**CS 487 DATA SECURITY**

**HOP04 – Deploy Key Vault Parameters**

12/16/2020 Developed by Mary Oh

Center for Information Assurance (CIAE) @City University of Seattle (CityU)



**Before You Start**

* Version numbers may not match with the most current version at the time of writing. If given the option to choose between stable release (long-term support) or most recent, please choose the stable release rather than beta-testing version.
* This tutorial targets Windows users and MacOS users.
* There might be subtle discrepancies along the steps. Please use your best judgement while going through this cookbook style tutorial to complete each step.
* For your working directory, use your course number. This tutorial may use a different course number as an example.
* The directory path shown in screenshots may be different from yours.
* If you are not sure what to do or confused with any steps:
  + Consult the resources listed below.
  + If you cannot solve the problem after a few tries, ask a TA for help.

**Learning Outcomes**

* Create an Azure key vault and key by using ARM template
* Create an Azure key vault parameters and key

**Resources**

* Microsoft Documentation - <https://docs.microsoft.com>

**Note**

This Hands-on-Practice is almost similar to the HOP03. However, in this HOP, we will use a parameters file so we do not need to enter the parameters during the deployment process.

**What is Azure Key Vault?**

Azure Key Vault is a cloud service that provides a secure storage for secrets such as keys, passwords, certificates, and other secrets.

**Using ARM template and parameters file**

1. Download the keyvault.parameters.json file.
2. Copy the json files created from HOP03 to HOP04 folder.
3. Open the json files with Visual Studio Code.
4. Open the integrated Visual Studio Code terminal using ctrl + ` key.
5. Sign in using your Azure account information. This will open up a new window to sign in.

Graphical user interface, text, application, chat or text message

Description automatically generated

1. Create the resource group.

Text

Description automatically generated

1. Open your browser and log in to your Azure account. Go to the arm-vscode resource group and modify the access control. Add yourself as the owner and save.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

1. Back in the VSCode integrated terminal, enter the following command to obtain your object ID. Replace the email with your Azure email.



1. On the keyvault.parameters.json file, modify the “objectId” with the ID returned, and modify the “keyVaultName” to your name.

Text

Description automatically generated

1. Deploy the template. Type the following command. Ensure you are in the correct directory where the file is saved.

Text

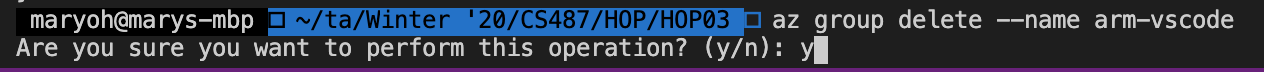
Description automatically generated

1. Verify deployment and creation of the key vault was successful.

Graphical user interface, text, application, email

Description automatically generated

1. To avoid incurring any unnecessary fees, clean up resources. Type the following command in your terminal and enter y.



1. Verify clean-up was successful. Head to your Azure in your web browser and refresh. The arm-vscode resource group should not be showing.
2. Purge the key-vault created. Select the key vault and click purge.

Graphical user interface, text, application, email, website

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**Push your work to GitHub**

1. Open the integrated Visual Studio Code terminal using ctrl + ` key. Make sure you are in the right path.
2. Type the following command:

git add . (to copy all changes you have made)

git commit -m “Submission for Module 3 - YourGitHubUsername” (To add a message to your submission)

git push origin master (to upload your work to Github)