##### Project2 #####

Digiterati team provided front end code (Angular), backend code (Python), and DB backup file (Need to restore); the ask is to  
1) Create terraform script to provision the resources.  
2) Deploy images to Azure storage account and link images to Angular frontend.  
3) Deploy Front end angular application to Azure App service using pipelines  
4) Created Azure Database and restore the database and create a new database  
5) Link DB to Backend Python application  
6) Create docker file for python application to Build Docker image  
7) Create a Yaml file for to deploy that image to ACK.  
8) Automate the whole process using Azure pipelines

**Below are the detailed task details**

**1)Create a terraform script to provision resources and push below resources terraform code to Azure repos**.

* 1)Azure ACR
* 2) Azure Kubernetes cluster
* 3) Azure MYSQL DB
* 4) Azure Storage account
* 5)Azure app service

**2) Push the given frond-end code to Azure Repos and create pipeline and deploy frond-end code to the app service which is created by terraform and test the front-end Azure app service URL**

**3) Upload the given images to Azure blob storage and update that blob storage URL in angular frontend end application and test it.**

**4) login into the Azure database and restore the database from the given file and test it. also create new database fitness.**

**5) Once DB restore is complete need to take that DB URL and update that URL in the Python backend application.**

**6) Create a docker file to build a docker image for the backend python application & create a Yaml file to deploy that docker image to the AKC**.

**And Push Backend code to Azure repos**

**7)** **Using pipelines need to build a docker image and push that image to ACR and using that Yaml file need to pull the image from ACR and deploy it to AKC**

**8) Once Kubernetes deployment is successful, take the IP of the load balancer for the python backend application and update that IP in the angular frontend.**

**THE LAST STEP IS CI/CD OF THE APPLICATION**

**First step**

**1)Create a terraform script to provision resources and push below resources terraform code to Azure repos**.

* 1)Azure ACR
* 2) Azure Kubernetes cluster
* 3) Azure MYSQL DB
* 4) Azure Storage account
* 5)Azure app service

**Terraform code for AppService**

Text

Description automatically generated

**Terraform code for Storage account**

Text

Description automatically generated

**Terraform code for ACR**

Text

Description automatically generated

**Terraform code for Azure Database MYSQL**

Text

Description automatically generated

**Terraform code for Azure Database MYSQL**

Text

Description automatically generated

**Finally pushed Terraform code to Azure repo’s**

Graphical user interface

Description automatically generated

**Second step**

**------------------------------------------------------------------------------------**

**Push the given frond-end code to Azure Repos and create pipeline and deploy frond-end code to the app service which is created by terraform and test the front-end Azure app service URL**

**Pushed Front end code to Azure Repo’s**

Graphical user interface, application

Description automatically generated

**Created Pipeline for Angular Frontend application**

A screenshot of a computer

Description automatically generated

**The first step in the pipeline is to install NPM.**

A screenshot of a computer

Description automatically generated with medium confidence

**The Second step in the pipeline is to Build Application.**

A screenshot of a computer

Description automatically generated

**The Third step in the pipeline is to Deploy application to Azure App Service**

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

**The pipeline was executed successfully**.Graphical user interface, text

Description automatically generated

**Here is App service configuration in Azure**

Graphical user interface, text, application, email

Description automatically generated

**Pipeline deployed Angular frontend application to Azure app service**.

A picture containing text, different, screenshot

Description automatically generated

A picture containing graphical user interface

Description automatically generated

**Third step**

**3) Upload the given images to Azure blob storage and update that blob storage URL in angular frontend end application and test it.**

**Uploaded images to the storage container**.

Graphical user interface, application

Description automatically generated

**Updated storage URL in Angular frontend application**.

Text

Description automatically generated

Text

Description automatically generated

**Able to see the image when trying to hit the Azure app service URL.**A picture containing text, different, screenshot, several

Description automatically generated

**4th step**

**4) login into the Azure database and restore the database from the given file and test it. also create new database fitness.**

**Added Azure DB connection details in MySQL workbench and trying to connect to DB.**

Graphical user interface, application, Word

Description automatically generated

**Able to connect to the** **database successfully.**

Graphical user interface, text, application, email

Description automatically generated

**Created and switched to database successfully.**

Text

Description automatically generated

**Able to see the Tables in database.**

Graphical user interface, text

Description automatically generated

**Able to see the Users in database.**

Graphical user interface, text

Description automatically generated

**5th step**

**5) once DB restore is complete need to take that DB URL and update that URL in the Python backend application.**

**Updated the database URL in Backend Python Application**

Text

Description automatically generated

**6th step**

**6) Create a docker file to build a docker image for the backend python application & create a Yaml file to deploy that docker image to the AKC**.

**And Push Backend code to Azure repos.**

**Created Python docker file Backend python application for to build the docker image**

Text

Description automatically generated

**Created Yaml file deploy that docker image to the AKC**

Text

Description automatically generated

**Pushed Backend end Python code to Azure Repo’s**

Graphical user interface, text, website

Description automatically generated

**7th step**

**7)Using pipelines need to build a docker image and push that image to ACR and using that Yaml file need to pull the image from ACR and deploy it to AKC**

**Created Pipeline for Python Backend application**

A screenshot of a computer

Description automatically generated

**The first step in the pipeline is to Build an image of the python application.**

A screenshot of a computer

Description automatically generated with medium confidence

**The Second step in the pipeline is to push an image to the ACR.**

A screenshot of a computer

Description automatically generated

**The last step is to deploy image to the Kubernetes cluster using YAML File**

A screenshot of a computer

Description automatically generated with medium confidence

**Backend Pipeline is success**

Graphical user interface, text, website

Description automatically generated

**Able to see the load balancer external IP of the backend python application**

Graphical user interface, text

Description automatically generated

**Copied that Backend application load balancer external IP and updated that IP in frontend application in required places.**

Text

Description automatically generated

Text

Description automatically generated

**Pushed updated frontend code to Azure Repos and executed pipeline.**

A screenshot of a computer screen

Description automatically generated with medium confidence

**Pipeline execution is success**

Graphical user interface, text

Description automatically generated

**Pipeline execution is a success, but I am unable to login to Angular frond-end application, looks like backend python and DB not getting attached to the frontend**

A picture containing text, different, screenshot, several

Description automatically generated

**Getting invalid credentials error and I need little help here.**

Graphical user interface, application

Description automatically generatedA sign on a tree

Description automatically generated with low confidence