a particular match based on their analysis and own preferences. This vote will be saved in database for any future analysis.

- Using the user profile, a user can change their name/nickname and password.

IMPORTANCE OF THE APP

- This kind of app is required for a user to make better betting decisions based on actionable insights and facts.

- It is an ideal platform to learn about European football teams if you are new to the football world.

- You can keep track of the upcoming matches from major leagues.

SCOPE OF THE APP

- The app can be used as a real-life application since it is built on 10 years of real football data for few major leagues. This could be scaled to all the European leagues including major and minor division teams.

- The app can work as an assistant to aid in sports betting by sourcing data from all the famous betting websites like bet365, bet fair, 888 sport etc.

- The app can be scaled to other types of sports like House racing, Footy, Rugby and Cricket.

- The App can be enabled with a real time data flowing into the database.

- By using the interactive dashboard, users will be able to read the trend and make more formal decision in the field of betting to increase their profits.

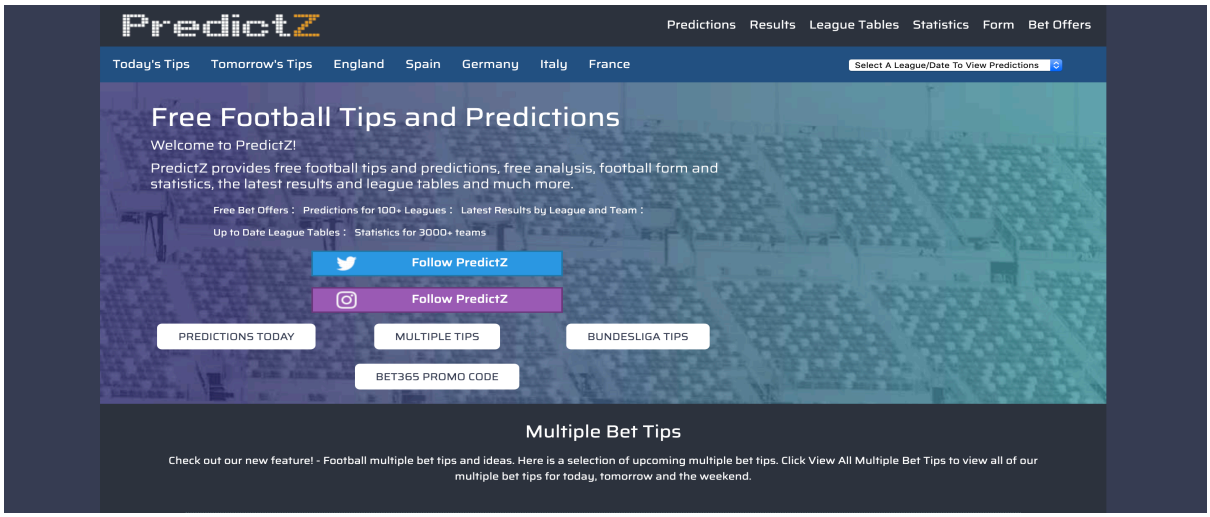
LIMITATION

- Because of the COVID -19 situation, there are no current matches going on in European leagues and therefore, the app is simulated for the previous year matches to showcase the functionality.

RELATED WORK

PREDICTZ

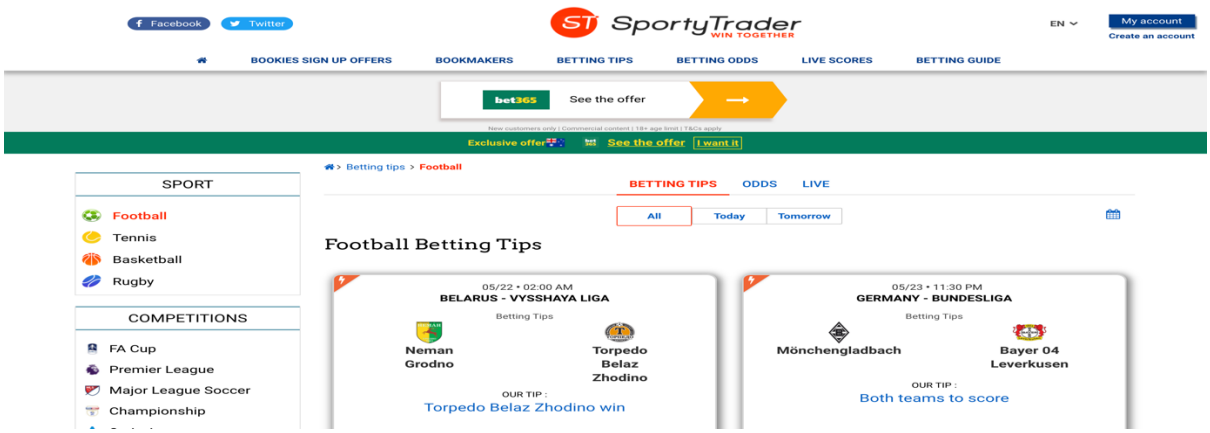
Link to website: https://www.predictz.com/



The predictz is a website with a similar idea. Predictz provides free footballs tips and predictions with analysis, football form, statistics and latest results of the matches. It was also providing multiple types of bet tips and ideas. It covers large number of leagues and football competitions. However, user interface is not that simple and easy-to-use. Most of the information is given in the form bet odds which can be difficult to interpret for a new use.

SportyTrader

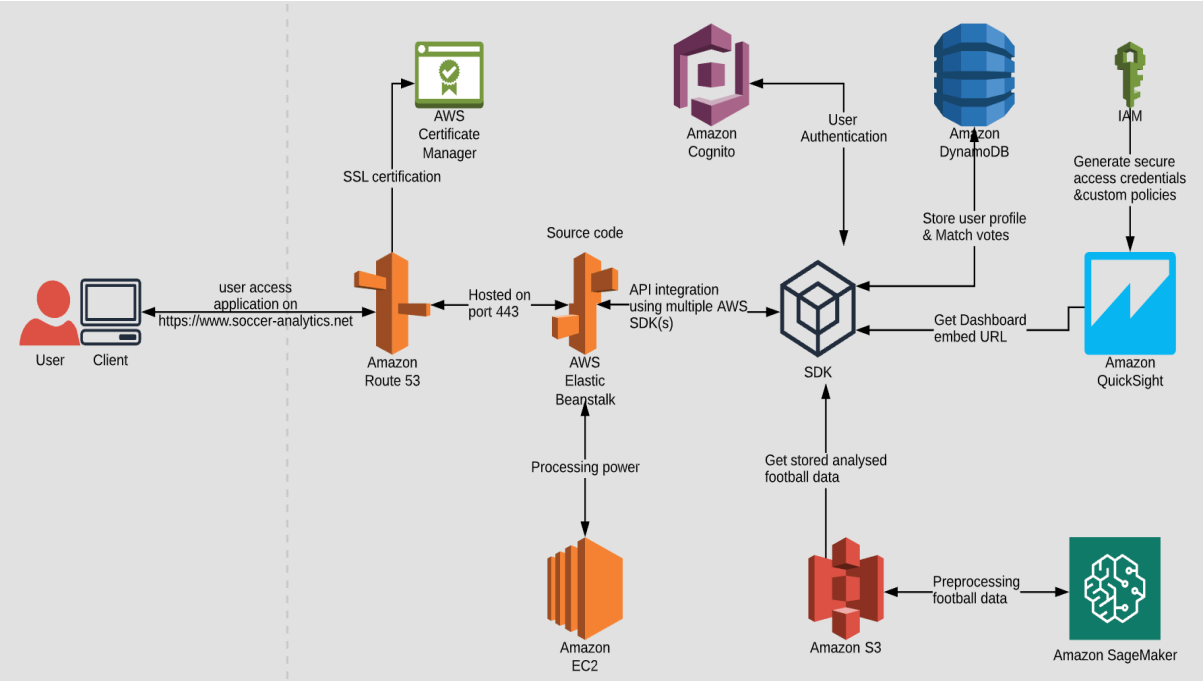
Link to website: https://www.sportytrader.com/en/betting-tips/football/



SportyTrader is another example which provide their match result prediction and have integration with multiple bet making websites also. However, they don't provide any kind of facts or trend analysis related to clashing teams.

