W10 Scenario:

Now that you have proven that the databases can run in a docker container, it is time to move them to the cloud. As you are starting to realize that moving databases is a consistent part of database administration, you should look for ways to do it more efficiently.

W10 Scenario Submission Guidelines:

Be sure you submit all elements labeled by the bolded word, **SHOW**.

1. If you have not already done so, apply your Google Cloud (GCP) coupon from your instructor according to these instructions. Please read the entire instructions carefully and do **NOT** enter a personal credit card. Your VMs should go in a new project now:
2. Create a new project in GCP and then a VM within it. Be sure to choose the options shown in this video and “deploy container” using the url here.

   Note: the term “preemptible” has been changed to “spot” and is under “Advanced>Management>Availability Policies” during VM creation. If you do not choose this, it may result in you running out of credits (if this ever happens, please see your instructor but, again, **do not** use a personal credit card).

   SHOW 1: Your new VM in the cloud with the settings matching from the video in step 2.

3. Move all of your databases to this VM. Choose **ONE** of the following methods:
   a. Follow this tutorial to create an image from your local docker container from last week, move it to your new cloud server, and create a container that **already includes** your databases from week 9. After you complete the tutorial, follow this video to create a firewall rule and prepare to make a connection. Be aware that the sa password might be the password you used from last week when you first created your original container. Refer to last week’s help document and these supplemental videos as needed. NOTE: This option would be wise and save you time if you were planning to deploy these databases to many different servers.
   b. NOTE: This step is an alternate option only available to those who struggle with **step a**. This alternate option is *more tedious and less technically valuable for your learning*. **Step a** is required for any students who used the alternate path in week 9. After this tutorial, you again backup your databases and copy them to the VM by clicking the SSH button and then the gear icon in the upper right corner and selecting ‘*upload file*.’ This is similar to what you did here in week 4, but you will use your IP address and related info for your new VM instead. If you use this method, you then have to issue copy commands (docker cp) to move the files into the running container. After which, you will connect and restore each database again. Then, you will enable the SQL Server Agent and create a job as you did in week 9. If you do **step a instead,** all of this will automatically be present in the image from week 9.

SHOW 2:

- Using the considerations in this video, connect from management studio (or Azure Data Studio) on your laptop. For the server, you should enter the external ip address of your vm then a comma and 49433. Be sure to demonstrate this connection in your video.
- The databases are now running inside your cloud docker instance as well as your local instance. These include: *Bowling, EntertainmentAgency, Recipes, SalesOrders, sample, SchoolScheduling (WideWorldImporters should be excluded*).

4. Because logins and agent jobs are on the instance level, they do not come with the database restores. Thus, you will not have any of your jobs from weeks 7 or 8 running inside of your new docker instance. Therefore, create a new backup job. **This is to be done inside the docker instance (through your management studio connection). Use the reference section below to enable the SQL Server agent in your container if it is not already running.**

   SHOW 3: Your new job inside of your docker instance and prove that all steps related to the job complete successfully.

If you did not create your container using -e 'MSSQL_AGENT_ENABLED=True', turn on the SQL Agent in your docker container later by:

- Login to your docker container as root user:

```
docker exec -it -u root <enteryourcontainername> bash
```

-Enable the agent:

```
# /opt/mssql/bin/mssql-conf set sqlagent.enabled true
```

- **Ignore the informational message and type exit to close the root session.**
- **Restart your VM**
- **After VM has started run this docker command to restart your container**
  - docker start putyourcontainernamehere