How to Deploy BucStop

1. How to set up the EC2 instance in AWS

- Create an EC2 instance and enable HTTP and *(optionally) HTTPS* on it: https://youtu.be/B22d7D6SPDU?si=THdVBNXZ3zywmzX1.
    - *To use HTTPS: I recommend you attach an elastic IP for your EC2 instance. Then you will need to point the DNS record of the domain you want to use to the IP you attached.*
- Create a new user account with sudo access on the instance: https://docs.rackspace.com/docs/create-a-sudo-user-in-ubuntu
- Add the files under the *scripts* folder in the BucStop project to a folder called *scripts* on the EC2 instance.
- Install docker on the EC2 instance if it isn't already installed

2. Docker Account Setup

- Create docker hub repositories with the same Naming as ours i.e.
    - {Username}/bucstop
    - {Username}/bucstop_apigateway
    - {Username}/bucstop_micropong
    - {Username}/bucstop_microtetris
    - {Username}/bucstop_microsnake
- Then you need to build the docker images locally and then push them to their respective repositories for first-time setup, or update the github actions file (under .github/workflows/docker-image.yml – to use your docker repos. You'll have to do this anyways. This means when you push a change to the latest sprint branch, it will auto update your docker repository)

- Build the two docker images. Update the *$(docker-user)* variable to your docker username.
  - `docker build -f BucStop/Dockerfile -t $(docker-user)/bucstop`
- Push the two docker images
  - `docker push $(docker-user)/bucstop`

4. Run Deploy.sh

- After all of this is setup you just need to run the Deploy.sh script to start the application.
- After it is running you only have to use it again to update your docker containers on the server.
- If everything is configured correctly, you should be able to go to the IP of the EC2 instance and see the website!