IST769 Homework Submission

Name: **Sathish Kumar Rajendiran**
SUID: **666555028**
Email: **srajendi@syr.edu**
Date Due: **07/13/2021**
Homework #:**1**

Exercise(s):

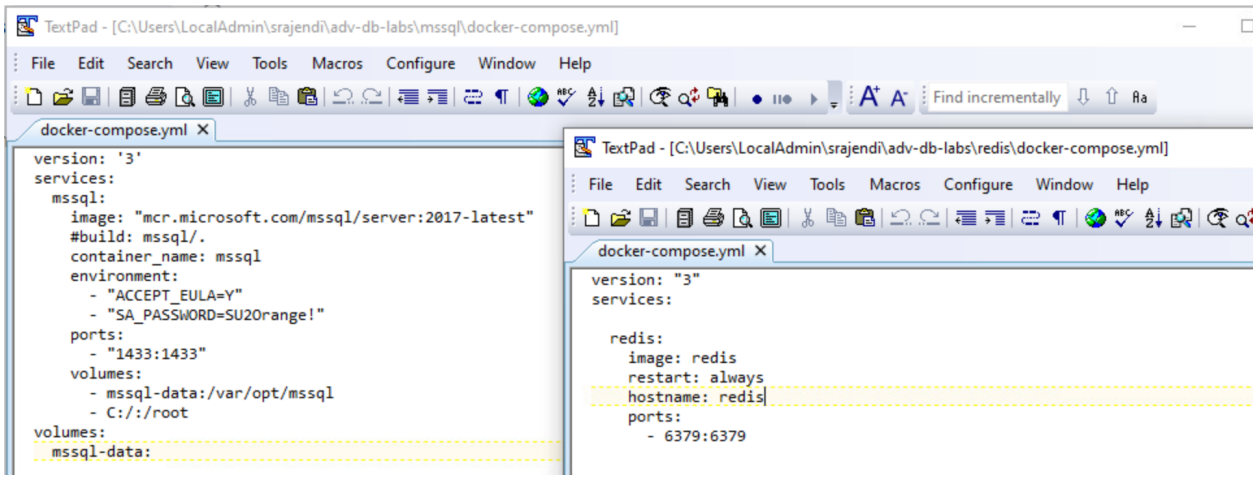
1**. What would be the command to bring up the redis environment? How is the command different from the mssql environment? How is it the same?**

The command used to bring up the redis environment was "**docker-compose up -d".** I was able to start the Redis instance on Docker container following the same. However, it has to be executed having the directory under **/srajendi/adv-db-labs/redis.** This command created a network for "redis_default" and it was running on port 6379.It used the image from docker compose.

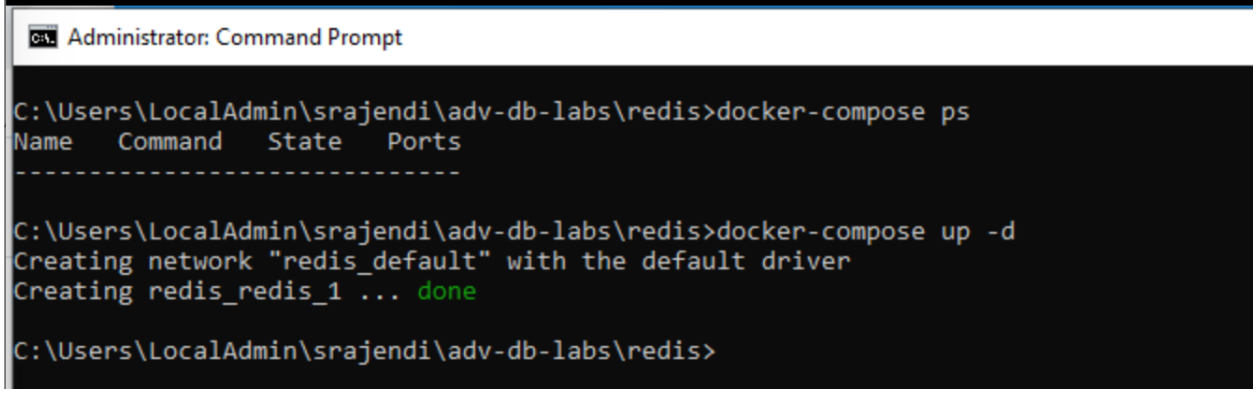
on MSSQL, command is also the same but the directory under the execution was on **/srajendi/adv-db-labs/mssql.** In addition, mssql image was pulled from the yaml file and built the image from that repository. This reference is updated on the yml file. Later, this command created a network for "mssql_default" and it was running on default port 1433.

I was also able to run both services on the docker container.

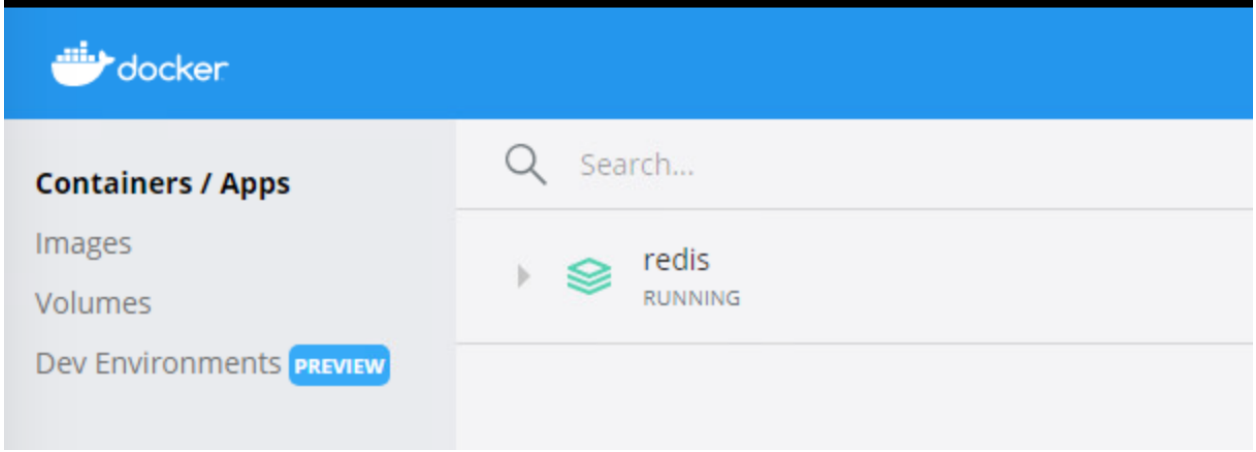
Evidence:

