Name: Kushagra Bansal

Roll\_No: 116

Sap\_ID: 500067414

**Experiment No. 12**

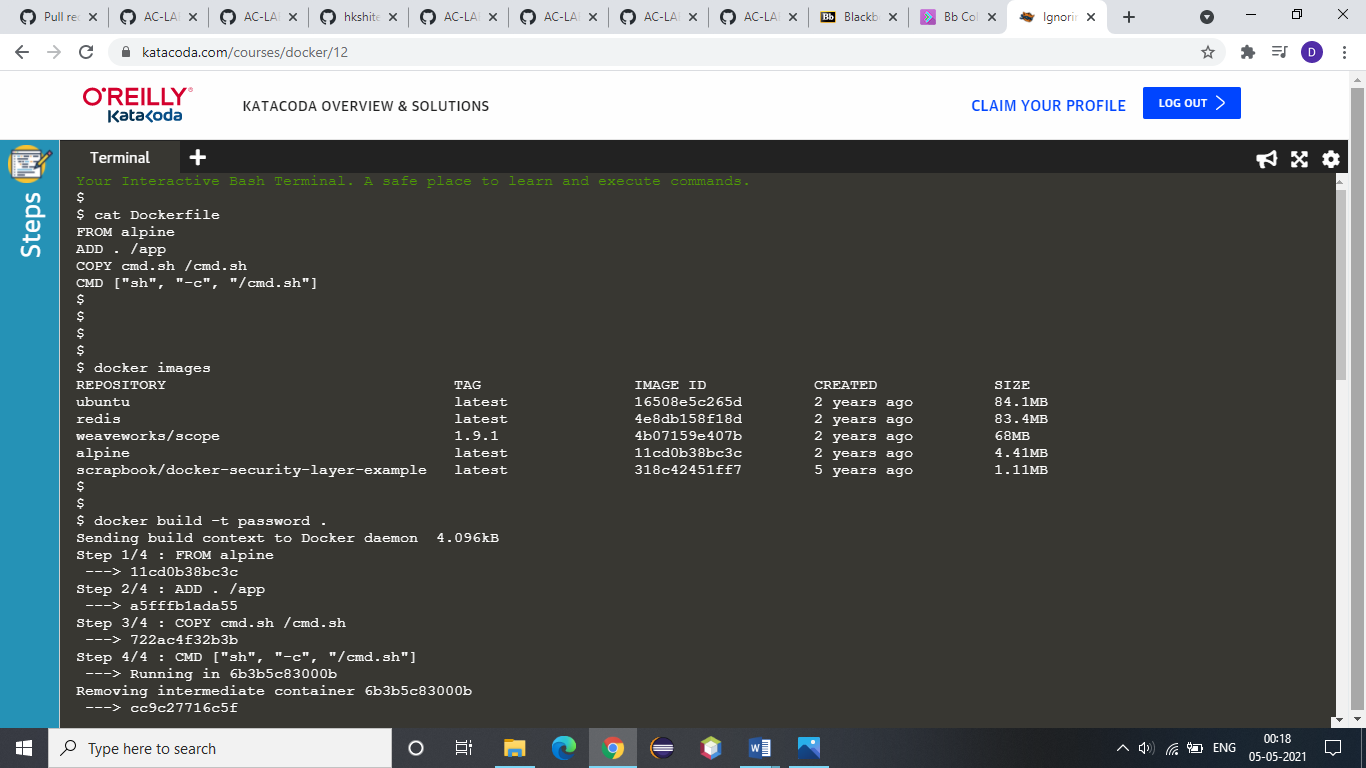
**Build Docker Image using .dockerignore file**

In this experiment, we are going to use a concept similar to .gitignore which is called .dockerignore file. It allows us to ignore some sensitive, important or large files from being included in images.