SOFTWARE DESIGN/ARCHITECTURE

HIGH LEVEL ARCHITECTURE



DATASET DESCRIPTION

Data was collected from football-data, a famous site in Europe http://www.football-data.co.uk/data.php#download , which captures computer-ready football results, match statistics and betting odds data to use with spreadsheet applications, to help with the development and analysis of football betting systems. They cover the data for every soccer league in Europe. The dataset is divided into each season and you will have to download the dataset for each season of a particular league manually using the website which you can combine later on using some data analytical tool.

Following are some important details/information which is captured by football-data-uk and therefore, being provided in the dataset:

```
Column names = Description
Div = League Division
Date = Match Date (dd/mm/yy)
Time = Time of match kick off
HomeTeam = Home Team
AwayTeam = Away Team
FTHG and HG = Full Time Home Team Goals
FTAG and AG = Full Time Away Team Goals
```

```
FTR and Res = Full Time Result (H=Home Win, D=Draw, A=Away Win)
HTHG = Half Time Home Team Goals
HTAG = Half Time Away Team Goals
HTR = Half Time Result (H=Home Win, D=Draw, A=Away Win)
HS = Home Team Shots
AS = Away Team Shots
HST = Home Team Shots on Target
AST = Away Team Shots on Target
HF = Home Team Fouls Committed
AF = Away Team Fouls Committed
HFKC = Home Team Free Kicks Conceded
AFKC = Away Team Free Kicks Conceded
HY = Home Team Yellow Cards
AY = Away Team Yellow Cards
HR = Home Team Red Cards
AR = Away Team Red Cards
B365H = Bet365 home win odds
B365D = Bet365 draw odds
B365A = Bet365 away win odds
BWH = Bet&Win home win odds
BWD = Bet&Win draw odds
BWA = Bet&Win away win odds
```

Each season runs for approximately 10-11 months and every season a team play approximately 38 to 45 games. For example, in English Premier League or the EPL, 20 teams contest every year. Seasons run from August to May with each team playing 38 matches (playing all 19 other teams both home and away). Most games are played on Saturday and Sunday afternoons (WIKI)(REFERENCE). Therefore, we had around 4,000 records with around 40+ columns for each league which covers matches from last 10 years.

IMPLEMENTATION

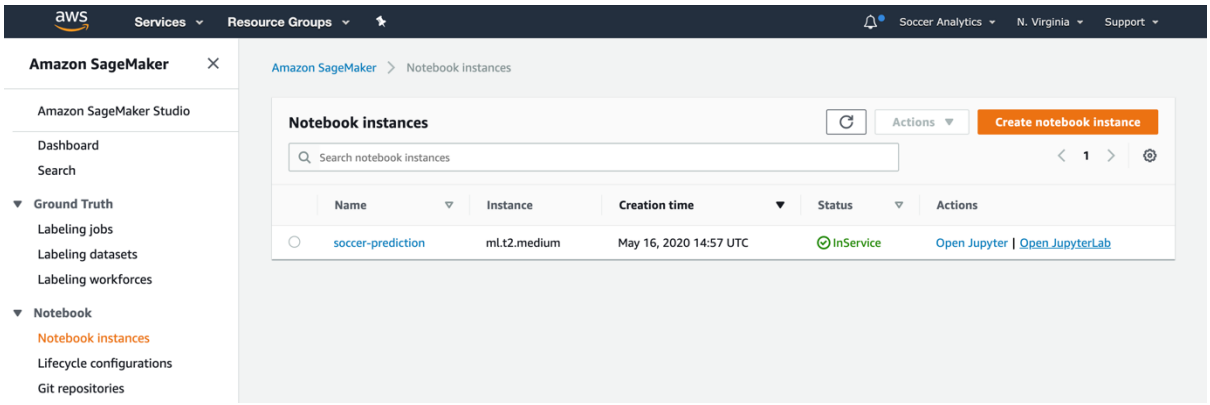
1. Creating and configuring AWS S3 Bucket

- Sign into AWS Management Console and open the Amazon S3 console.
- Go to create bucket and provide the name for the bucket. Also select your preferred region from the provided list of regions in the given drop down.
- Disable the checkbox for “block all public access” to grant public access and click on create bucket.
- Once the bucket is created, open the bucket by clicking on the hyperlinked name of the bucket.
- Using the upload button, select the downloaded dataset from the local machine to upload it into your S3 bucket.
- Select the uploaded items and make them public using the dropdown “Action” button given inline to upload button.

2. Setup up and Configure Amazon SageMaker

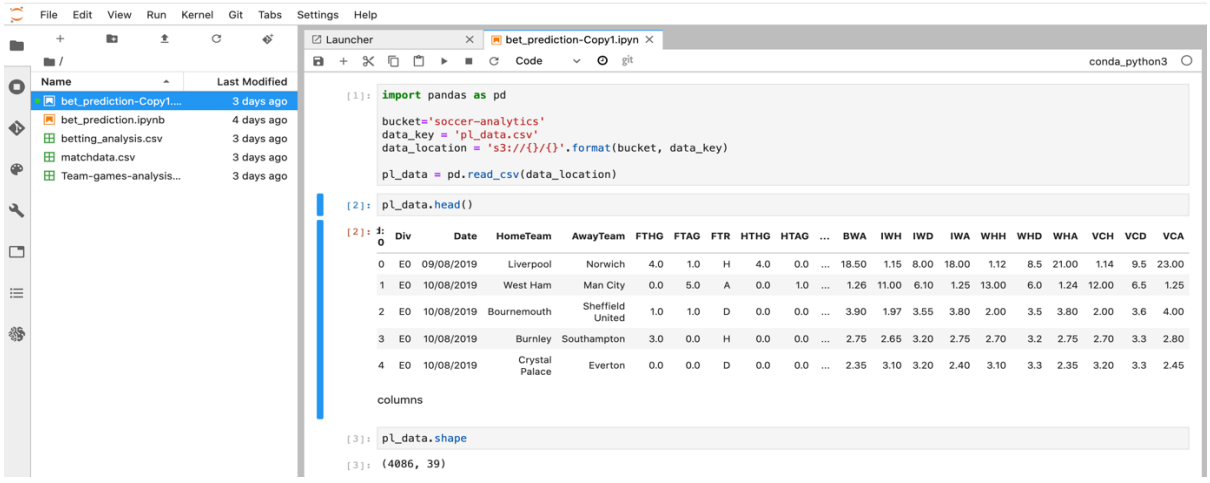
- Using the AWS Management Console, open Amazon SageMaker.
- Click on Notebook instances given on the left panel under Notebook label.
- Click on “create notebook instance” button and give a name to the notebook.
- Choose the Notebook instance type as “ml.t2.medium” from the dropdown.
- Keep Elastic Inference as none.
- Under the permissions and encryption tab, for IAM role, create a new role and give this role the access to “Any S3 bucket” and click on create role.
- Leave rest of the field as it is and click on “create notebook instance”.
- Once the notebook instance is created, wait for the status to turn from pending to “inService” as shown in Snapshot 1. This process will take few mins.
- Once the instance is ready to use, click on “JupiterLab” which will take you to the next page where you will select the environment for your notebook.
- Choose “conda_python3” as environment from the listed options.

