Mac Deployment using EC2 instances and docker

https://www.youtube.com/watch?v=oK3Olw4wlLs&list=PLLVrCqkyEWa_6qhZhzTfgVTu3WL6ywW2G&index=1

The deployment is automatic after setup. All that needs to be done to deploy for automatic deployments are the following:

- Add changes with feature branches off of the dev branch
- Pull request back into dev when done
- When you're ready for deployment, do a pull request from the dev branch in the main branch. Deployment will trigger automatically once it is merged.

How it works

- Deployment uses the deploy.yml workflow in the GitHub repository.
- Links to the two repositories:
    - https://github.com/chrisseals98/BOBBY
    - https://github.com/chrisseals98/GameInfoMicroService

Most things are relatively the same, but I ran into some issues on mac that will be covered. I will be skipping the optional SSL Cert and the github secrets part.

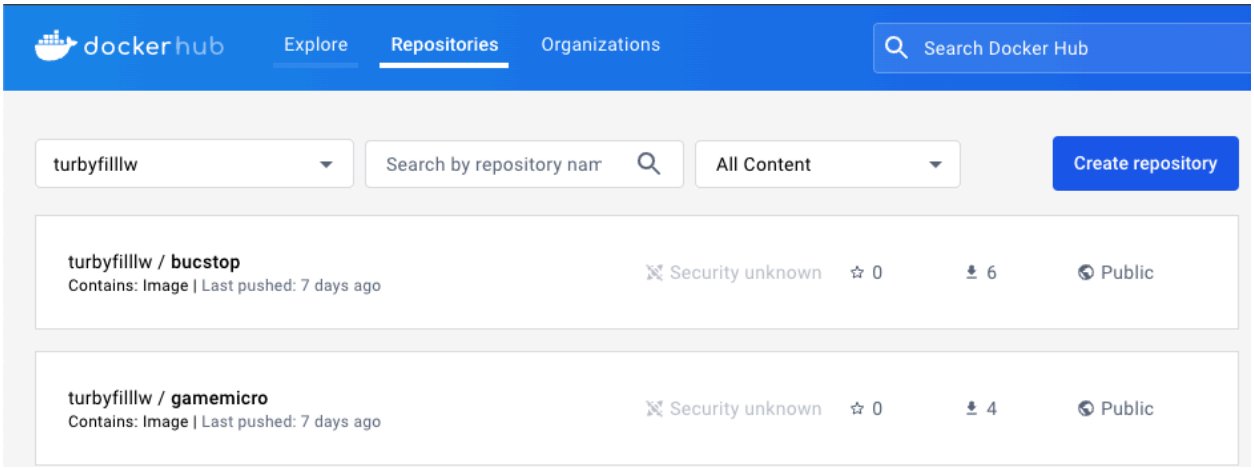
Some things needed:

1. A docker account
2. An AWS account (I am using a student account from system administration).

Docker account setup

1. Create an account on docker.com and download docker desktop.

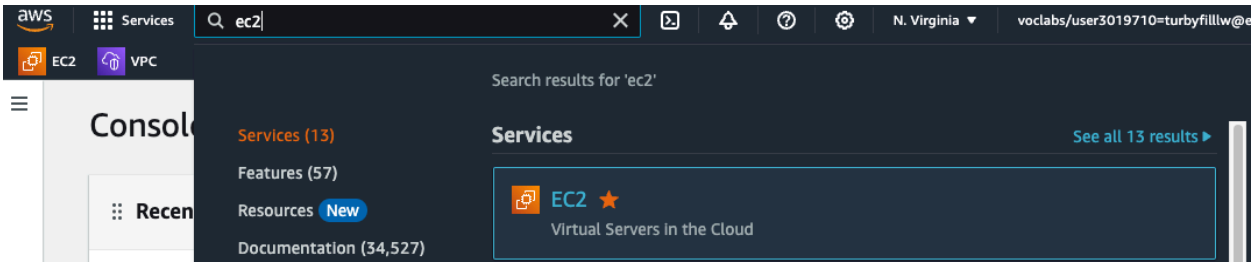
2. On the website hub.docker.com, create two repositories. (I recommend naming them bucstop and gamemicro)



