**Assignment : Create an Azure Key Vault. Create a simple Spring Boot Application that can access this Key Vault, package it in a docker container and deploy it to azure app service. After successful deployment, delete all the resources.**

1. Create an Azure Key Vault.

2. Create a Azure SQL data base with SQL authentication (refer to the recording)

3. Update the firewall rules of server so that you can connect to the database. Also modify it such that AZ resources can connect to it.

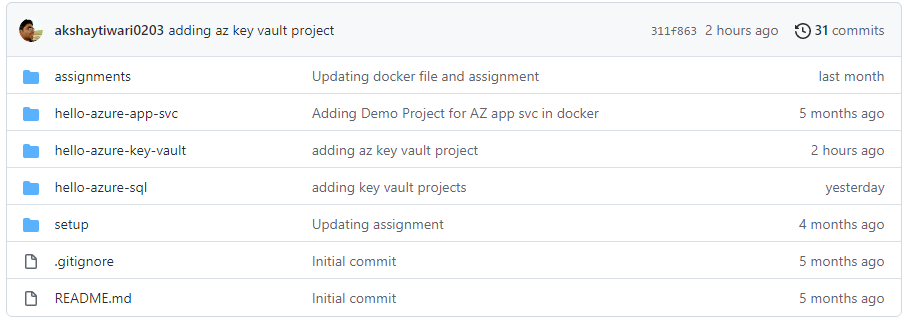
4. Create a table in newly created database by executing following query in query editor

CREATE TABLE EMPLOYEE (EMP\_NAME VARCHAR(100), EMP\_SAL INTEGER);

4. Store the password for DB into key vault as a secret.

5. Take a note of the connection String for the DB

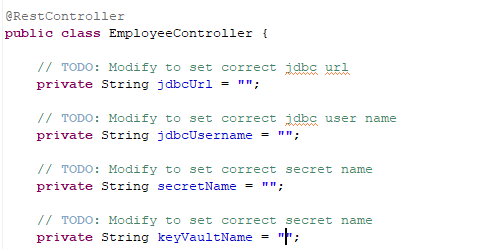
6. Download the project as a zip from <https://github.com/akshaytiwari0203/azure_learning> and extract it.



7. Copy the folder hello-azure-key-vault to a location where you would like to keep your code.

8. Import this folder as a maven project into eclipse.

9. Update the connection String in the EmployeeController. Change the connection string to have correct password. Also update the key vault name and secret name.



10. Open a command prompt and navigate inside the location where you copied hello-azure-sql

cd <base\_path>\ hello-azure-key-vault

11. Build the project using maven

mvn clean install

12. Login to dockerhub

docker login

13. Build the docker image (note there is a . in the end of command.

docker build -t <docker\_hub\_user\_name>/hello-azure-key-vault .

14. Check the image that you have created and note the imageid

docker images

15. Tag the image

docker tag <image\_id> <docker\_hub\_user\_name>/hello-azure-key-vault

16. Push the image to docker hub.

docker push <docker\_hub\_user\_name>/hello-azure-key-vault

17. Create an App service instance on Azure using this image

18. Create a system defined managed identity for app services

19. Use the managed identity created above to provide access on key vault to app service

20. Explore the service using create and get urls

21. Explore modifySecret and deleteSecret urls.

22. After verification delete the resource group