## Assignment 4

Udit Desai (00001176777)

COEN 315: Web Architecture & Protocols

March 4, 2017

#### Domain Name

* bart.incognitech.in

#### AWS EC2 Instance

* Created a new EC2 instance on Amazon Web Services console.

#### Packages Installed

* Nginx
* Docker
* NodeJS
* MongoDB
* MeteorJS - NodeJS Framework for Cross Platform Applications

##### Access Information

* Instance Public IP Address - 54.191.99.167
* PEM Key to access the server instance - [Download Link](https://drive.google.com/open?id=0B8EVno684cJXWEdNOElrbEVPV0U)
* Project Source Code - <https://github.com/desaiuditd/bart-guide>

##### How to Access from SSH

* In the terminal, apply following command to login to EC2 instance via SSH. Use the pem key mentioned earlier. (Download it from the link)
  + ssh -i <path-to-pem-key> ubuntu@54.191.99.167
* The website webroot path is as follows:
  + /var/www/html/portfolio
* You can use the same combination of pem key, user (ubuntu) & Server IP (54.191.99.167) to access the server instance if you are using any GUI tool like Putty or FileZilla.

#### Meteor App Information

* Meteor app is deployed using [MUP](https://github.com/arunoda/meteor-up) tool with help of Docker.
* Meteor app is running on port 3000 on EC2 server within a docker container.
* This app is made public via a local Nginx HTTP Proxy on port 80.

#### Nginx Information

* A local HTTP proxy is created from port 80 to port 3000 to redirect all external the traffic towards the app.
* So when we access <https://bart.incognitech.in> from browser, the Nginx redirects the request to port 3000 where Meteor app is running.
* Response from Meteor app from port 3000 is sent back to Nginx and Nginx sends this response on the browser via port 80.
* You can check Nginx config using following command. It will display the Nginx Proxy config on terminal.
  + cat /etc/nginx/sites-available/bart.incognitech.in

##### How to Access from Browser

* Website URL: <http://bart.incognitech.in/>
* **Home Page** - Gives general information about the web app
* **Stations Page** - A list is displayed for all the BART stations available.
* **Single Station Page** - You can go to a Single Station page by clicking on any Station from the Stations list page.
* **Trip Page** - Select Source Station. Select Destination Station. Click on Go button. Web App will fetch all the information related to available trips and routes.
* **Trip Google Map** - If you click on View Map button for any trip, a popup will be shown and you can see Google Map with the route for that trip.
* **Remaining Time Counter** - Below the trip you can see time counter for all the available real time estimate for a trip.
* **Visit Count** - In the header, in top right corner, a welcome message with visit count is displayed.

#### Answer for The Same-Origin Policy - Part #5

My website is accessing the BART API on the server end. The Same-Origin policy only applies when you access any API of third party domain using HTTP requests on the browser itself. So my server script code provides the endpoints to access the BART API on the same domain and that’s why the client side JavaScript code from my website can access the API using HTTP requests from the browser because they are hosted on the same domain.