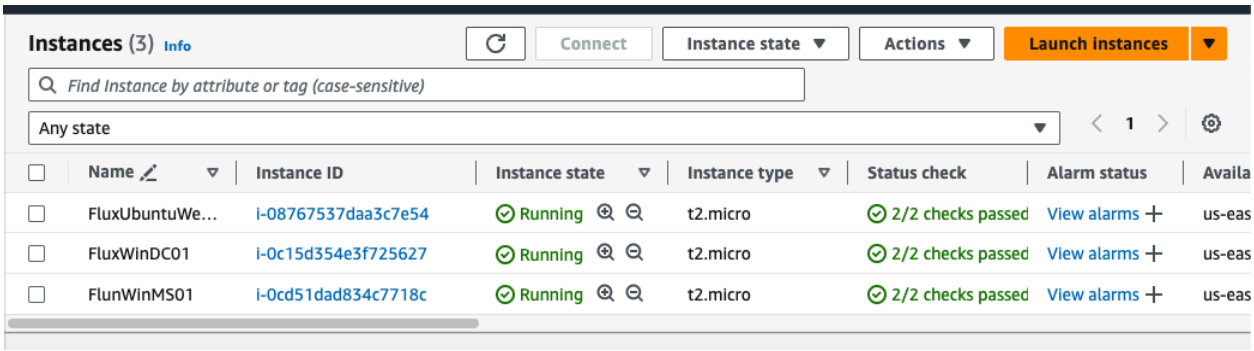
AWS EC2 instance

As mentioned before, I am using a student account from system administration. You are able to create a free account, but you must input a credit card.

1. Search EC2 in the search bar



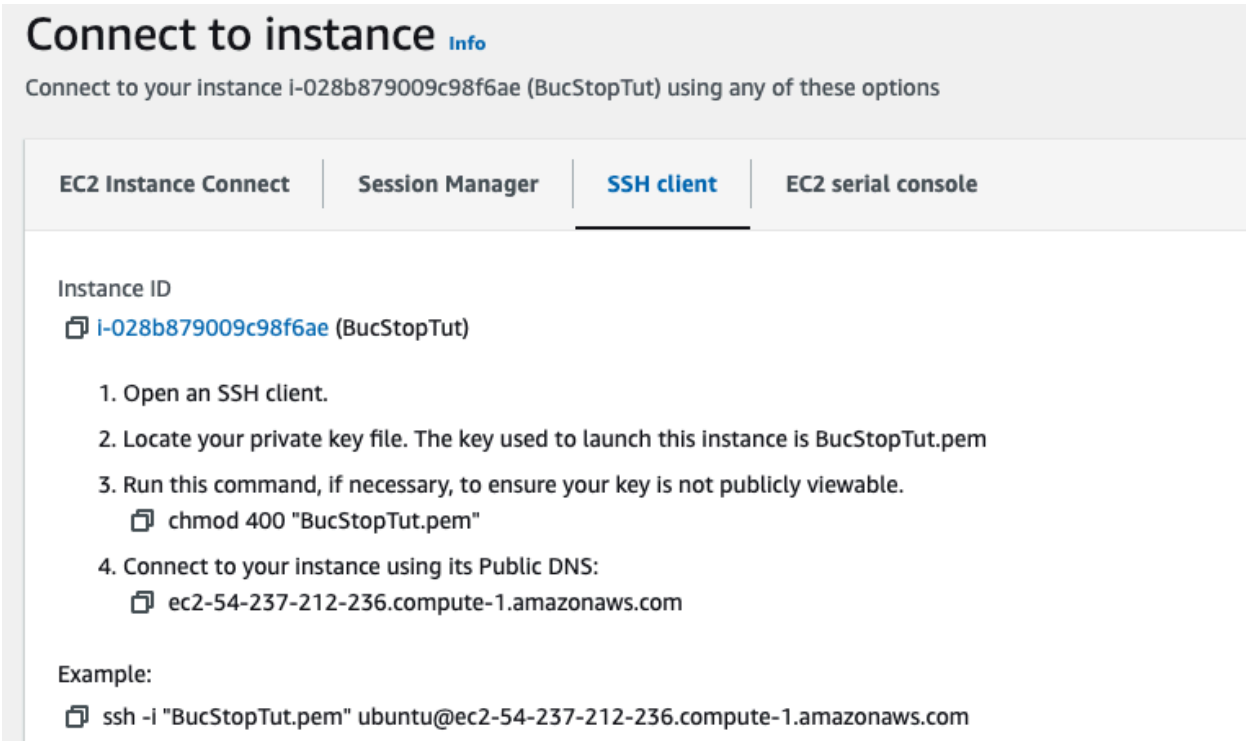
2. Once in the EC2 dashboard, click on Instances from the menu on the left side.
3. Select launch instances on the top right of the instances dashboard.