- Now, using python code, import panda's library and read data from S3 bucket that you uploaded in the first step as shown below in snapshot 2.



Snapshot: 1 AWS SageMaker

- Once the data is loaded, use the notebook to clean, transform, analyse and model data using data analysis and machine learning techniques.
- After processing, save the results back into the S3 bucket to be used further by other services.

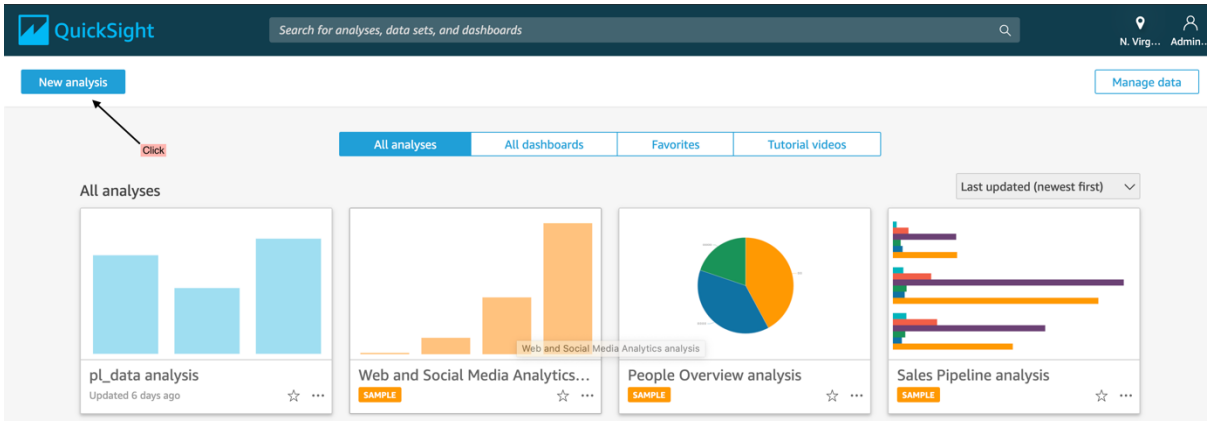


Snapshot: 2 Data reading from S3 to AWS SageMaker

3. Creating interacting Dashboards using AWS QuickSight service

- Using the services available in AWS management console, search for QuickSight under the Analytics category.

- Select QuickSight and register yourself as a user of QuickSight by providing your email ID. (Note: This service can't be accessed using the AWS educate account as it need some IAM role permissions and therefore any attempt to register a user will be declined. Hence, we create our own AWS account to get control of permissions)
- Once you logged in into the QuickSight Dashboard, you would be able to see a button to make new analysis as shown in snapshot 3.

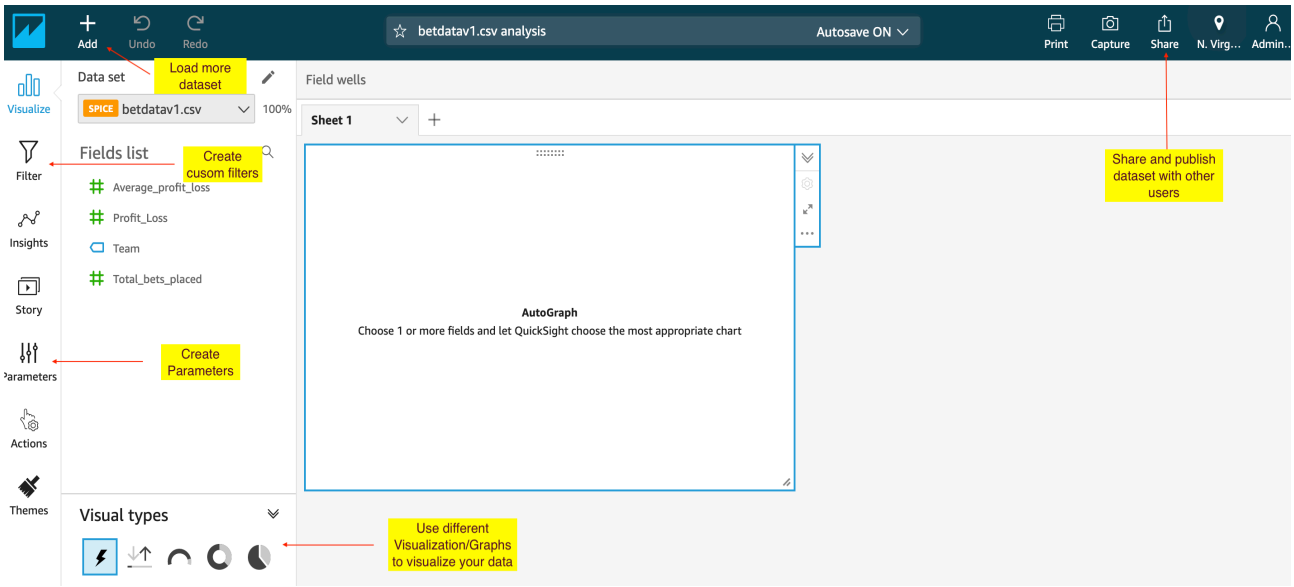


Snapshot: 3 QuickSight Dashboard

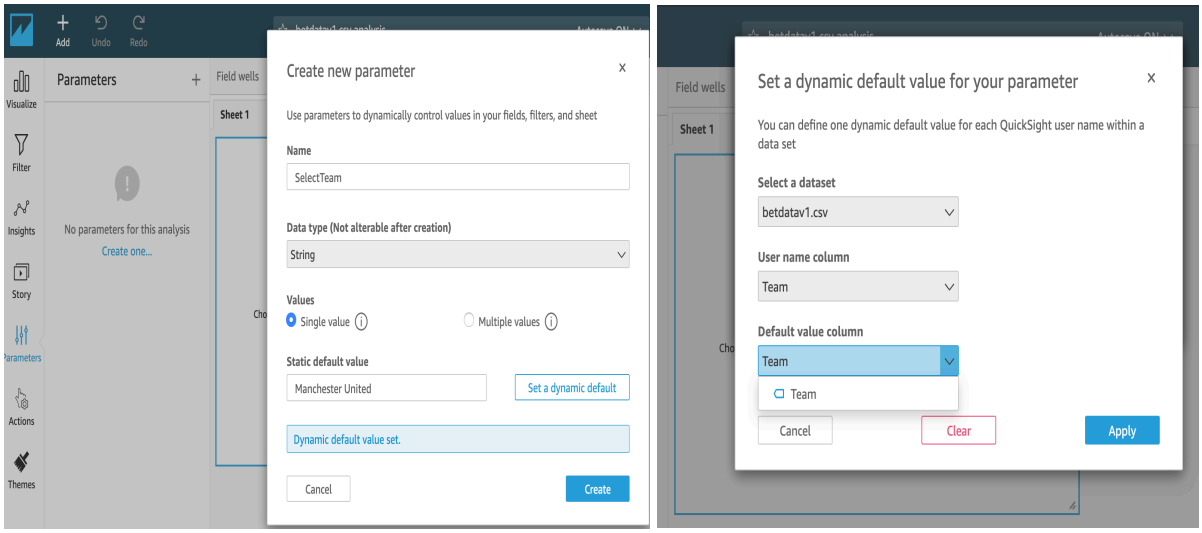
- Upload the dataset using new dataset option given on the panel. There are lot of options to source the dataset. We uploaded the dataset using our local machine which was downloaded from S3 bucket.
- Once the uploading is done, QuickSight will give a prompt to move forward by clicking next. Click on next and then finally click on visualize to get into the QuickSight panel (Snapshot:4) to create dashboards from the uploaded dataset.
- Once the dashboard is created, you can set the parameters which will act as controls to filter your data and visualization. To create a new parameter,