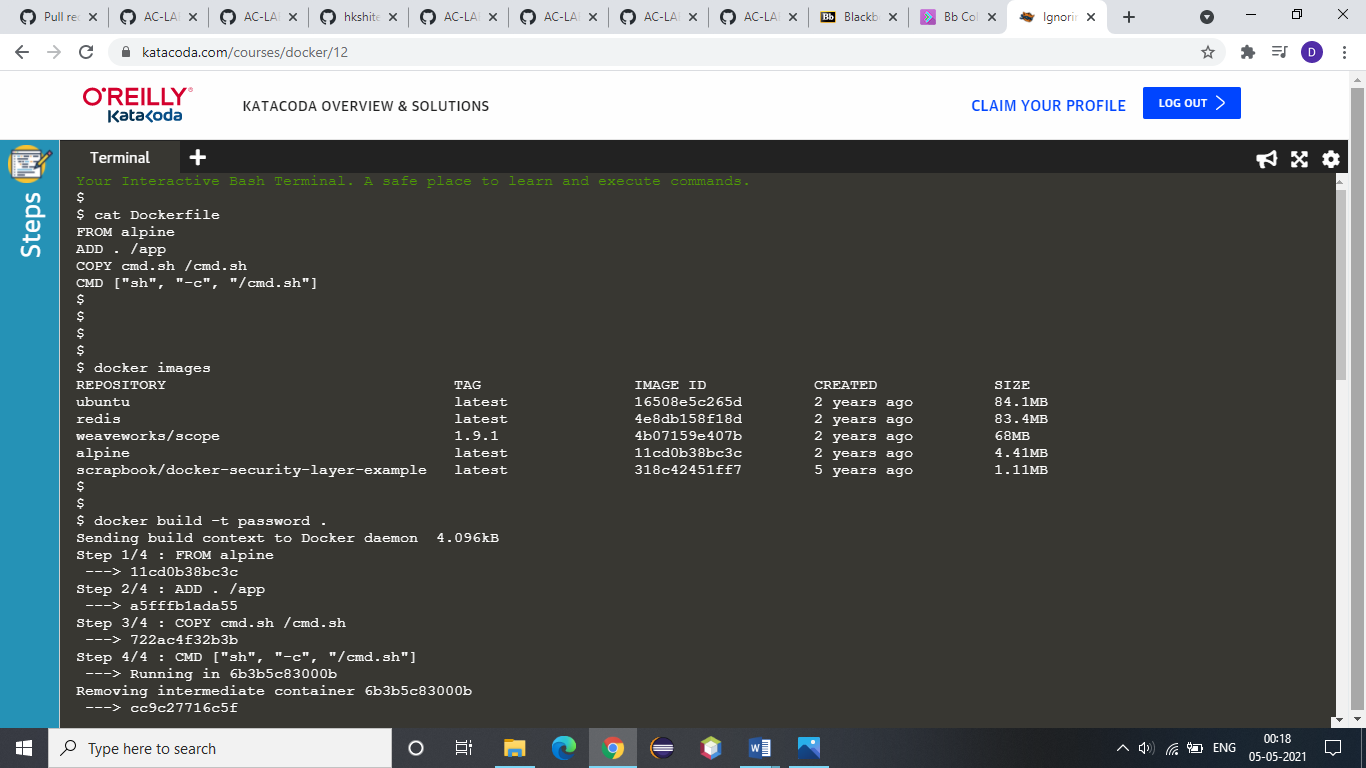
To prevent sensitive files or directories from being included by mistake in images, you can add a file named *.dockerignore*.