4. Creating an instance:
    a. You can name the instance whatever you desire.
    b. Select Ubuntu for the OS image.
    c. The default image and instance type is fine for our use case.
    d. Create a new .pem key pair.

e. Allow HTTPS and HTTP traffic from the internet.

5. When it is created, select your instance from the list and click connect.
   a. Switch over to the SSH client tab



   b. Open your terminal and cd to the directory that your public key is saved in. Run this command to ensure your key is not publicly viewable “chmod 400 (yourkeyname).pem".
   c. Copy the command from the example as seen in the screenshot above. “ssh -i "BucStopTut.pem" ubuntu@ec2-54-237-212-236.compute-1.amazonaws.com”

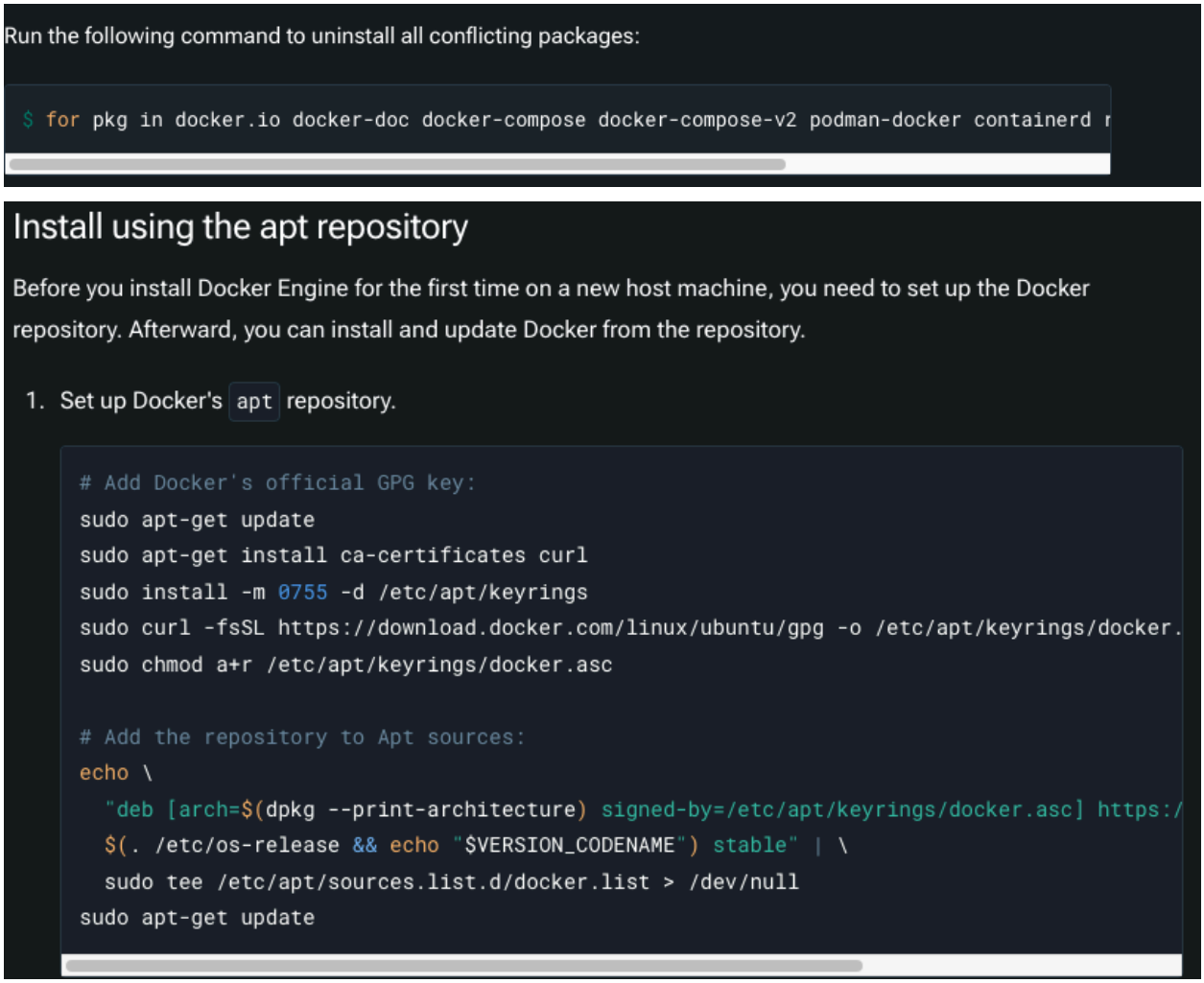
Within the instance:

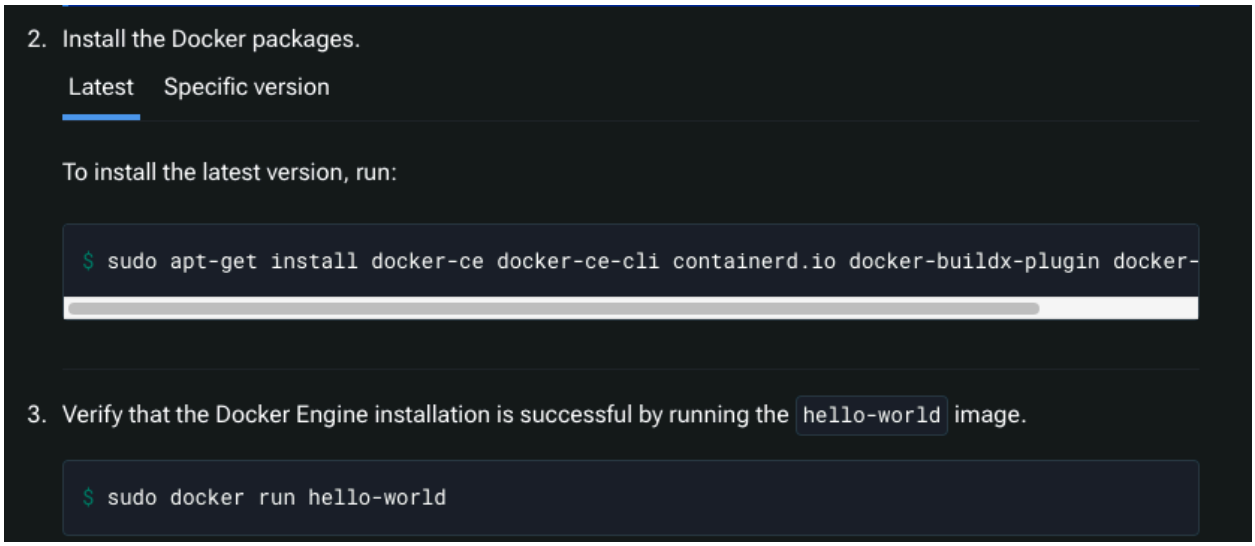
1. Run the command “sudo adduser (username)”.
2. Add your user account to the sudoers group by running “sudo usermod -G sudo (username)”.
3. Switch to your user account by running “su (username)” and cd into your home directory (cd /home).
4. Make a directory called scripts (sudo mkdir scripts).
5. cd into your scripts directory.

6. Run the command “sudo nano docker-compose.yml" and copy and paste the contents of the docker-compose.yml from the scripts folder in the bucstop github. (There are more technical ways to do this, but for our use case this is fine.)
    a. Replace image: $DOCKER-REPO-PATH/bucstop with your username and the name of your repo. Do this for all occurrences.
    b. Repeat this process for firstDeploy.sh, redeployBucStop.sh, and redeployMicro.sh.

Install Docker on your instance

We will be running the commands on this link:
https://docs.docker.com/engine/install/ubuntu/

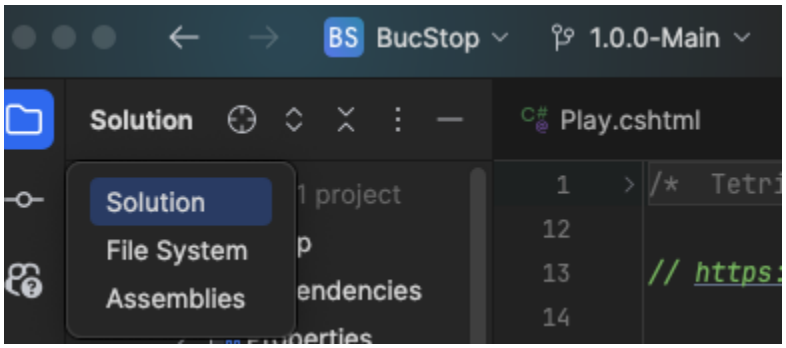
1. Type cd to change your directory to home.
   Run these commands from the above link
2. 