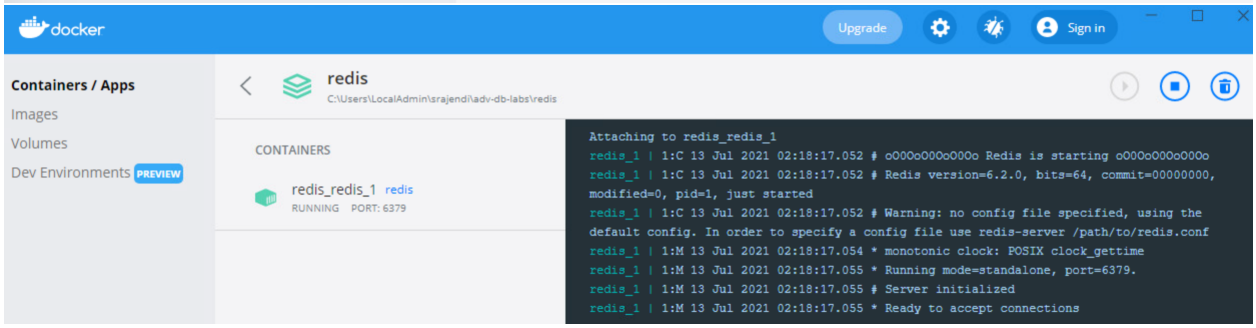
- 

- **769-Win10Docker-srajendi**



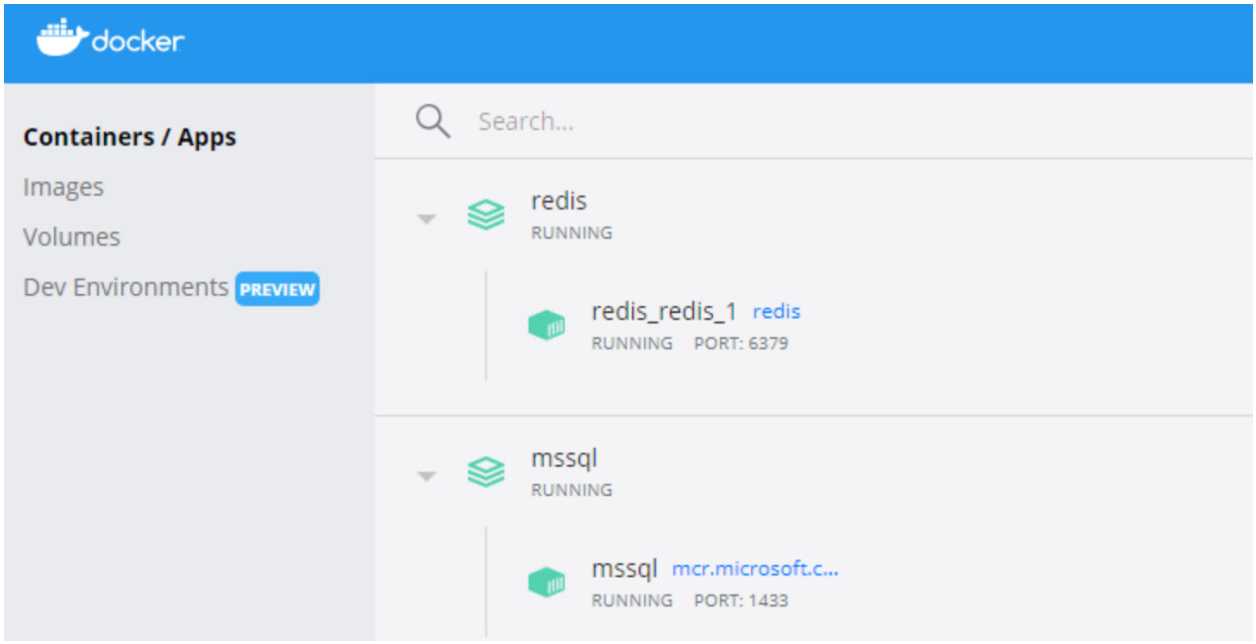
- **769-Win10Docker-srajendi**



- 

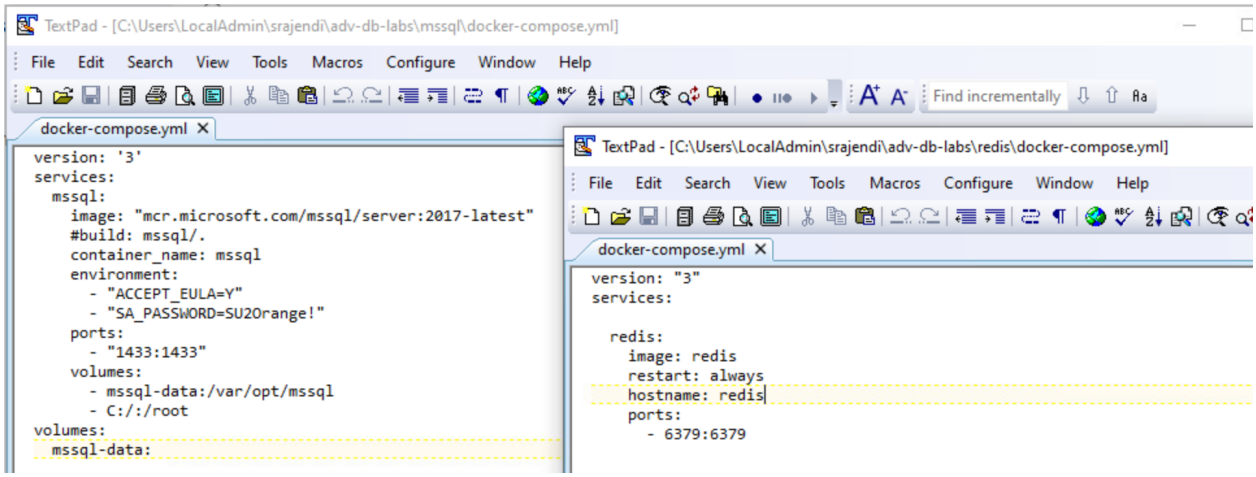
```
C:\Users\LocalAdmin\srajendi\adv-db-labs\mssql>docker-compose ps
Name   Command   State   Ports
------------------------------

C:\Users\LocalAdmin\srajendi\adv-db-labs\mssql>docker-compose up -d
Pulling mssql (mcr.microsoft.com/mssql/server:2017-latest)...
2017-latest: Pulling from mssql/server
61e03ba1d414: Pull complete
4afb39f216bd: Pull complete
e489abdc9f90: Pull complete
999fff7bcc24: Pull complete
ef0b62888e99: Pull complete
9464681f0202: Pull complete
884f6b75bc0d: Pull complete
aea6c44dc7a9: Pull complete
Digest: sha256:046fdb8988b92a9da973d2c4a237a769afee9010d79d76dd807b930a3d0e56e9
Status: Downloaded newer image for mcr.microsoft.com/mssql/server:2017-latest
Creating mssql ... done

C:\Users\LocalAdmin\srajendi\adv-db-labs\mssql>
```

```
C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>cd..

C:\Users\LocalAdmin\srajendi\adv-db-labs>cd mssql

C:\Users\LocalAdmin\srajendi\adv-db-labs\mssql>docker-compose ps
Name   Command   State   Ports
------------------------------

C:\Users\LocalAdmin\srajendi\adv-db-labs\mssql>docker-compose up -d
Creating network "mssql_default" with the default driver
Creating mssql ... done

C:\Users\LocalAdmin\srajendi\adv-db-labs\mssql>docker-compose ps
Name               Command               State                    Ports
-------------------------------------------------------------------------------------------
mssql   /opt/mssql/bin/nonroot_msg ...   Up      0.0.0.0:1433->1433/tcp,:::1433->1433/tcp

C:\Users\LocalAdmin\srajendi\adv-db-labs\mssql>
```

- 

2**. Where is the specific configuration information about each environment stored?**

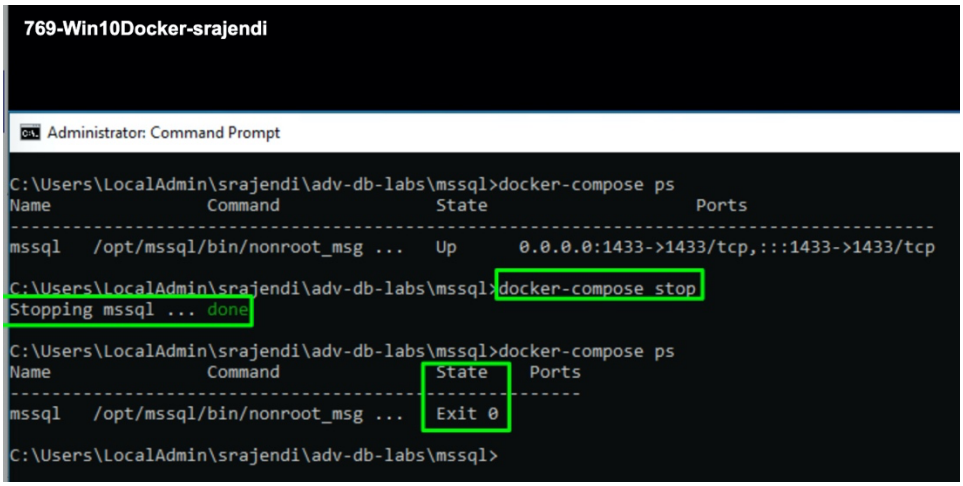
Configuration information about each environment is stored in "**docker-compose.yml**" file under respective components/services folder.