**Follow these steps below:**

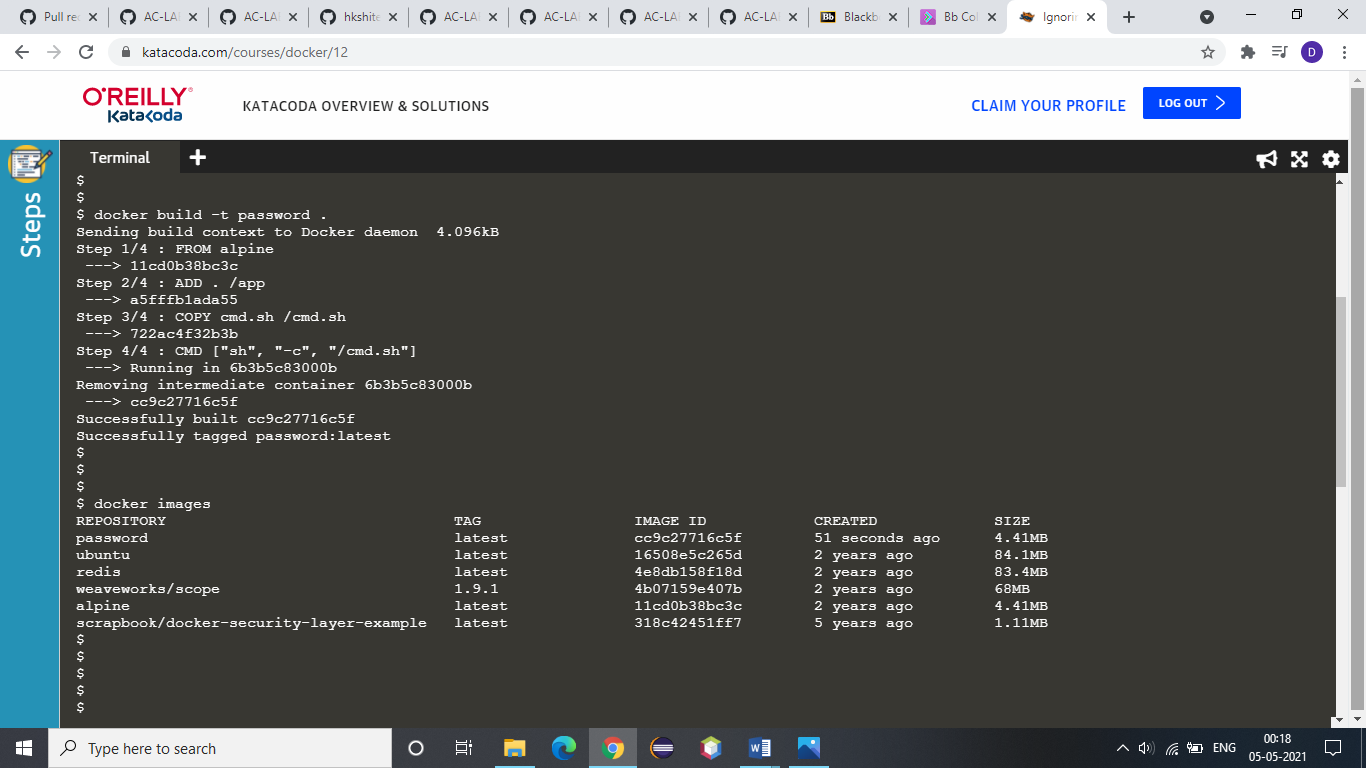
1. Create a dockerfile which actually copies the working directory into the Docker Image. As a result, this would include potentially sensitive information such as a passwords file which we'd want to manage outside the image.



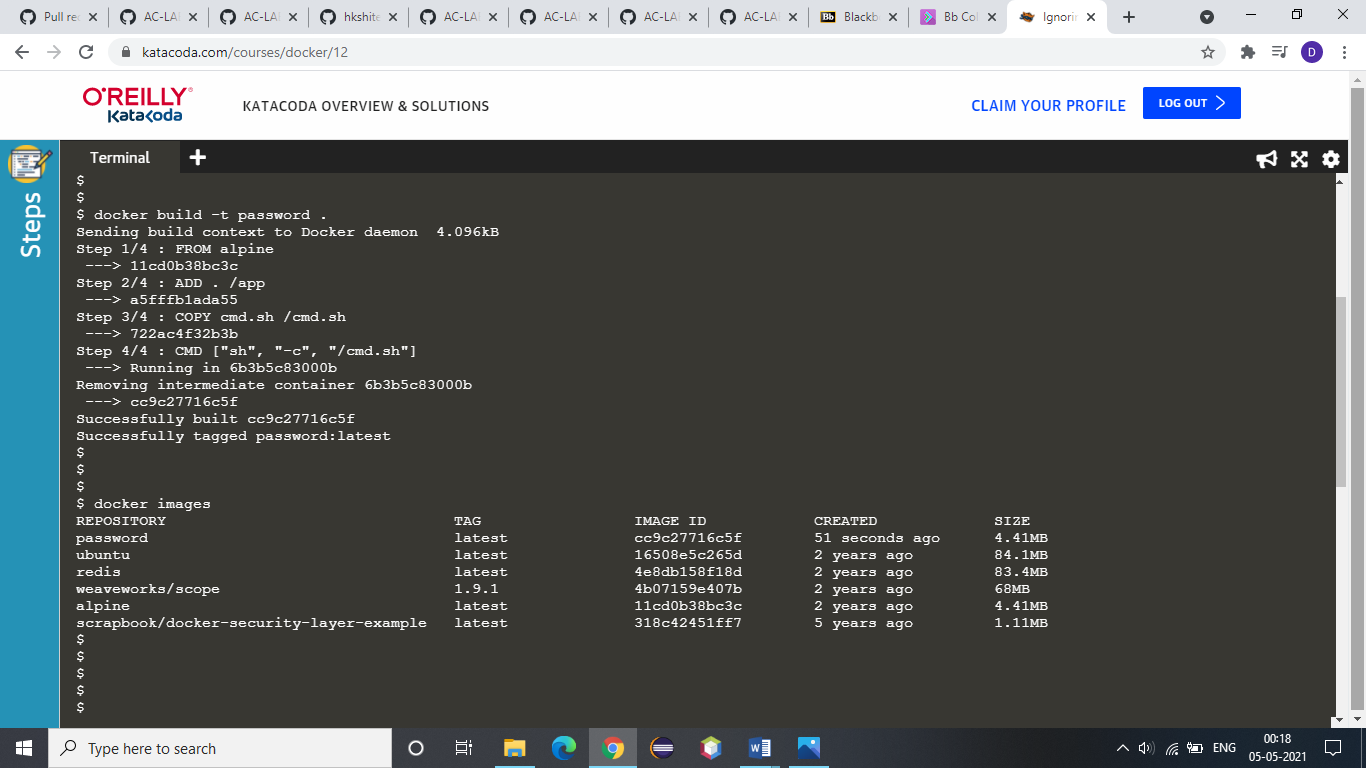
2. Check for the existing images first:



3. Build the image using the following command:

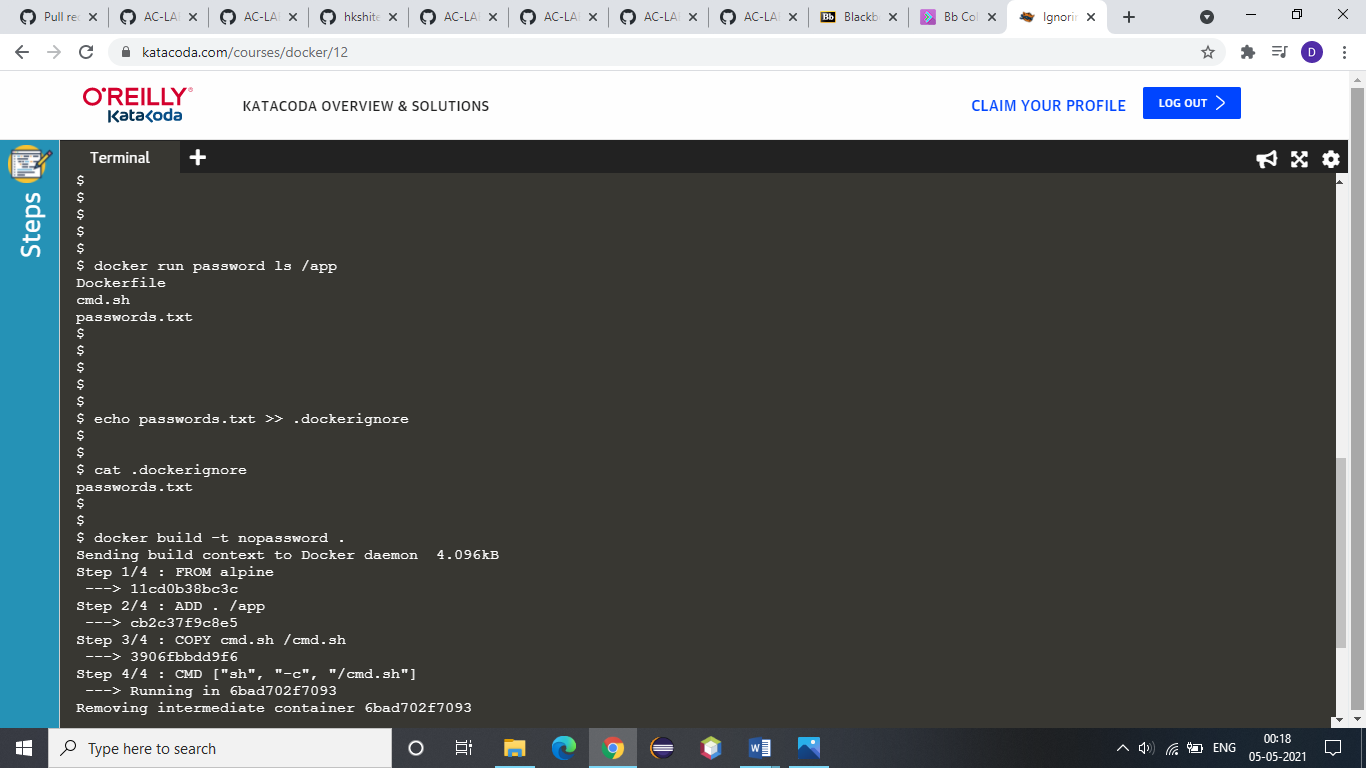


4. Now, check that the newly created image is build or not!

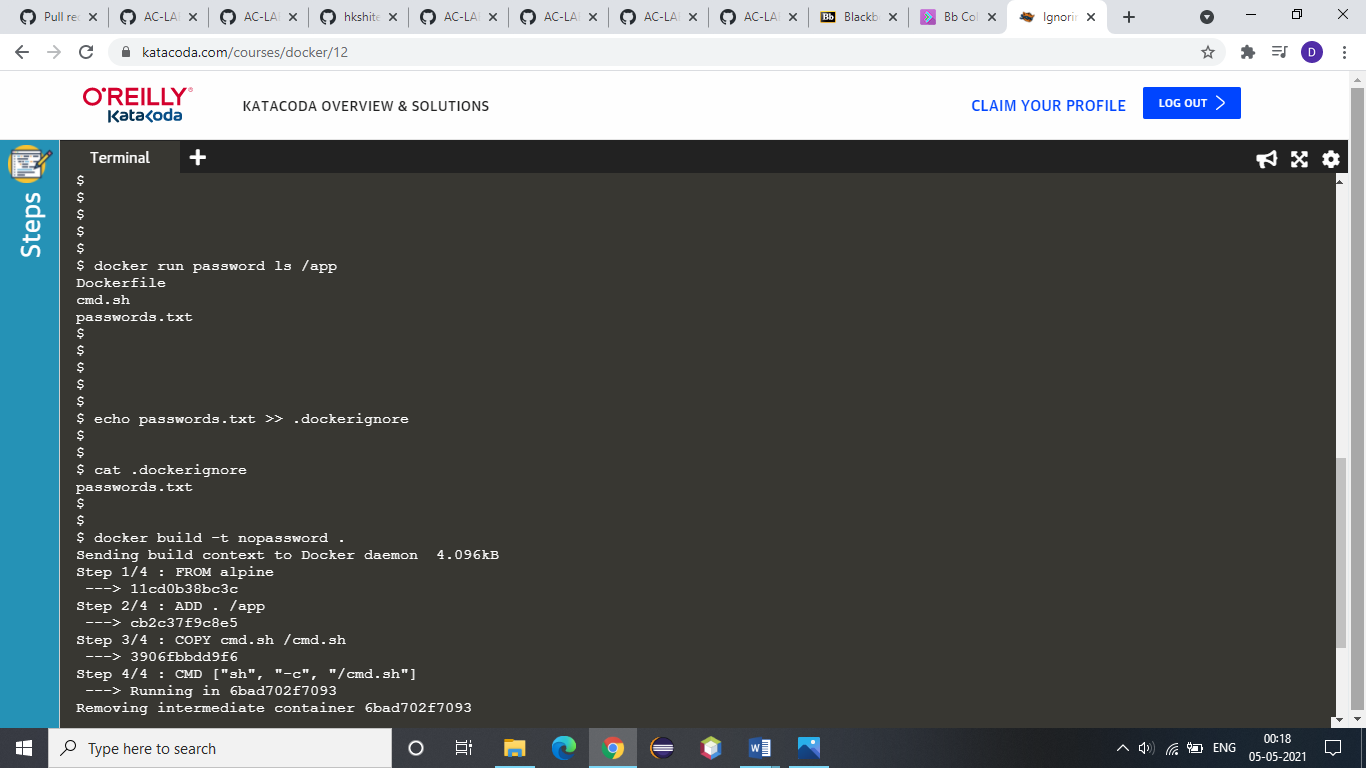