  select parameters option from the left panel and then click on "+" given against the heading parameters.
- Provide the required fields such as name for the parameter, select the data type of the parameter, select whether it will be single or multiple value selection. To populate the data directly from the dataset, click on the given button "Set a dynamic default".

- Choose the dataset and column for the parameter. The example is given in the Snapshot: 5 below. After this, click on apply and create the parameter. Once the parameter is created, select the “Add Control” option using the dropdown button given against the created parameter to display it on your dashboard.
- Once the dashboard is ready with the required visualization and controls, you have to publish it using the share option given on the top bar. Now you will be able see this dashboard on your home panel of QuickSight under “All Dashboards” tab.



Snapshot: 4 QuickSight Visualization panel



Snapshot 5: Creating parameters for visualization controls

4. Configuring QuickSight for Dashboard Embedding

The idea was to embed the published dashboard on our web application to provide interactive football world analytics to the user. The official documentation is also provided by AWS on their website under this the below link https://docs.aws.amazon.com/quicksight/latest/user/embedding-dashboards.html

The following steps are required to achieve this:

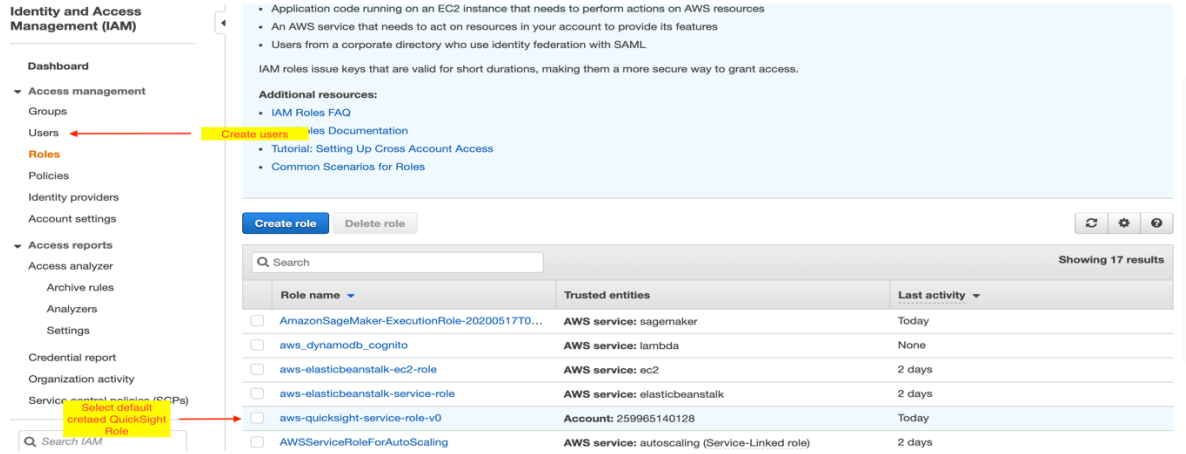
Step 1: Add Domains to the Allow list

- Using the QuickSight panel, click on the user profile icon on the top right bar and choose “manage QuickSight” from the dropdown.
- Upgrade the account to Enterprise version. If it is already an Enterprise version, then you can skip this step. (Note: Dashboard Embedding option is only available in enterprise version which is a paid version)
- In the manage QuickSight panel, select Domain and Embedding option from the left panel and provide the domain for whitelisting. (Check “include subdomain” checkbox)

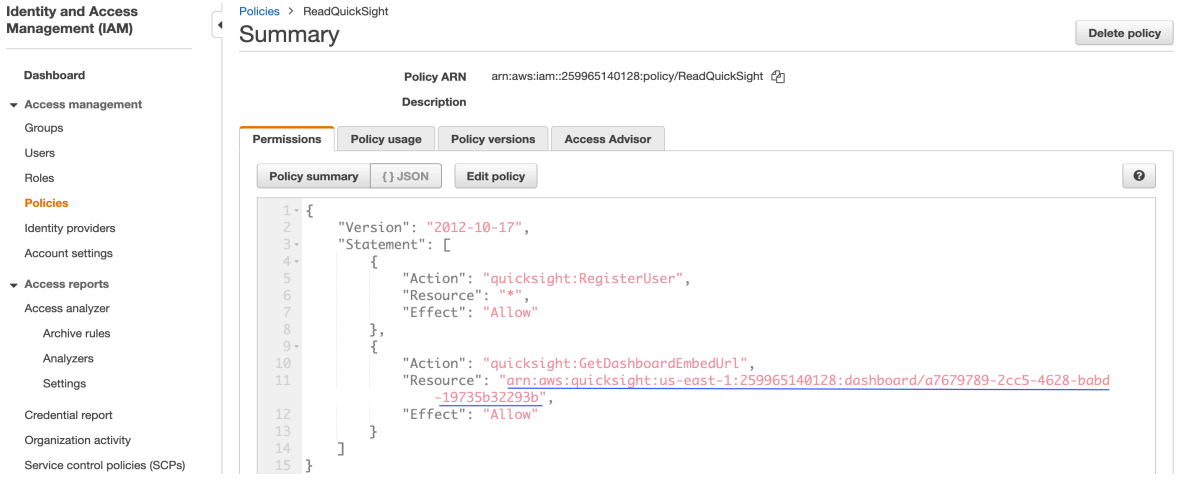
Step 2: Set Up Permissions for Dashboard Viewers (Root user can’t be used for this step)

- For this step, you need to use another AWS service called **IAM** (Identity and Access Management). Using the AWS management console, search for IAM in services.
- Open the IAM management console and go to “Users” using the left panel.
- Click on “Add User” button on the screen to create a new IAM user. Provide the name for the new user and select “AWS Management Console access” in the access type.
- Create a custom password for this user and uncheck “User must create a new password at next sign-in” option. Then, click on “Next: Permissions” button.
- To set the permissions, choose “Attach existing policies directly” option from the screen and search for “AdministratorAccess” policy.
- Go next, skip the tag and finally, click on “create user” button to create a new IAM role with full AWS management console rights. Similar to root users’ rights.