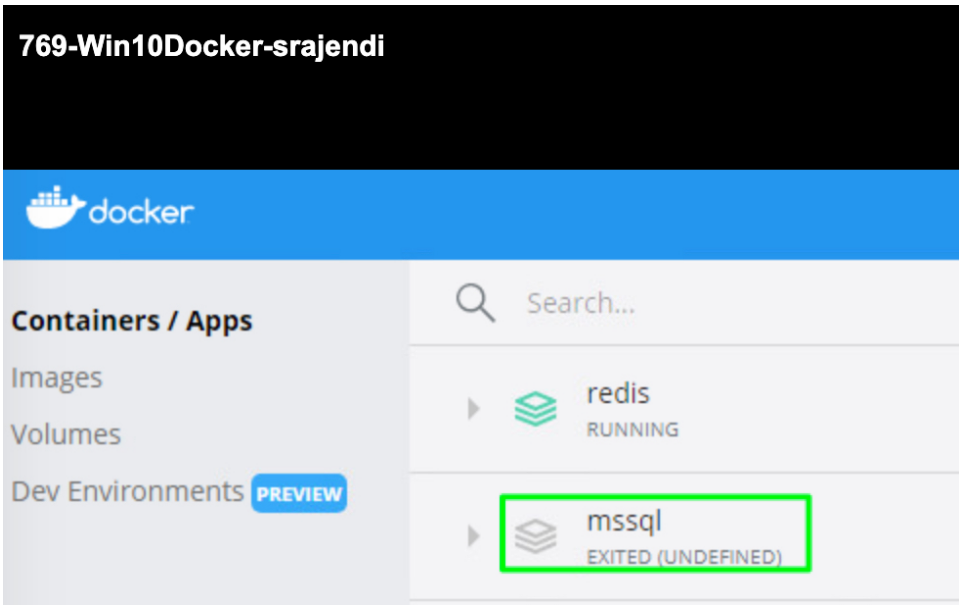
- 

3. Explain the difference between stopping an environment and bringing it down. Elaborate with use-cases for each.

- Stopping an environment:
  - Docker command: **docker-compose stop**; Whenever the services are up, executing a **stop** command would bring the services to halt state and the resources allocated would

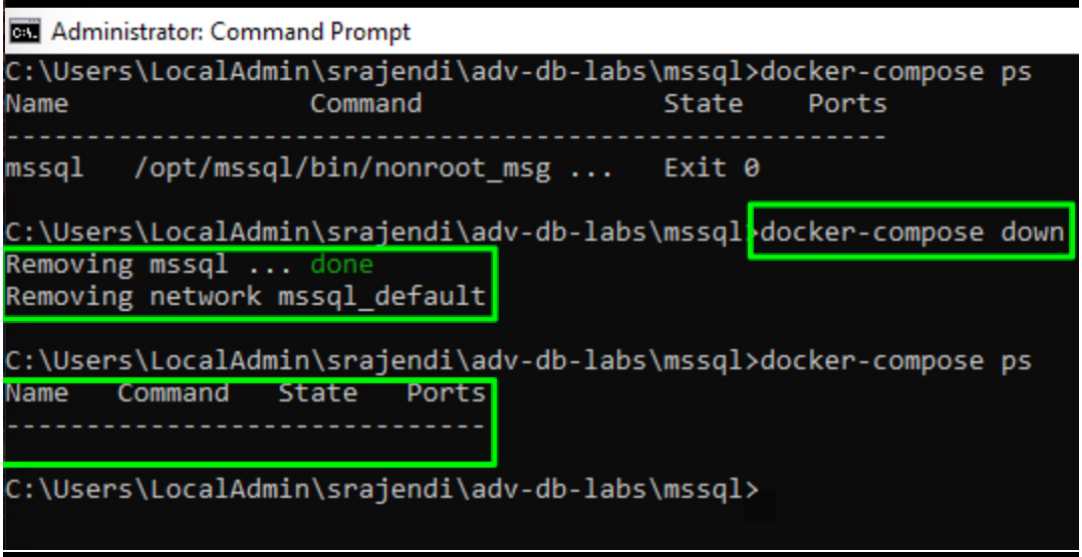
be released immediately. However, the data collected, and components created are still associated to that image and can be resumed later. This would also update the current state of the service to “Exit 0”. i.e., graciously stopping the service.

