Hi Juan,  
  
It's ok. Don't worry. That's why I always have some "ideas" as Plan B in   
my hand.  
  
Based on your previous work on local microservice implementation, I   
believe that the so-called "imaging database updates" project suits you   
well. There are three small milestones to reach:  
1. Prepare an immediately functional database image for use.  
The logic is exactly the same to this post:   
<https://dev.to/umeshdhakar/how-to-build-a-docker-image-which-can-restore-databases-whenever-a-new-container-is-created-from-it-dbm>  
But you should use MySQL to implement this, so as to match/reuse your   
previous practice. The secret is the Dockerfile and the .sh scripts. You   
may refer to our last lecture video for more explanations.  
  
2. Update the database image.  
Try to make some updates in the database. The changes should be   
serialized/merged into the .sh scripts. Please also think about changing   
the database's user name and password to match your application's needs.   
Rebuild the image. Then all the updated will be immediately functional   
when the new image-based container is launched.  
  
3. Use your two-container microservice to verify the functionality.  
You can simply reuse your previous work as a demo, say, the student name   
lookup system. By preparing the functional database in advance, your   
microservice should be able to look up students with the existing data   
by default. After updating the database images (say, adding some new   
students), the microservice will be able to look up new students.  
  
In other words, this project is to replace the live database update with   
static database update for containers. And we will have good reasons to   
justify the complexity in this student project for the final evaluation.   
Please do NOT forget to document your step-by-step work process,   
including troubleshooting details.  
  
By the way, please do not tell the others about using the open-idea   
project to waive our final examination yet. Because we will need to see   
the whole class's performance (who knows if examination will still be   
needed for some of you).  
  
Anyway, I strongly suggest working on your project ASAP. Do not wait   
until the last minute. Feel free to let me know if you have any   
questions or concerns.  
  
Best Regards,  
Zheng

Hi Juan,  
  
I think I made a mistake about the tutorial reference:   
<https://dev.to/umeshdhakar/how-to-build-a-docker-image-which-can-restore-databases-whenever-a-new-container-is-created-from-it-dbm>  
Our goal is not like that, but purely preparing the data in the image.   
Sorry for my previous misleading. To compensate for this mistake, I   
managed to borrow one Linux machine and cleaned my own laptop to try the   
database part (Milestone 1&2).  
  
The Dockerfile and the relevant source files are shared at   
<https://drive.google.com/drive/folders/1mGh-jFB6K4r23aodtK2rL6-F4IxPR3VA?usp=sharing>  
You can download all of them and please read the Dockerfile first, and   
then you will understand what the other files are.  
Then you can use them to try build an image and launch a container to   
see the effects of the so-called imaging data. (There might be debates   
here, that's why we do this work.)  
  
I also shared the available image at   
<https://drive.google.com/drive/folders/1B_3RlbmdfWwNTcJAHGBbr3-REb2ynPuS?usp=sharing>  
We can copy this image to a new machine and reload it and make it into a   
container, by following   
<https://stackoverflow.com/questions/23935141/how-to-copy-docker-images-from-one-host-to-another-without-using-a-repository>  
  
To make this trial, I cleaned my laptop with Windows 10. I guess my   
experience could also be useful for you to replicate if your machine is   
also with Windows 10. If so, please follow   
<https://hackernoon.com/how-to-run-docker-linux-containers-natively-on-windows-ti1i3uxr>  
If you cannot successfully install the Linux subsystem, you will   
probably have to enable virtualization in BIOS:   
<https://stackoverflow.com/questions/62340566/fix-wslregisterdistribution-failed-with-error-0x80370102>  
If you are able to go through everything, you should also be able to   
directly reuse my image.  
  
Anyway, I hope this can save you some time. Do not forget that this is   
just half work of this project. Please feel free to modify the source   
files and make the database become a student information database, and   
eventually supplement source code to make the two-container   
microservice. You may have noticed that I have not shown you the correct   
documentation work. You will need to go through the database part as   
well, so I leave this documentation job to you.  
  
Feel free to let me know if you have any questions.  
  
Best Regards,  
Zheng