5. Look at the output using **docker run password ls /app**

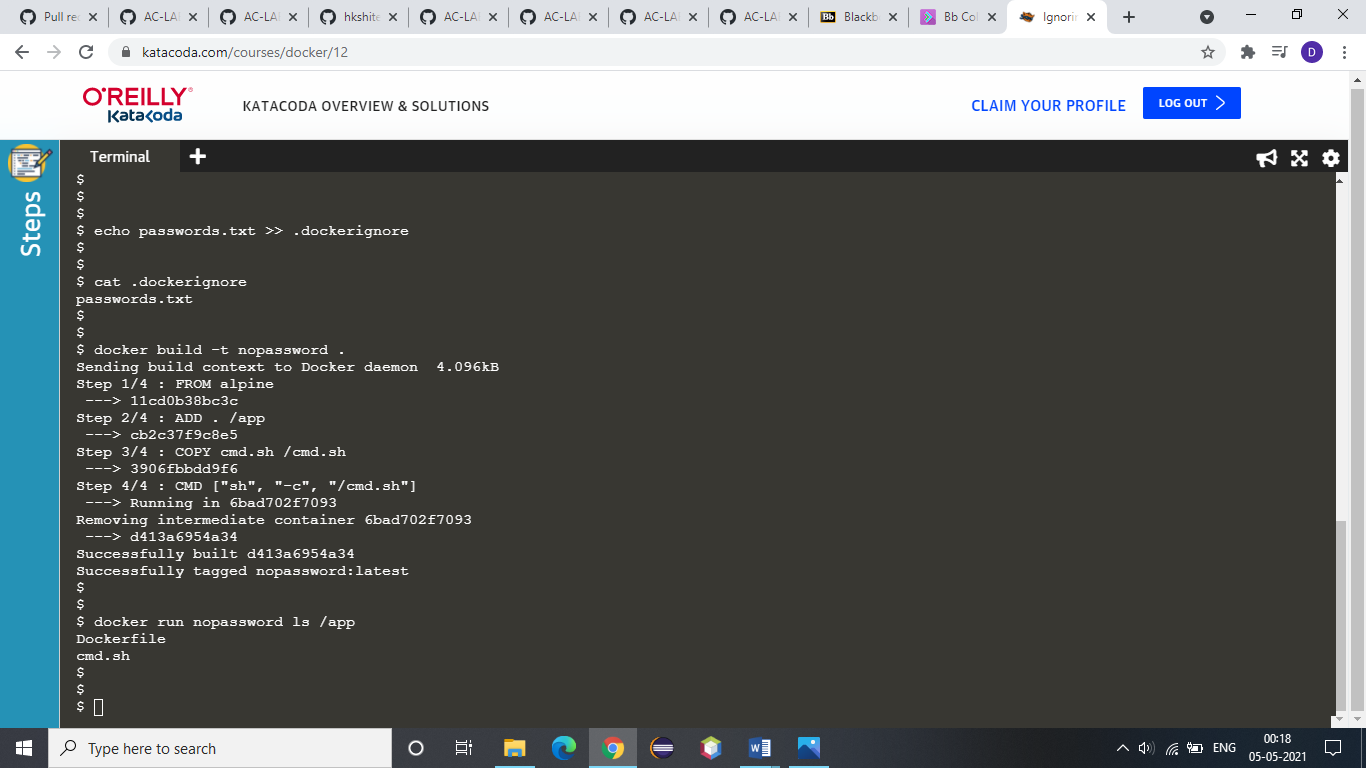
This will include the passwords file.



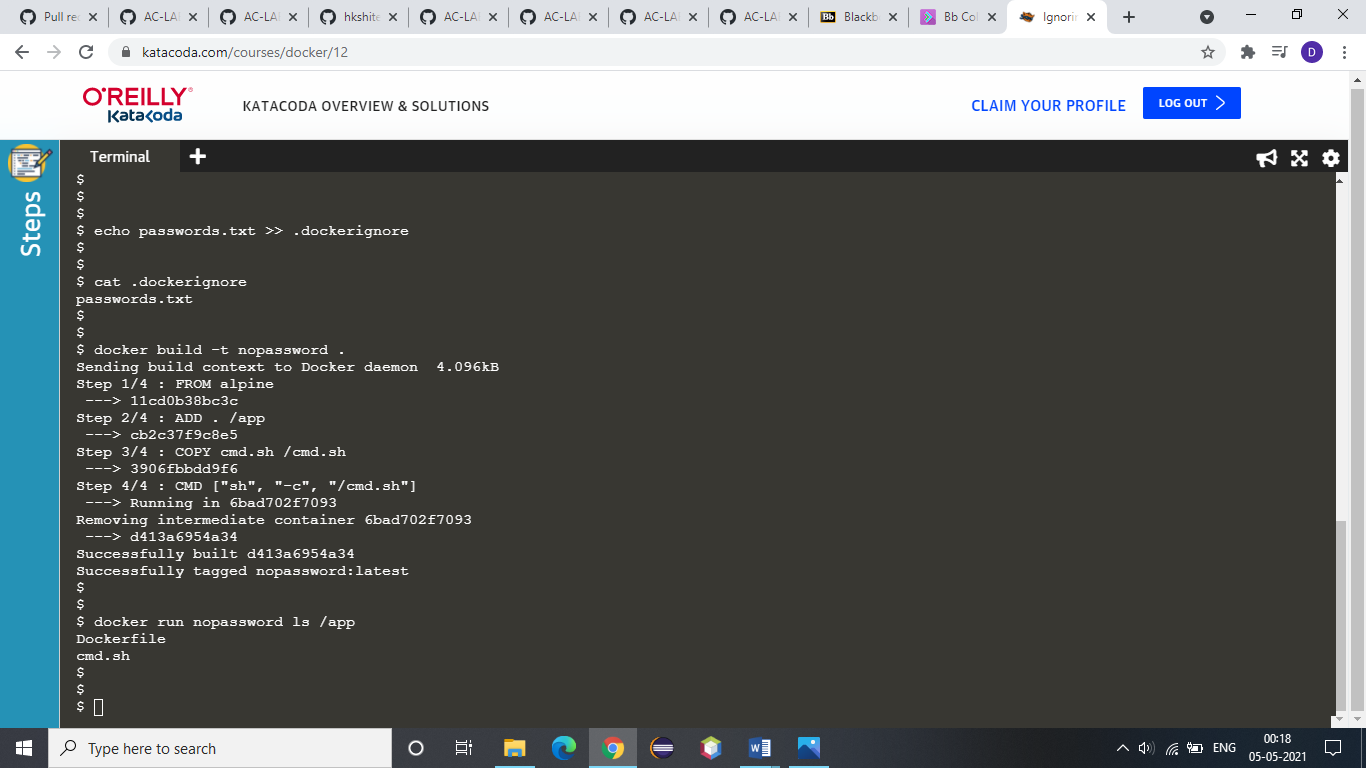
6. Now, we are going to use a command which would include passwords.txt in our *.dockerignore* file and ensure that it didn't accidentally end up in a container. The *.dockerignore* file would be stored in source control and share with the team to ensure that everyone is consistent.