- 

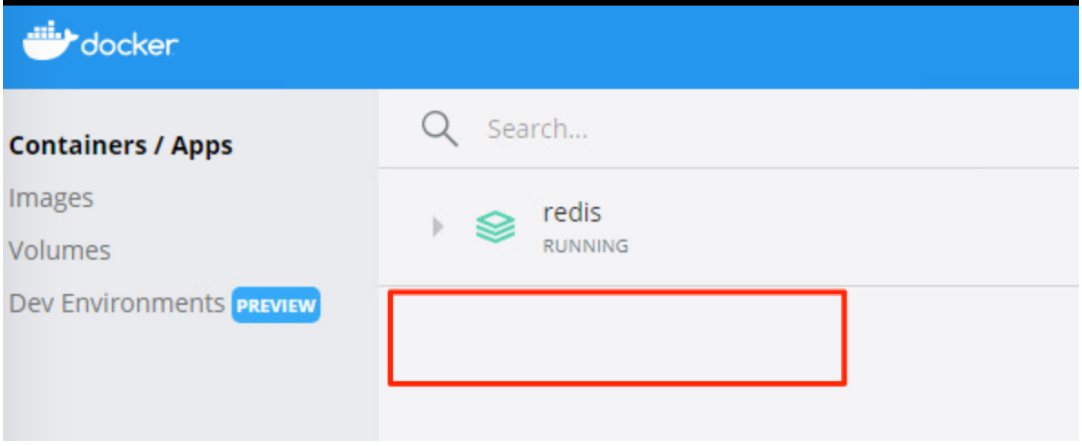
- 

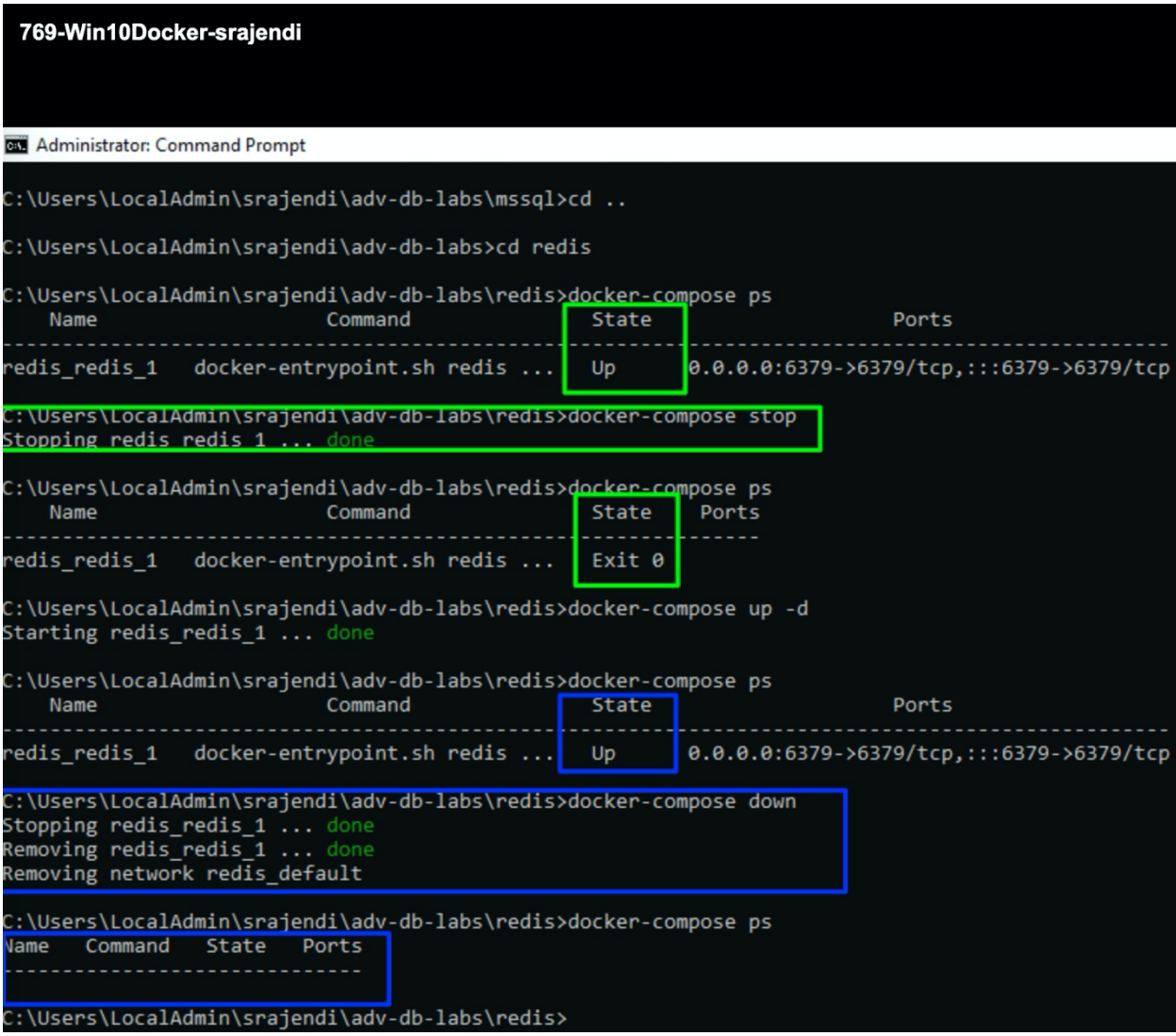
- bringing it down: **docker-compose down**
  - Whenever the services are stopped, executing a “down” command would remove the image and default network created on that port immediately. It’s a hard delete of the docker image and all the data would be lost with a successful execution. There won’t be any information on the current status if we execute the “ps” command.
  - On another hand, if we execute “down” command while having the service in running mode, it does 3 steps
    - stopping the service
    - removing the image
    - removing the network created.

- **769-Win10Docker-srajendi**



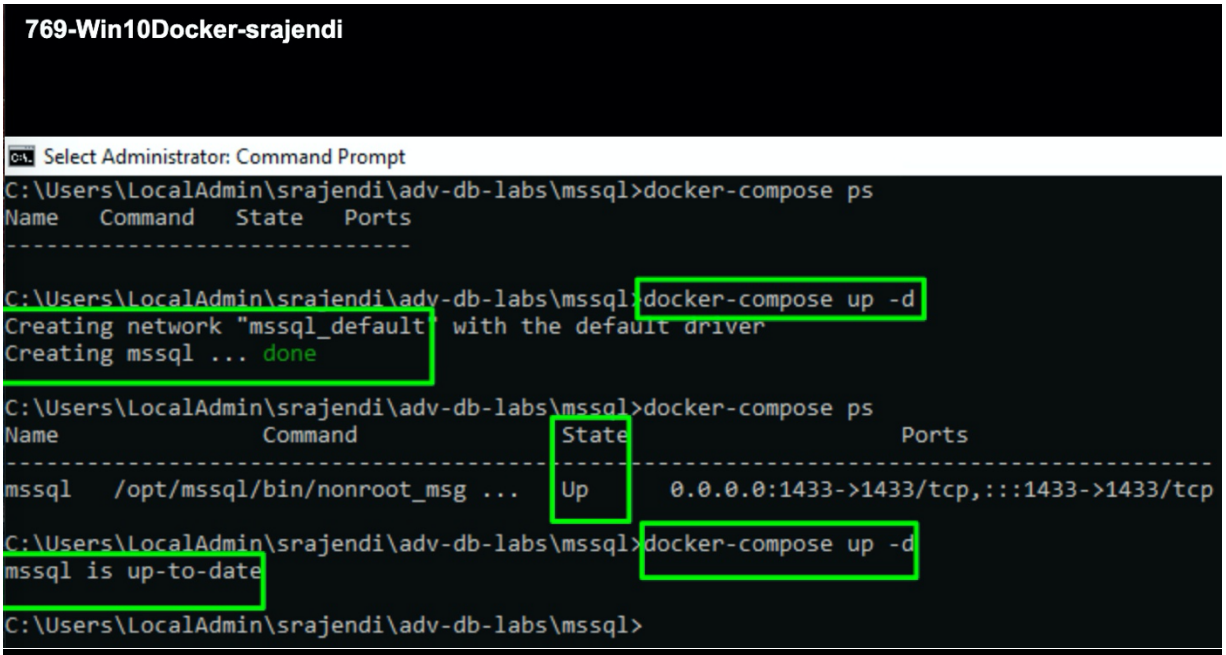
- **769-Win10Docker-srajendi**

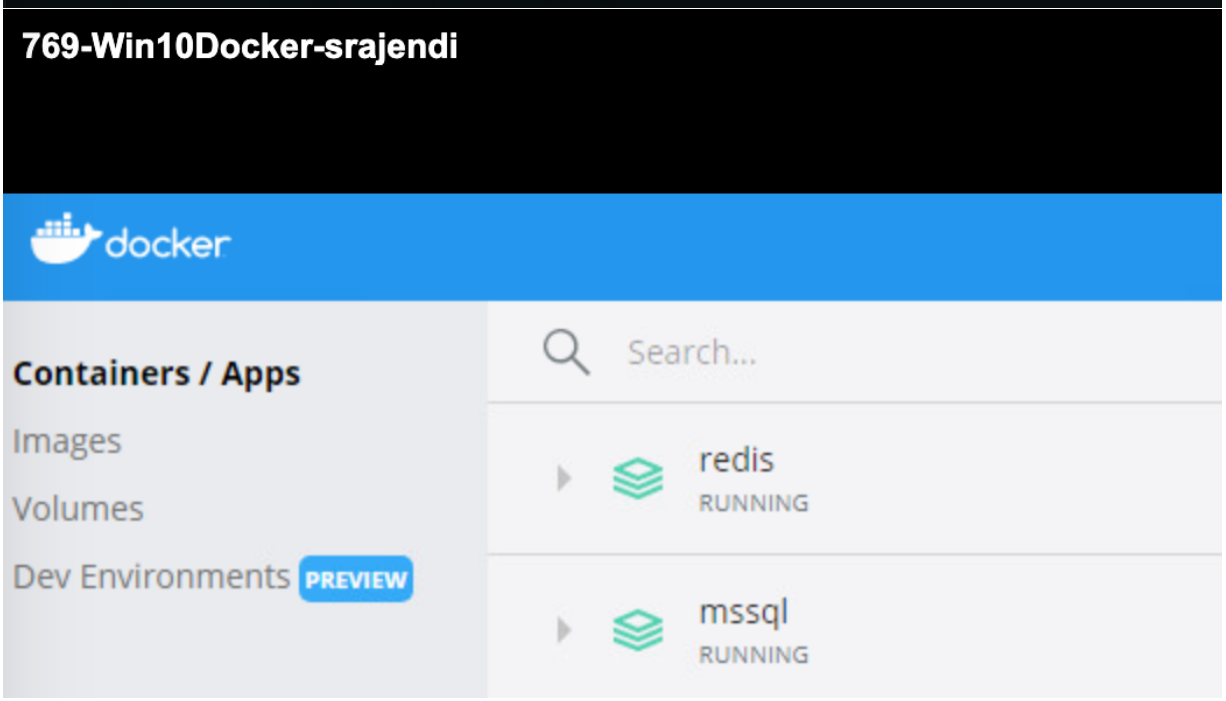


- 

4. **What happens when you bring up an environment that is already up?**

- start the service: **docker-compose up -d;** When the service is running, executing the “up -d” command would retain the same session. It doesn’t really cause any changes to the execution state. However, it may look for any upgrades available on the image between the time it was created vs current time and might update the image. I also believe there won’t be any data/network loss as all other would remain the same. In our case, it didn’t make any impact as it returned with a status as “mssql is up-to-date”