- Once the user is created, a new custom role needs to be defined to ensure that the reader can access Amazon QuickSight in a read-only fashion, and not have access to any other data or creation capability.
- For this, select roles from the left panel of IAM management dashboard and search and select “aws-quicksight-service-role-v0” (Refer to Snapshot 6)
- Inside this default role, attach a custom policy by clicking on “Attach Policies” button and then, click on “create policy” button.
- Select JSON tab to create custom policy and paste the JSON code given on AWS documentation. Please replace the dashboard ID with your own published dashboard ID in the JSON before moving forward (Refer to Snapshot 7). Get the Dashboard ID from your published dashboard’s URL.
- Click on review policy button to save the changes and give the name of this new policy and then click on “create policy” button to finish the process.



Snapshot 6: IAM Dashboard



Snapshot 7: Custom JSON policy for QuickSight Reader

Step 3: Authenticate the User and Get the URL

- In the IAM dashboard, go to users again and select the new user we created in the previous step.
- Inside the user, select the security credentials tab and generate the ACCESS KEY and SECRET KEY for this new IAM user. Keep them safe somewhere.
- This page will contain a Console sign-in link under Sign-in credentials heading. Copy paste this URL to another browser and now, login into newly created IAM role console using IAM username (name of IAM role) and custom password which was set during the user creation process.
- After log-in into the console, go to QuickSight service and provide an email ID to register this user on QuickSight panel. Now we are ready to call dashboard-embed-URL API from our app server.
- The PHP code is given below (Snapshot 8) to demonstrate the process of getting the URL

```php
<?php
    require 'vendor/autoload.php';

    use Aws\Sts\StsClient;
    use Aws\QuickSight\QuickSightClient;

    $access_key = 'AKIATZBZBHSQMUGKACSQ';
    $secret_key = '[redacted]';
    $region = 'us-east-1';

    $client = QuickSightClient::factory(array(
        'credentials'=> array(
            'key' => $access_key,
            'secret' => $secret_key
        ),
        'region' =>  $region,
        'version' => 'latest'
    ));

    $result = $client->getDashboardEmbedUrl([
        'AwsAccountId' => '259965140128', // REQUIRED
        'DashboardId' => 'a7679789-2cc5-4628-babd-19735b32293b', // REQUIRED
        'IdentityType' => 'IAM',

    ]);

    $url = $result['EmbedUrl'];
```

Snapshot 8: PHP code to call dashboard-embed-URL API

5. Creating and configuring AWS Cognito

- Using the AWS management console, go to AWS Cognito service.
- Click on “Manage user pools” and then click on “Create a user pool” button.
- Provide a pool name and then select “step through settings” option to move next. Following are the steps to setup the pool for app.

How do you want your end users to sign in?

- For this select Email or phone number radio button and then, select “Allow email address” radio button to allow only email ID to be used for login into the app.

Which standard attributes do you want to require?

- For this section, just select email and name as the attribute and go next.

What password strength do you want to require?

- Reduce the Minimum length from 8 to 6 characters.

Do you want to allow users to sign themselves up?

- Check “Allow users to sign themselves up” and go next.

Do you want to enable Multi-Factor Authentication (MFA)?

- Keep it off

How will a user be able to recover their account?

- Select “Email only” option from the listed options

Which attributes do you want to verify?

- Select “Email” only and go to next step.

Do you want to send emails through your Amazon SES Configuration?

- Select “Use Cognito (Default).

Do you want to customize your email verification messages?

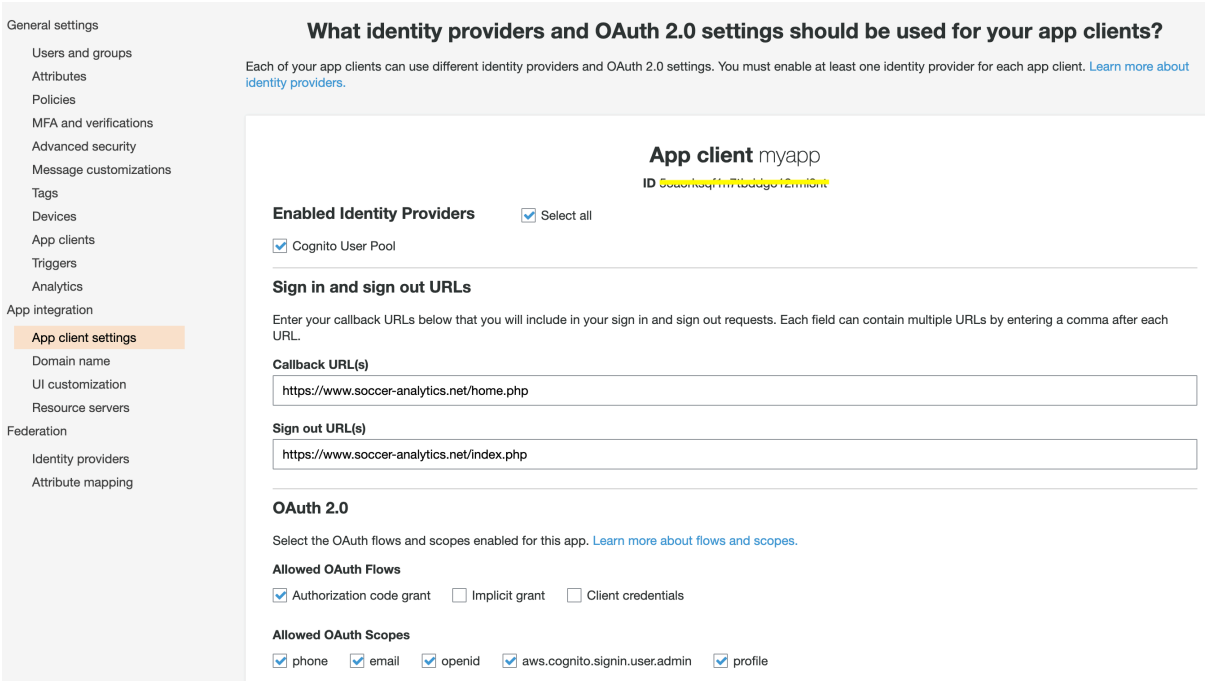
- In verification type, select “link” and go to next step and skip tag.

Do you want to remember your user's devices?

- Select “No” and move forward.

Which app clients will have access to this user pool?

- Click on “Add an app client” and provide a name for app client.
- Uncheck the checkbox given for “client secret”
- And click on “create app client” button to create an app client for the pool.
- Review the provided information and lastly, click on “create pool”.
- Once the pool is created, note down the generated pool ID and app client ID.
- To setup the created app client for the pool, follow the information given in the snapshot 9 and save changes.



Snapshot 9: AWS Cognito app client settings

6. Creating tables in DynamoDB

DynamoDB is used to create tables for the purpose of storing users' details and user votes for an upcoming match. The steps to create a table in DynamoDB is given in the week 8 tutorial sheet (Link is provided below) (*https://rmit.instructure.com/courses/67441/files/11177732/download?wrap=1*)

Follow the same steps to create the following three tables with the mentioned partition (Primary key) and sort keys.