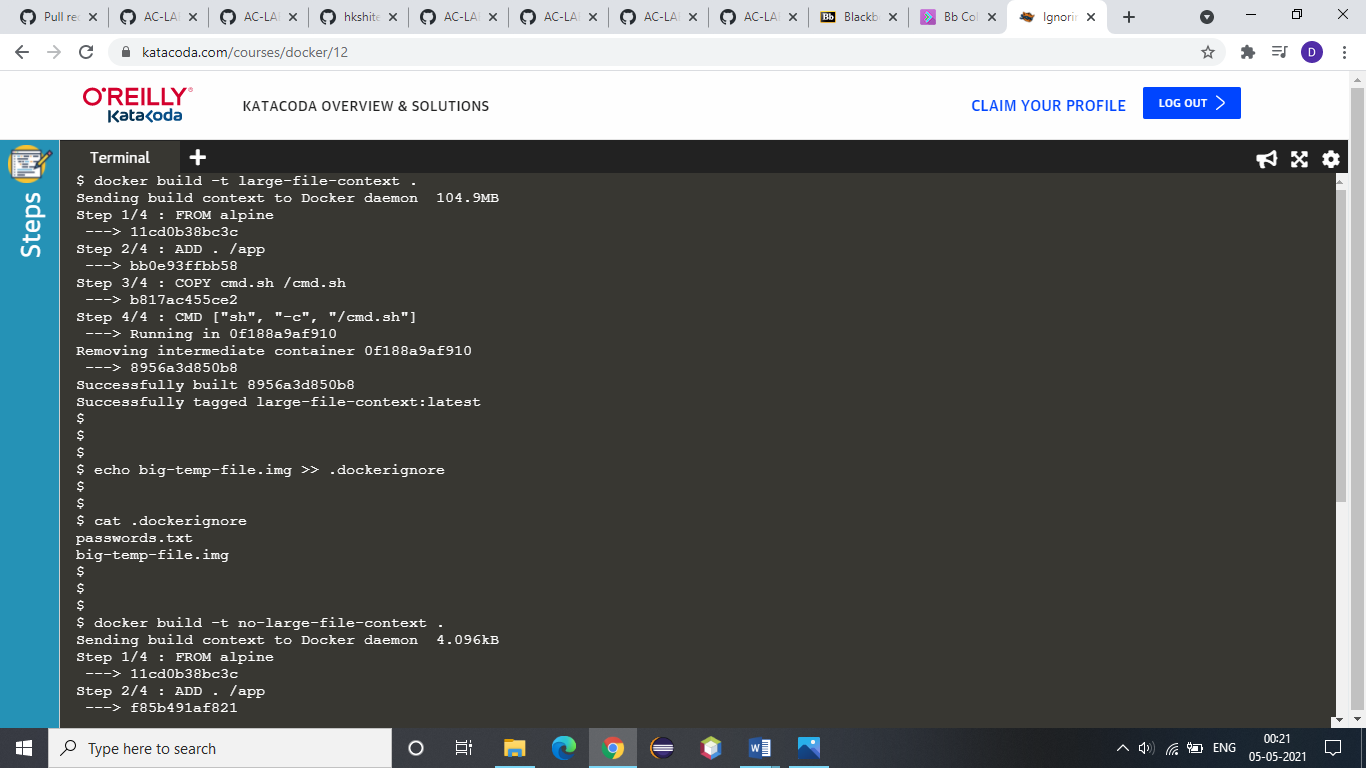
7. Now, build the image, because of the Docker Ignore file it shouldn't include the passwords file.