3. 

In your IDE:

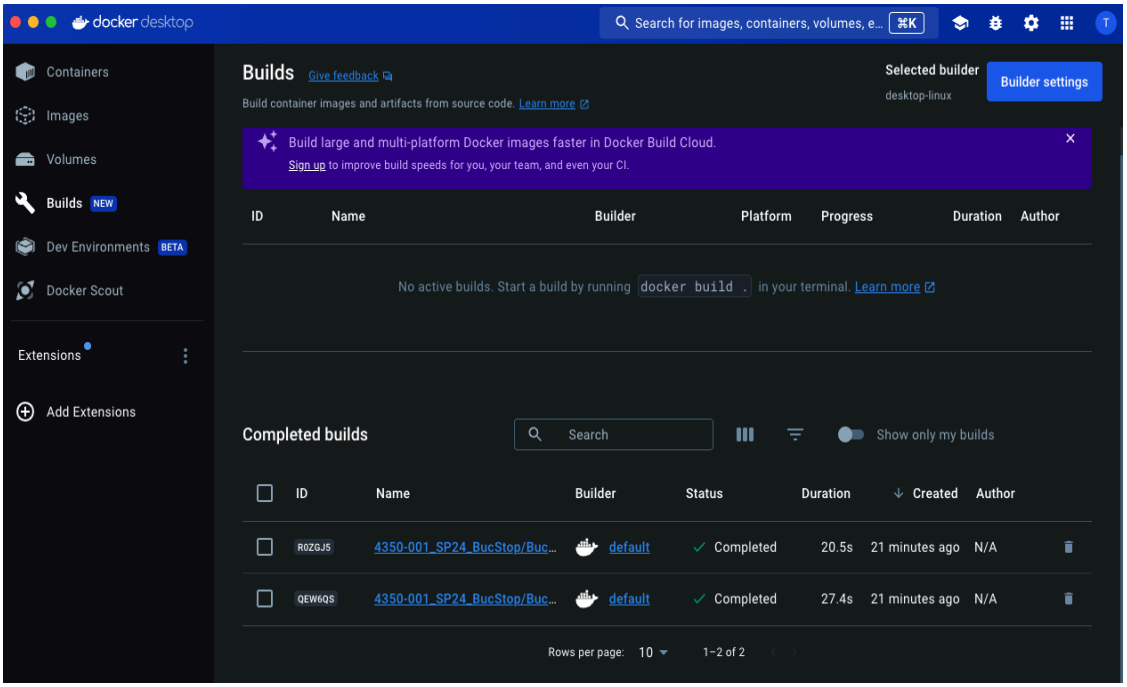
I personally use Rider (free for students) because Visual Studio support has been discontinued. This should work for Visual Studio as well though.

1. If you use rider: In the top left change solution to file system. In visual studio, you should not have to change anything.



   a. Open the terminal from within rider and run the command (Be sure to copy everything, including the period at the end or the command will not run): **docker build -f BucStop/Dockerfile -t DOCKER_USERNAME/DOCKER_REPO --platform linux/amd64 .**
      i. Replace DOCKER_USERNAME with your docker username and DOCKER_REPO with the name of your docker repo.
      ii. Afterwards, run the command "**docker push DOCKER_USERNAME/DOCKER_REPO**"

iii. If done successfully you should see your progress in docker desktop



2. Repeat for the microservice repo, but change the command to **docker build -f GameMicroServer/Dockerfile -t DOCKER_USERNAME/DOCKER_REPO --platform linux/amd64 .** and **docker push DOCKER_USERNAME/DOCKER_REPO**

Deployment!

1. Run cd /home and then cd /scripts to get back into your scripts directory on your instance.
2. Run sudo chmod +x firstDeploy.sh to make the script executable.
3. Run ./firstDeploy.sh
4. Now type the public ip address of your instance into your browser and bucstop should be deployed!.