- 

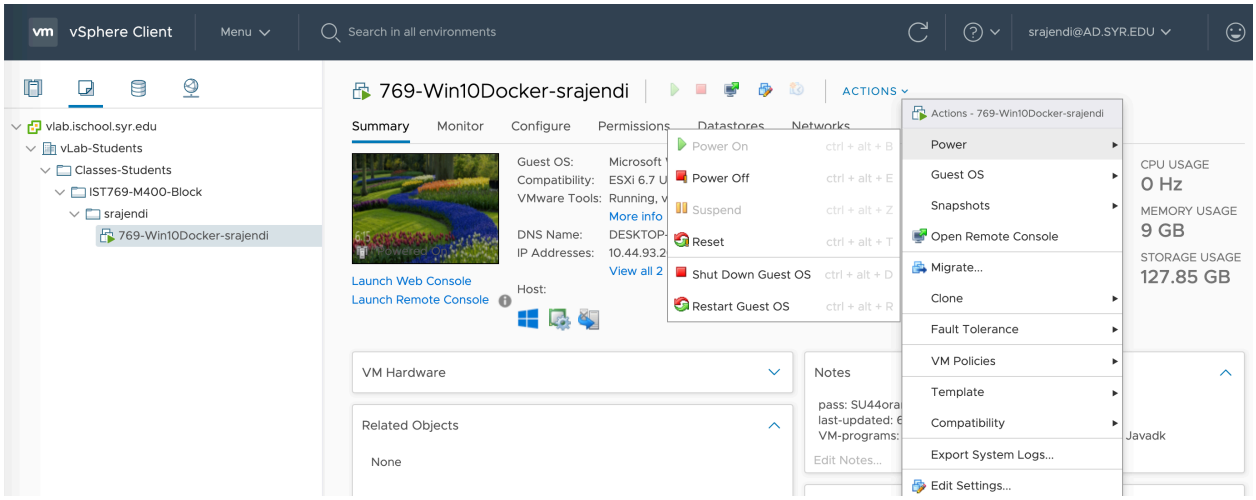
- 

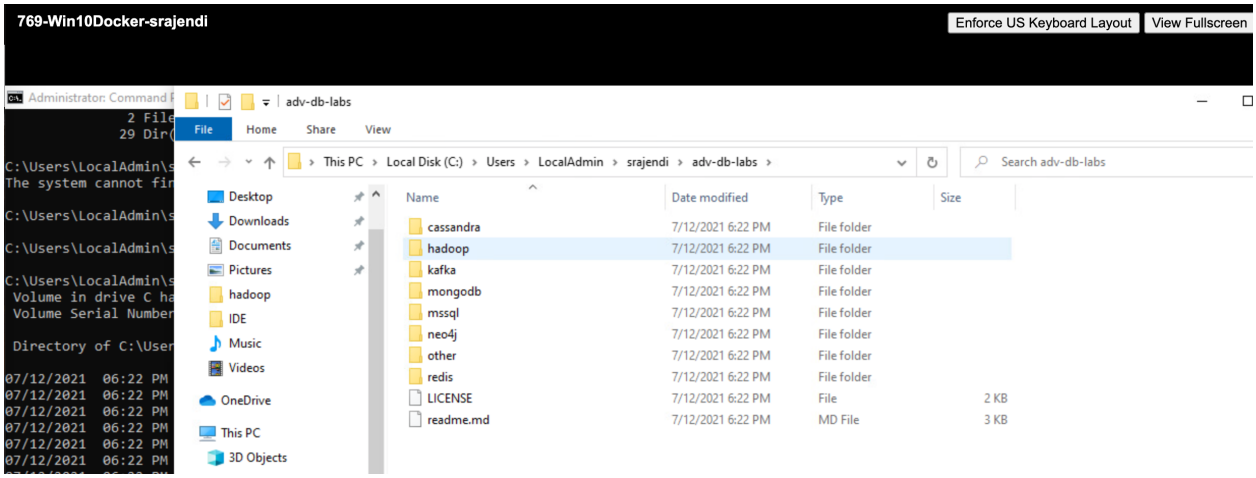
5. What was the most difficult aspect of this lab? What changes could be made to make it less difficult?

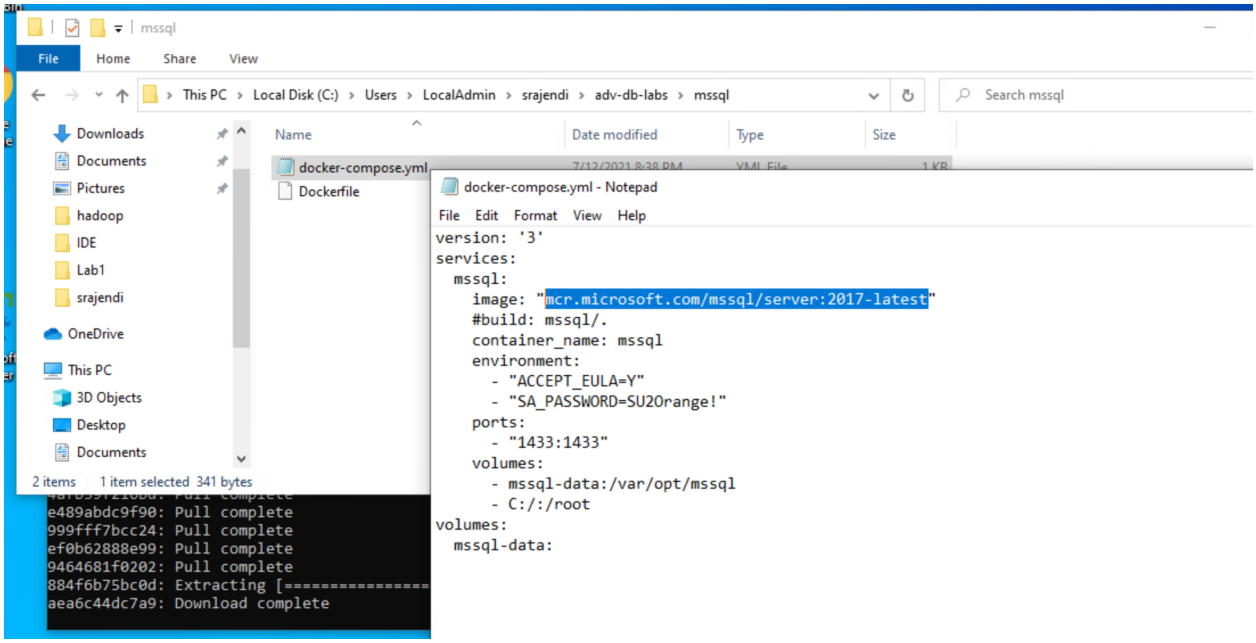
I don't think it was challenging. Maybe it would have been better if we had few hands on for Redis. I personally haven't used Redis database in the past. Only other challenge was with mssql image version. But one of the students had mentioned about the workaround to fix it. So, it was quite straightforward.

As long as vlab instance on iSchool – I am quite comfortable with this exercise. However, I would like to try to the same on my mac machine. I am not a huge fan of SQL Workbench. So, preferred MSSQL on the vlab.