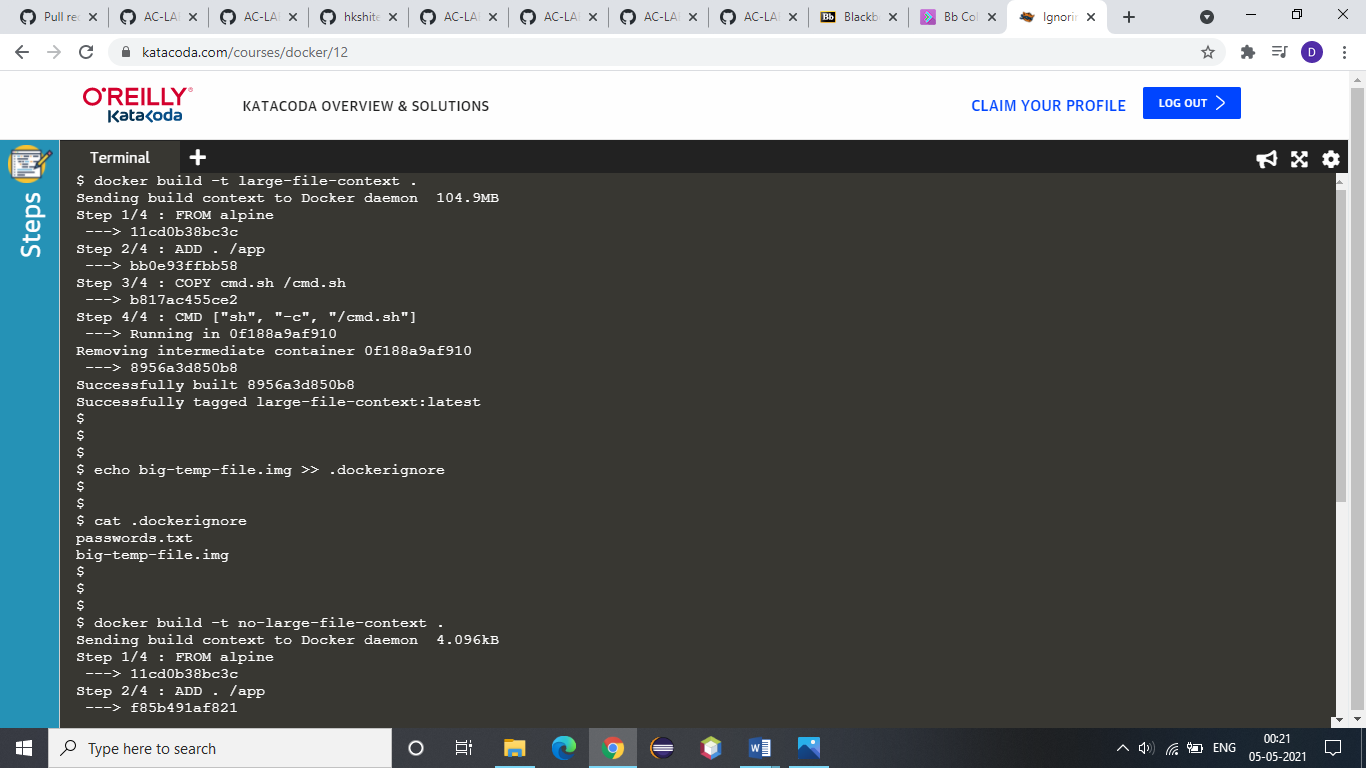
8. Look at the output using:



9. Now, we are creating a 100M file which is never used the dockerfile. Now, create a build which will include this large file and notice its impact on build performance:



10. Now, to optimize the build, to speed up our build, simply include the filename of the large file in the ignore file.



11. When we rebuild the image, it will be much faster as it doesn't have to copy the 100M file.