Table Name: userTable

Email (partition key)	fullname	password	favteam
xyz@john.com	john	dummy@123	Manchester United

Table name: match_votes

matchID (Partition Key)	hostteam_t1	Guestteam_t2	matchdate	T1_votes	T2_votes	Draw_votes	Last_modified_dt
PI_3026	Man United	Southampton	2017-05-17	5	2	3	2017-05-17

Table name: user_votes

userID (partition key)	matchID (sort key)	T1_votes	T2_votes	Draw_votes
Xyz@john.com	PI_3026	1	0	0
	PI_3037	0	1	0

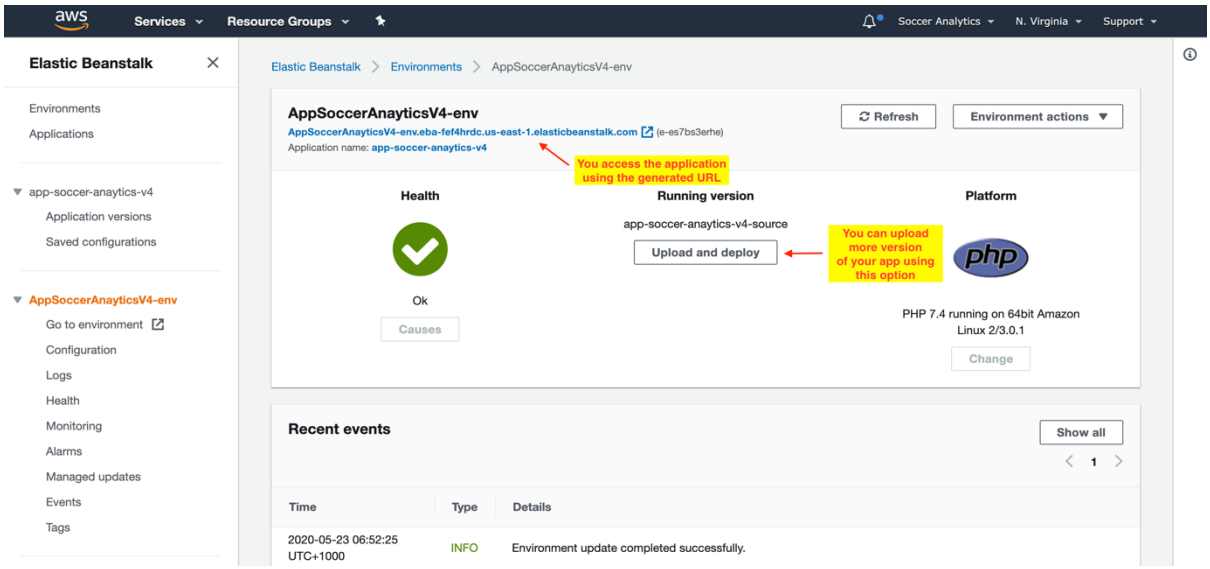
7. Creating and configuring Elastic beanstalk environment

To upload the source code of the app in a zip file, AWS Elastic beanstalk environment needs to be created and configured.

- Using AWS management console, open Elastic Beanstalk service.
- Click on “Create new environment” and then, select “Web Server Environment” in select environment tier. Now click on “Select” button to move forward.
- Provide the application name. The environment name will be populated automatically based on your application name.
- (Optional) Provide the desirable domain name and check the availability.
- Next, under platform section, select “Managed Platform” and choose PHP as your platform from the given dropdown list. Ensure platform branch is configured to “PHP 7.4 running on 64bit Amazon Linux 2” and platform version to “3.0.1”.
- Next, under application code, select “upload your code” radio button and upload the application folder using local file option. (Make sure you don’t zip the parent folder with your application file. Just application files need to be zipped)
- Now click on “configure more options” button given at the bottom of the form and then, edit the “capacity” configuration. On this page, under auto scaling group section, change the environment type from single instance to load balanced. Once done, click on “save”.
- Now edit the load balancer configuration, and under application load balancer section, click on “Add listener”. Provide “443” in port and select “HTTPS” in protocol. For SSL certificate under security settings, select “soccer-analytics.net - 7ddba0b2-c7ba-4eda-97d7-c5a34bf3b5ff” from the

available ssl certificates and click on “Add” button and then click on save button to save the configuration.

- Now click on “create environment”. The process to create the environment and application will take few mins. Once the app is ready to your see a screen shown in snapshot 10.



Snapshot 10: Elastic Beanstalk environment

8. EC2 instance Configuration

Open the EC2 service using the console. Go to “instances” using the left panel. You will see an instance running as you created an environment on EBS in the previous step. It’s using this instance to run the deployed application.

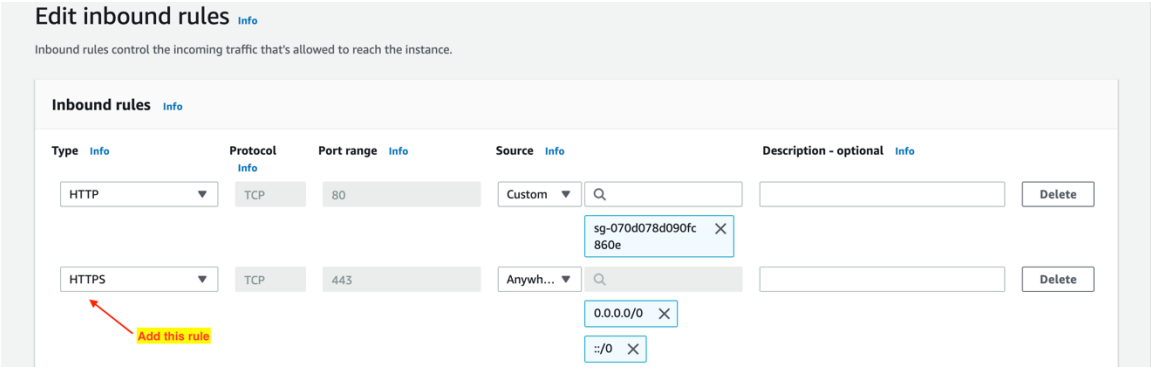
- Here, scrolls the page to right until you see “Security Groups” column. Click on the security-group value in link form.

- This will take you the security group page with applicable preselected security-group-id.

- On this page, click on “Inbound rules” and then, click on edit. Now add an extra rule as shown in snapshot 11 below.

9. Route 53

At last, we need to host the EBS generated app on a custom domain with issued SSL certificate to enable AWS QuickSight embedding service inside the app. (Note: To



Snapshot 11: Edit Security group's Inbound rules

whitelist a domain for QuickSight dashboard embedding, the URL needs to be HTTPS enabled and not AWS URL or HTTP enabled).

- Buy a domain from the AWS Route 53 service. Request a SSL certificate using AWS certificate manager for *.soccer-analytics.net, www.soccer-analytics.net and soccer-analytics.net addresses.

- To validate the certificate, add the generated CNAME into the purchased domain using "Add Record in Route 53" for each provided address and wait for at least 30 mins to see the change in status from "pending validation" to "issued". Now you have a SSL certified domain.

- Now, to point this domain to the EBS generated domain, open Route 53 panel. Go to "Hosted Zones" from the left panel.

- Select the domain name and then click on "Go to record set" button on the top.