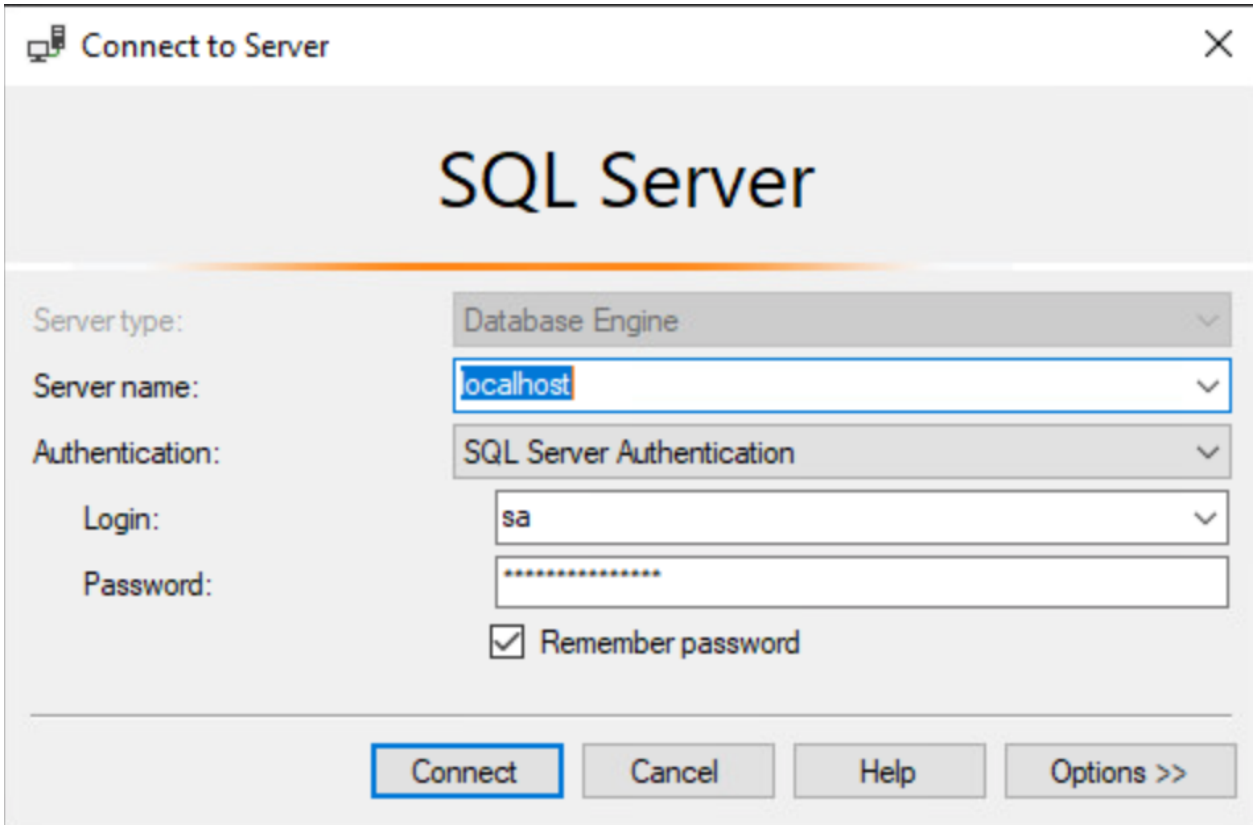
Appendix:

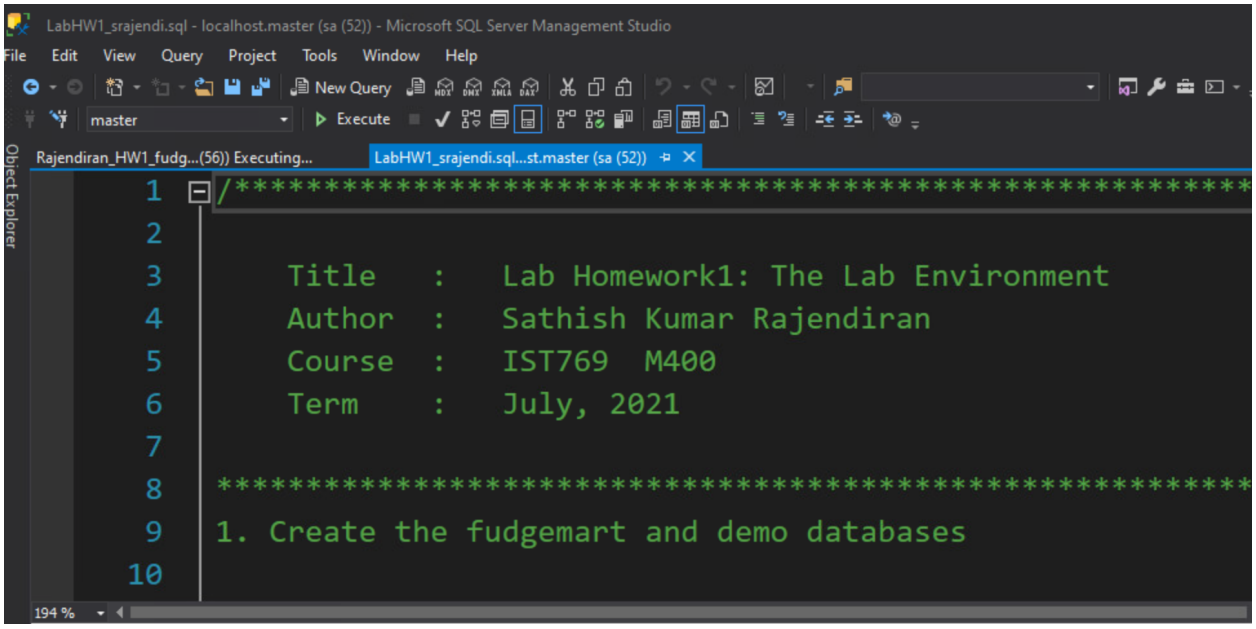







```
C:\Users\LocalAdmin\srajendi\adv-db-labs\mssql>docker-compose ps
Name   Command   State   Ports
------------------------------

C:\Users\LocalAdmin\srajendi\adv-db-labs\mssql>docker-compose up -d
Pulling mssql (mcr.microsoft.com/mssql/server:2017-latest)...
2017-latest: Pulling from mssql/server
61e03ba1d414: Pull complete
4afb39f216bd: Pull complete
e489abdc9f90: Pull complete
999fff7bcc24: Pull complete
ef0b62888e99: Pull complete
9464681f0202: Pull complete
884f6b75bc0d: Pull complete
aea6c44dc7a9: Pull complete
Digest: sha256:046fdb8988b92a9da973d2c4a237a769afee9010d79d76dd807b930a3d0e56e9
Status: Downloaded newer image for mcr.microsoft.com/mssql/server:2017-latest
Creating mssql ... done

C:\Users\LocalAdmin\srajendi\adv-db-labs\mssql>
```


Connect to Server
SQL Server
Server type:
Database Engine
Server name:
localhost
Authentication:
SQL Server Authentication
Login:
sa
Password:
Remember password
Connect
Cancel
Help
Options >>


LabHW1_srajendi.sql - localhost.master (sa (52)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
master
Execute
Rajendiran_HW1_fudg...(56)) Executing...
LabHW1_srajendi.sql...st.master (sa (52))
Object Explorer
/***
Title : Lab Homework1: The Lab Environment
Author : Sathish Kumar Rajendiran
Course : IST769 M400
Term : July, 2021
****
1. Create the fudgemart and demo databases
194 %

