**Continuous Integration And Deployment With AWS**

* **Services Overview:**

**Github**

**Travis CI**

**AWS**

* **Github setup:**

**Create Github account if you don’t have.**

**Create a new repo named docker-react.**

**Push our frontend project files to the repo.**

* **Travis CI Setup:**

Diagram

Description automatically generated

Go to travis-ci.org

Login with Github account.

Under profile, go to settings.

There you will find all the repositories you created in Github.

Select the “docker-react” from the list. And click on “activate repository”.

After completing all the above steps, you must see the same screen as below.

Graphical user interface, text, application

Description automatically generated

* **Travis YML File Configuration:**

We have to create a configuration file for travis to explain what needs to be done against that repository.

Create a file named “.travis.yml” file and put it under projects root folder.

Note: The file name should have the “.” In front of the name.

As we are writing all the information on travis.yml file, if we just run the test cases using npm run test, once the test cases ran successfully, it will wait for an input from the user to continue with further steps. So below command will not work with travis, as it will be expecting an input from enduser.

docker run image\_id npm run test

Use below command to exit once the test cases ran successfully.

docker run image\_id npm run test -- -- coverage

Diagram

Description automatically generated

* **Add the following contents on .travis.yml file:**

sudo: required

services:

- docker

before\_install:

- docker build -t dspandiyan/docker-react -f Dockerfile.dev .

scripts:

- docker run dspandiyan/docker-react npm run test -- --coverage

* **AWS Elastic Beanstalk:**

Create an AWS account.

Login to AWS console.

Search for elastic bean stalk in the search bar.

Create a new elastic beanstalk instance.

* **More on Elastic Beanstalk:**

**Diagram

Description automatically generated**

Elastic beanstalk checks the incoming traffic for the given application, if the traffic reaches a threshold it will add more instances to handle the high traffic.

Diagram

Description automatically generated

* **Travis config for deployment:**

**A picture containing table

Description automatically generated**