- On this page, click on "Create Record Set" and add A Record for .soccer-analytics.net

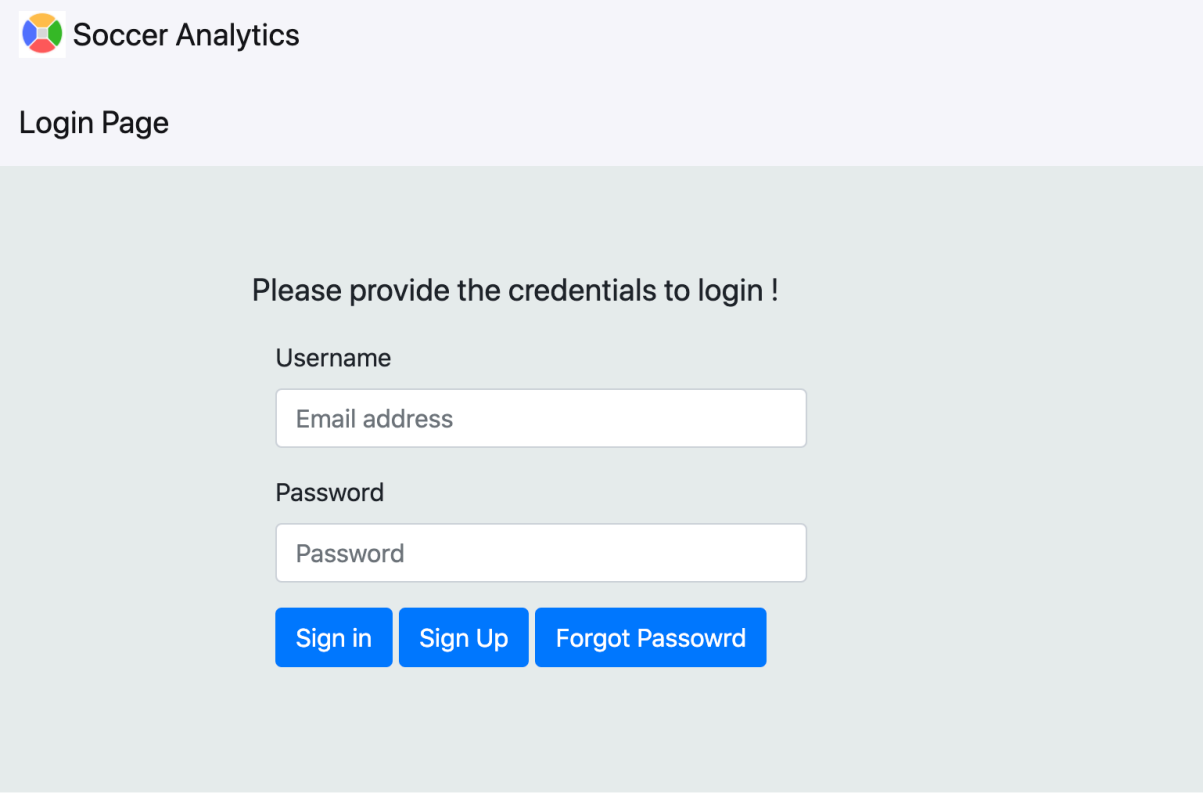
- Select "Alias" option as "Yes" and search & select the EBS URL in "Alias Target" search bar and click on "create"

- Follow the last two steps for another A record for www.soccer-analytics.net.

- Wait for some time to reflect the changes on the custom domain.

USER MANUAL

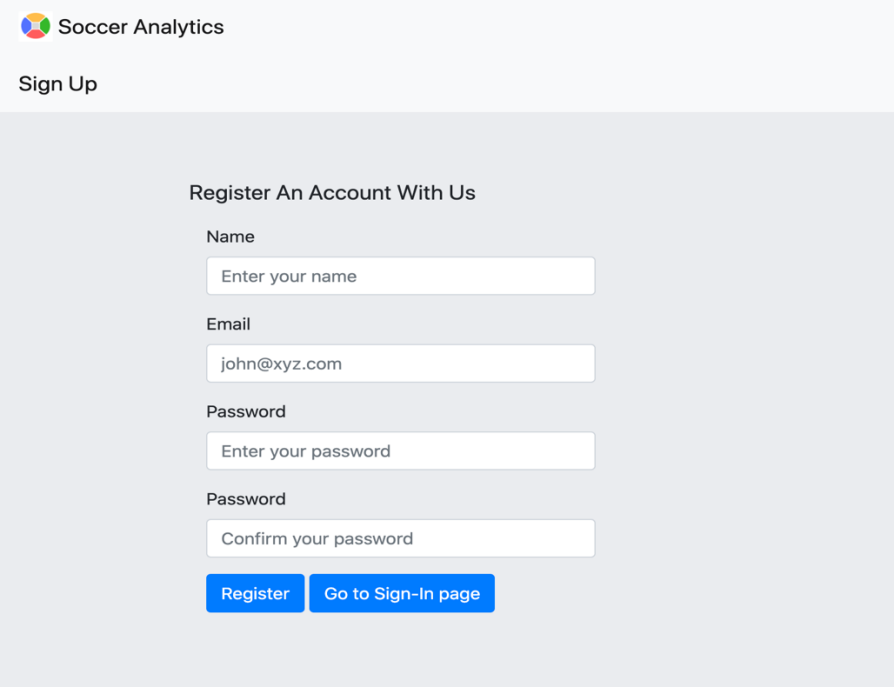
Login Page (Snapshot 10)



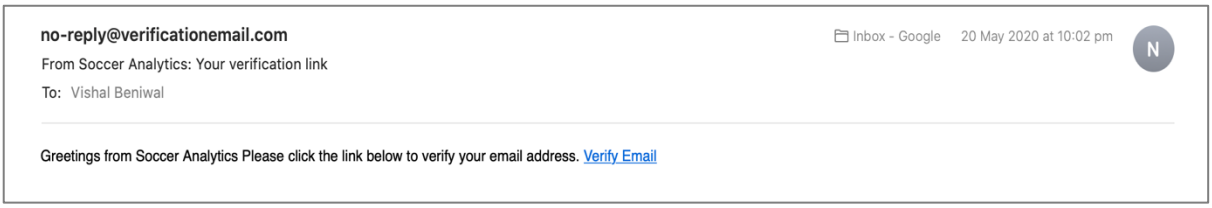
Snapshot 10: Soccer Analytics Login Page

When a user opens the Soccer Analytics app, they will land on the login/Register home screen where a user simply needs to enter his/her login details and press the “Sign In” button to redirect to the application dashboard.

If a user doesn’t have an account on the app, he/she can click on “Sign Up” to register themselves on the app. They will need to provide the fields shown (Snapshot 11) for registration. To confirm the registration, user will get a verification link on the provided email ID (Snapshot 12). Upon verification, user can login into the application dashboard using the provided email address and password.