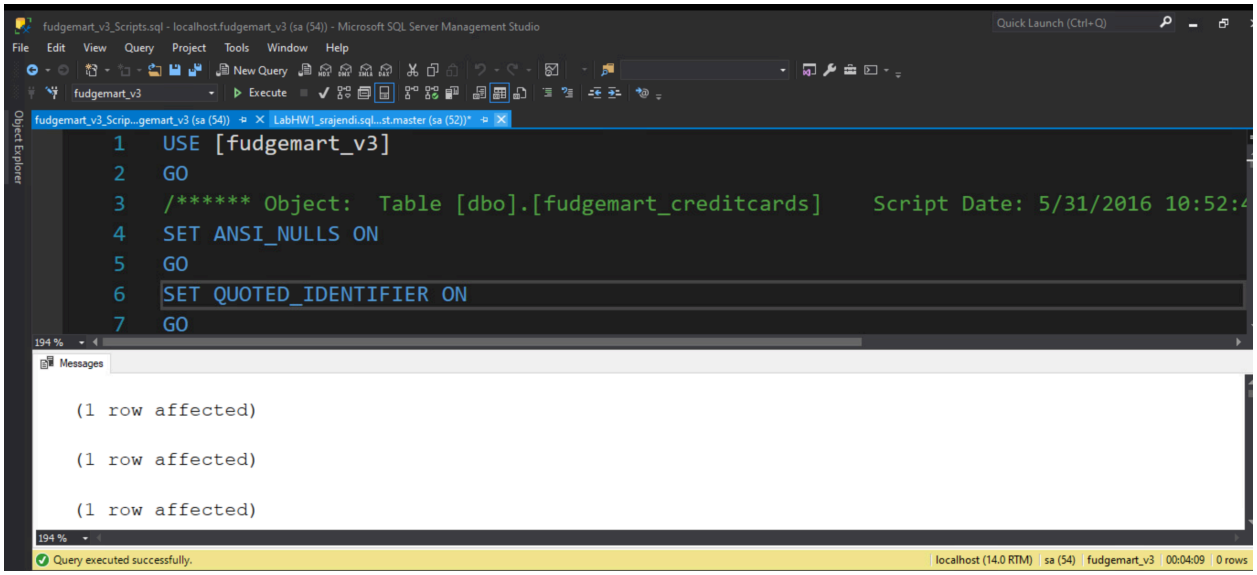
```
32  IF NOT EXISTS (SELECT * FROM sys.databases WHERE name='fudgemart_v3')
33      BEGIN
34          CREATE DATABASE fudgemart_v3
35      END
36      GO
37
38  IF NOT EXISTS (SELECT * FROM sys.databases WHERE name='demo')
39      BEGIN
40          CREATE DATABASE demo
41      END
42      GO
43  SELECT name FROM sys.databases
```

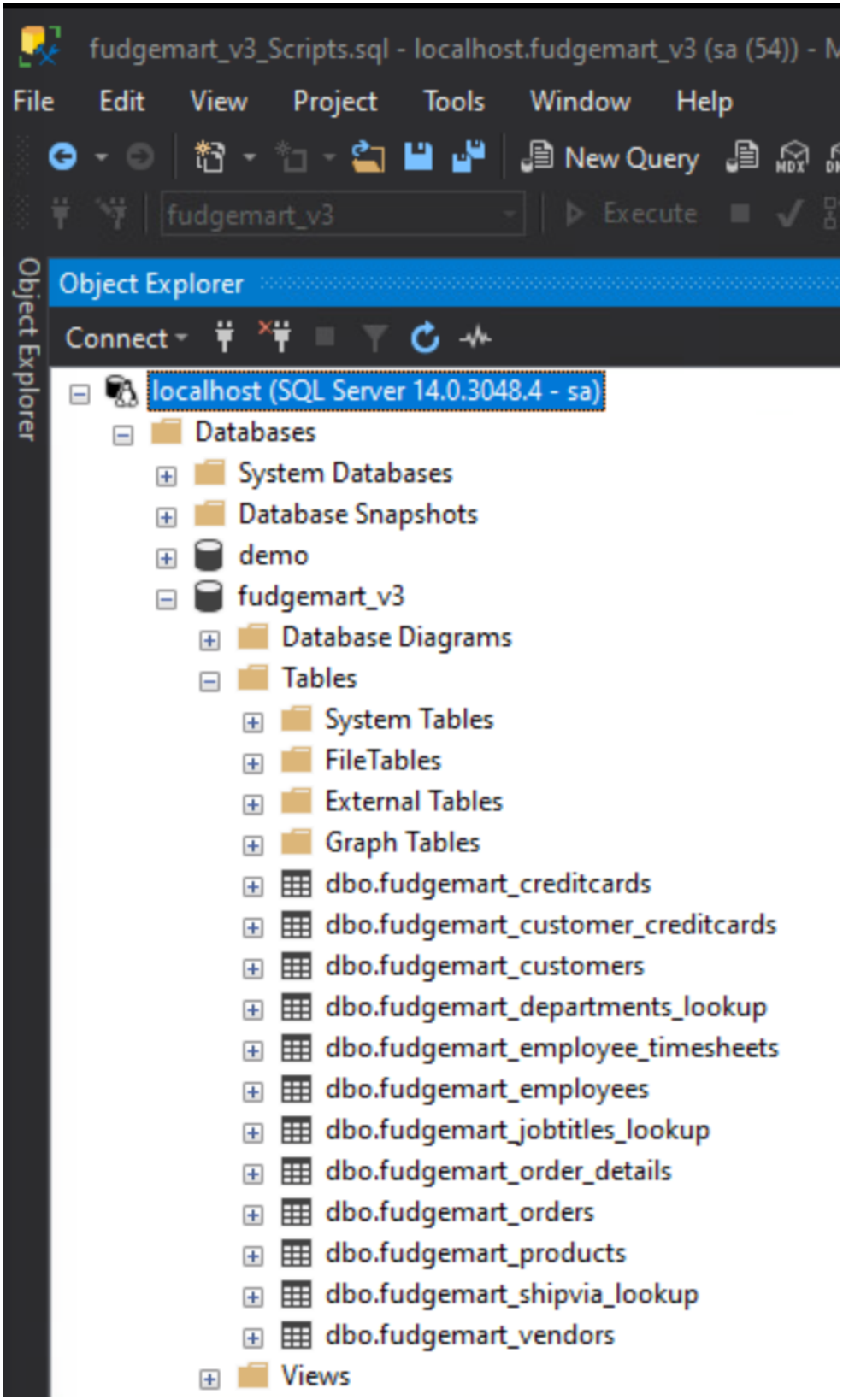
194 %

Results | Messages

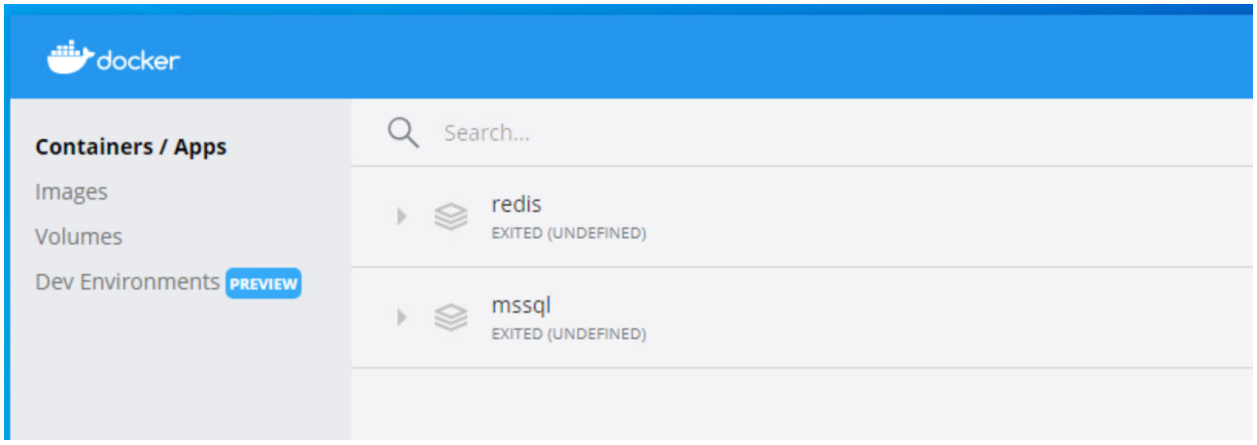
	name
1	master
2	tempdb
3	model
4	msdb
5	fudgemart_v3
6	demo

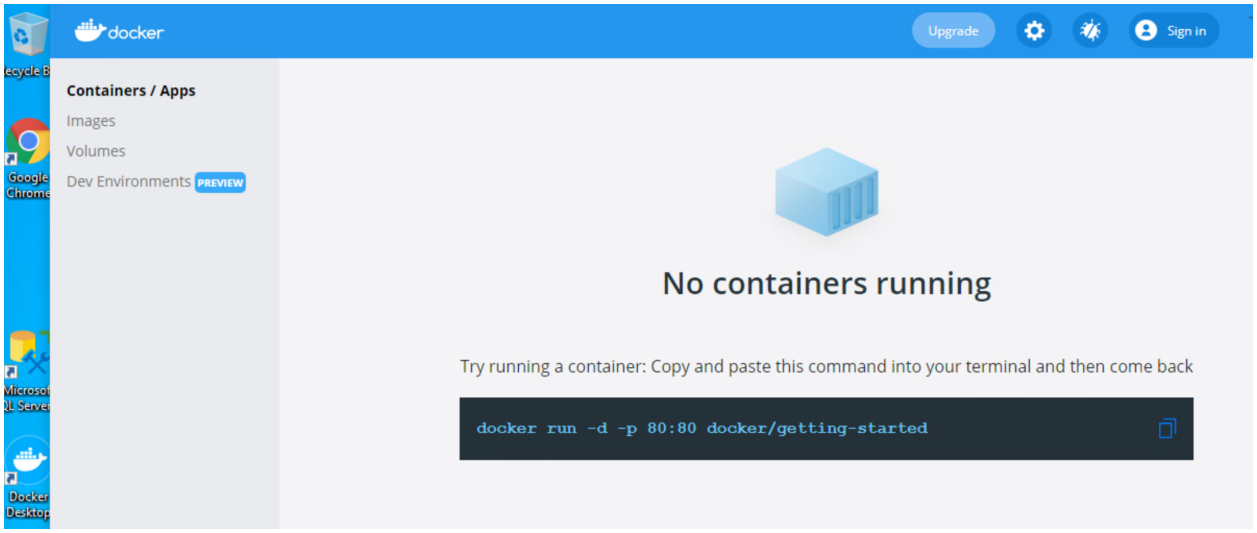
Query executed successfully. | localhost (14.0 RTM) | sa (52) | master | 00:00:02 | 6 rows



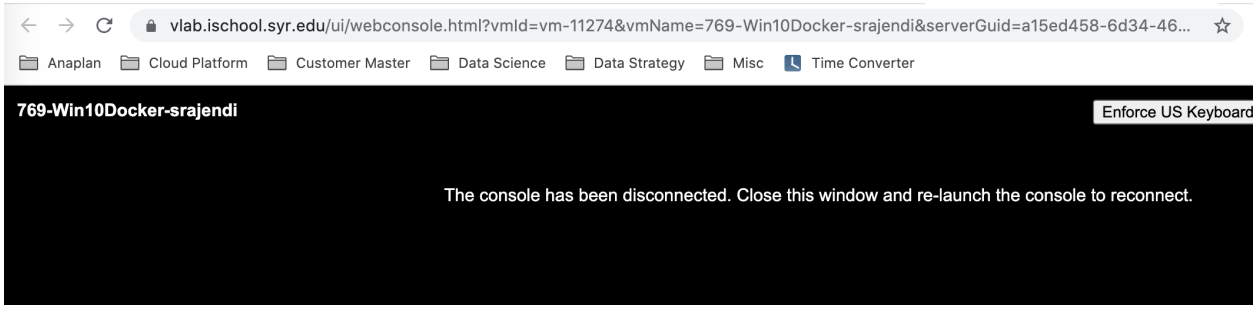

fudgemart_v3_Scripts.sql - localhost.fudgemart_v3 (sa (54)) - M
File
Edit
View
Project
Tools
Window
Help
New Query
fudgemart_v3
Execute
Object Explorer
Connect
localhost (SQL Server 14.0.3048.4 - sa)
Databases
System Databases
Database Snapshots
demo
fudgemart_v3
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.fudgemart_creditcards
dbo.fudgemart_customer_creditcards
dbo.fudgemart_customers
dbo.fudgemart_departments_lookup
dbo.fudgemart_employee_timesheets
dbo.fudgemart_employees
dbo.fudgemart_jobtitles_lookup
dbo.fudgemart_order_details
dbo.fudgemart_orders
dbo.fudgemart_products
dbo.fudgemart_shipvia_lookup
dbo.fudgemart_vendors
Views

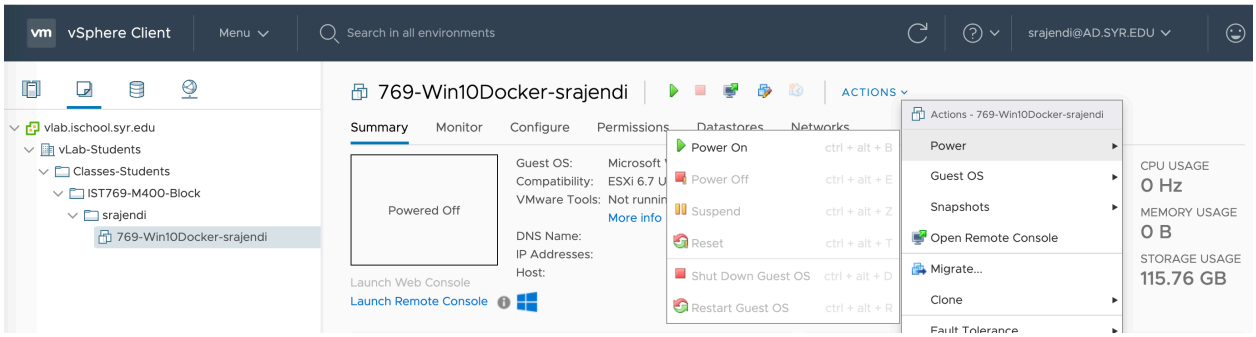
```
C:\Users\LocalAdmin\srajendi\adv-db-labs\mssql>docker-compose ps
Name            Command           State                    Ports
-------------------------------------------------------------------------------
mssql   /opt/mssql/bin/sqlservr   Up      0.0.0.0:1433->1433/tcp,:::1433->1433/tcp

C:\Users\LocalAdmin\srajendi\adv-db-labs\mssql>docker-compose down -v --rmi all
Stopping mssql ... done
Removing mssql ... done
Removing network mssql_default
Removing volume mssql_mssql-data
Removing image microsoft/mssql-server-linux:2017-latest

C:\Users\LocalAdmin\srajendi\adv-db-labs\mssql>
```






769-Win10Docker-srajendi
Shutting down


vlab.ischool.syr.edu/ui/webconsole.html?vmId=vm-11274&vmName=769-Win10Docker-srajendi&serverGuid=a15ed458-6d34-46...
Anaplan
Cloud Platform
Customer Master
Data Science
Data Strategy
Misc
Time Converter
769-Win10Docker-srajendi
Enforce US Keyboard
The console has been disconnected. Close this window and re-launch the console to reconnect.


vSphere Client
Menu
Search in all environments
srajendi@AD.SYR.EDU
vlab.ischool.syr.edu
vLab-Students
Classes-Students
IST769-M400-Block
srajendi
769-Win10Docker-srajendi
769-Win10Docker-srajendi
ACTIONS
Summary
Monitor
Configure
Permissions
Datastores
Networks
Powered Off
Guest OS:
Compatibility:
VMware Tools:
More info
DNS Name:
IP Addresses:
Host:
Launch Web Console
Launch Remote Console
Power On
Power Off
Suspend
Reset
Shut Down Guest OS
Restart Guest OS
Actions - 769-Win10Docker-srajendi
Power
Guest OS
Snapshots
Open Remote Console
Migrate...
Clone
CPU USAGE
0 Hz
MEMORY USAGE
0 B
STORAGE USAGE
115.76 GB