Snapshot 11: Soccer Analytics Sign Up page



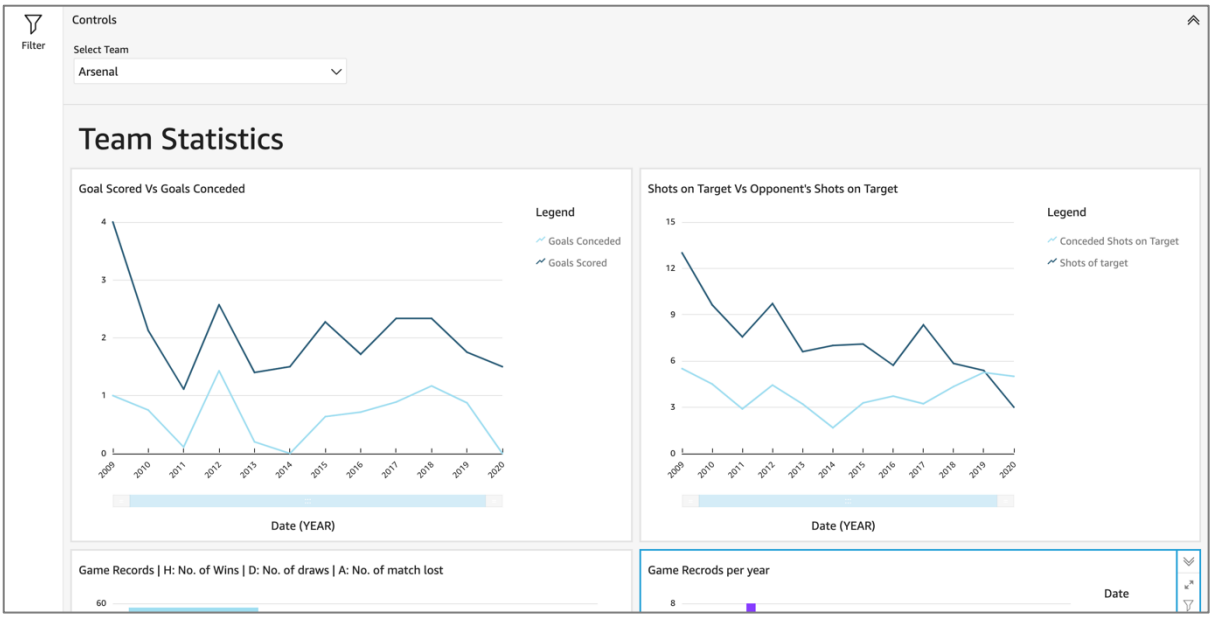
Snapshot 12: Verification link on user email

App Dashboard (Snapshot 13)

Snapshot 13: Soccer Analytics Dashboard

On the dashboard, user will be able to see the upcoming matches with the involved team's performance and betting statistics to assist them to know which team has higher chances of winning the match. Using the user prediction section of the table, the user will be able to see other user's perspective also. Here, T1 refers to the host team, T2 refers to the guest team and X refers to the draw between T1 and T2. The counter against each button is the count of votes that other users have given for a particular upcoming match. A user can reset their choice of vote until they press "vote" button. Upon pressing the "vote" button, user won't be able to change their voting choice for that particular match.

Analytics Dashboard (Snapshot 14)



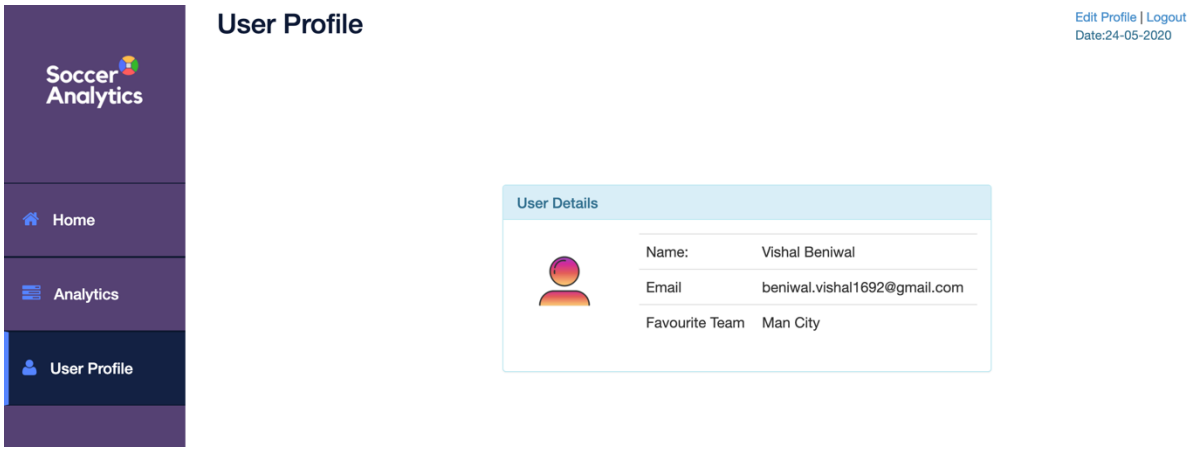
Snapshot 14: Analytics Dashboard

To further support the user decision, the app has an analytical dashboard which shows the present and past facts related to a football team. Dashboard contains multiple simple interactive actionable visualizations using which a user can do their own analysis to take better decisions and can also be used to gain knowledge about their favourite team.

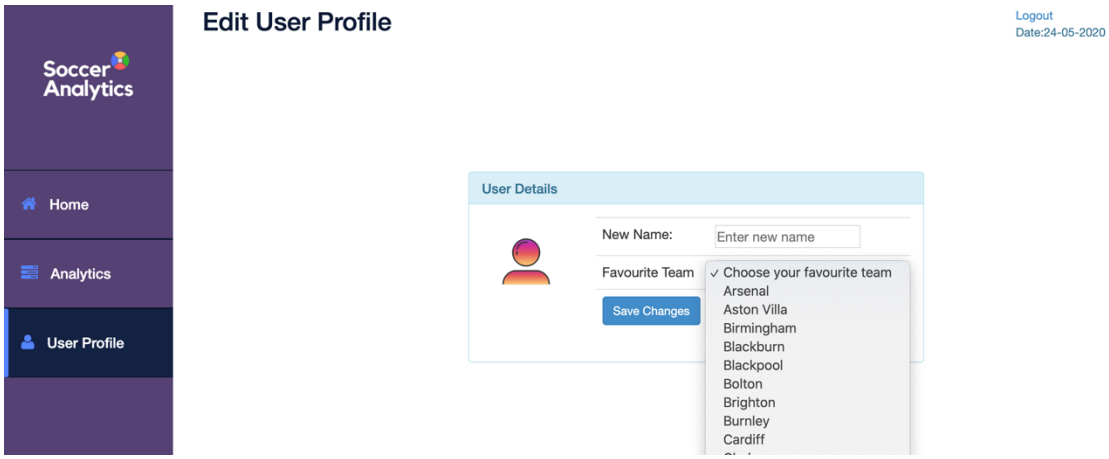
User can select their desirable team from the dropdown given and scroll down to see all the results. The dashboard is embedded in the analytics tab on the left navigation panel.

User Profile (Snapshot 15)

The user profile page shows the logged in user's details and have a functionality to change their name and password using the "Edit profile" link on the top. The other option on this page is to logout and end the session using the "Logout" link, also on the top. The left panel is the navigation panel using which you can switch between Home, Analytics and User profile.



Snapshot 15: User Profile Page



Snapshot 16: User profile edit page

The user profile page shows the logged in user's details and have a functionality to change their name and password using the "Edit profile" link on the top (snapshot 16). The other option on this page is to logout and end the session using the "Logout" link, also on the top. The left panel is the navigation panel using which you can switch between Home, Analytics and User profile.

REFERENCES

[1] AWS Documentation (2020). Retrieved 24 May 2020, from https://docs.aws.amazon.com

[2] England Football Results Betting Odds | Premiership Results & Betting Odds. (2020). Retrieved 24 May 2020, from http://www.football-data.co.uk/englandm.php.

[3]] Mark Otto, a. (2020). Introduction. Retrieved 24 May 2020, from https://getbootstrap.com/docs/4.5/getting-started/introduction

[4] PHP Tutorial. (2020). Retrieved 24 May 2020, from https://www.w3schools.com/php