

CS5224 Cloud Computing Assignment 1

Name: Maddi Kamal Divya

Student ID: A0178511E

1. What is Cloud Foundry software in the context of cloud?

Ans. Cloud foundry is free, open source platform as a service Paas, developed by vmware on which developers can build, test, deploy, run and scale applications.

Cloud Foundry has defined a new type of platform that allows to write software in any language and deploy it to organization's datacenter, on - site via VMWare's vSphere or OpenStack, or off - site on top of a public cloud. So, the users are locked into a single framework or a provider. Cloud Foundry uses container- based architecture to provide own isolated view of system resources and limit the resource usage.

2. In which way does Cloud Foundry help us when developing and updating web applications using boilerplates such as Node.js Boilerplate? (1 mark)

Ans. A boilerplate is a container for an application and its associated runtime environment and predefined services So, boilerplate is usually generic and can be included with minimal changes required. Boilerplate code is efficient and is easy to use.

In cloud foundry, node.js can set up a development environment, deploy an app locally and on the cloud, and then integrate a database service to the app. All of the tools and services are ready to be used and the environment guides how to use it.

3. How does Cloud Foundry software determine dependencies for the application which is uploaded/pushed by a user via the Cloud Foundry command line interface (CLI)? (1 mark)

Ans. When deploying applications via Cloud Foundry manifest.yml and package.json sets the deployment parameters and helps to resolve the dependencies. App dependencies like databases, messaging queue are delivered as services. Services can be bind or unbind to an app.

4. What is the use of VCAP_SERVICES environment variable in IBM Bluemix applications? (1 mark)

Ans. The VCAP_SERVICES environment variable contains information that is required to interact with an IBM Cloud service instance. When a service instance is binding to an application, the fields in this environment variable are set. A running application reads these environment variables after a service is bound to extract the required name-value pairs.

5. Hands-on exercise Example 1 demonstrated a simple example of using a web application to output the result of an SQL script using Node.js. In the example, we output the whole database table into the web interface. **Modify the Node.JS** code such that the output contains **only the first three fields** of the database table, and, push the updated app to cloud. (include necessary screenshots including the web page) (1 mark)

Ans. As the number of rows is not mentioned, I extracted the first three fields from the entire table (all rows).

Congratulations. Your connection to Db2 is successful.

only the first three fields from the STATS table

Country_Code		Short_Name	Table_Name
MKD		Macedonia	Macedonia, FYR
MDG		Madagascar	Madagascar
MWI		Malawi	Malawi
MYS		Malaysia	Malaysia
MDV		Maldives	Maldives
MLI		Mali	Mali
MLT		Malta	Malta
MHL		Marshall Islands	Marshall Islands
MRT		Mauritania	Mauritania
MUS		Mauritius	Mauritius
MEX		Mexico	Mexico
FSM		Micronesia	Micronesia, Fed. Sts.
MEA		Middle East & North Africa (all income levels)	Middle East & North Africa (all income levels)
MNA		Middle East & North Africa (developing only)	Middle East & North Africa
MIC		Middle income	Middle income
MDA		Moldova	Moldova
MCO		Monaco	Monaco
MNG		Mongolia	Mongolia
MNE		Montenegro	Montenegro
MAR		Morocco	Morocco
MOZ		Mozambique	Mozambique
MMR		Myanmar	Myanmar
NAM		Namibia	Namibia
NPL		Nepal	Nepal
NLD		Netherlands	Netherlands
NCL		New Caledonia	New Caledonia

6. Hands-on exercise Example 2 demonstrated a simple example of using Bluemix Personality Insights. Modify the app.js of Example 2 such that the output is **word count for input text**. (Hint: output the “word_count” field of the json you get from Personality Insights). View this

demo (<https://personality-insights-demo.ng.bluemix.net/>) for assistant. (include necessary screenshots including the web page) (2 mark)

Ans. Below screenshots shows that Obama speech text file is uploaded, and the word count is “6535”.

The image consists of two screenshots. The top screenshot shows the IBM Cloud console for a Node.js application named 'e0267822-ex2'. The application is running, and the 'Runtime' section displays four metrics: BUILDPACK (SDK for Node.js™), INSTANCES (1, All instances are running, Health is 100%), MB MEMORY PER INSTANCE (256), and TOTAL MB ALLOCATION (256, 1.375 GB still available). The bottom screenshot shows a file upload dialog box titled 'Open' with the path 'Cloud Computing > week 3 > H01-code-data > eg2'. The file list includes 'public', '.cfignore', 'app', 'manifest', 'obama-speech', and 'package'. The 'obama-speech' file is selected, and the 'File name' field is empty. The 'File type' is set to 'All Files'.

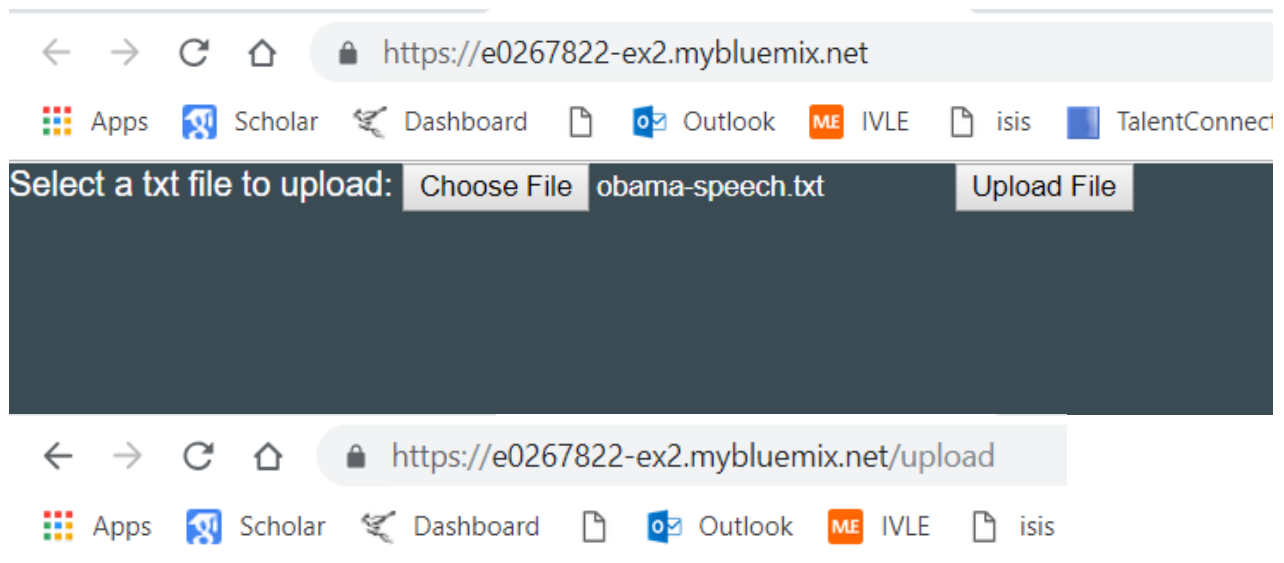
IBM Cloud console showing the runtime status of the application 'e0267822-ex2'.

Runtime Metrics:

- BUILDPACK: SDK for Node.js™
- INSTANCES: 1 (All instances are running, Health is 100%)
- MB MEMORY PER INSTANCE: 256
- TOTAL MB ALLOCATION: 256 (1.375 GB still available)

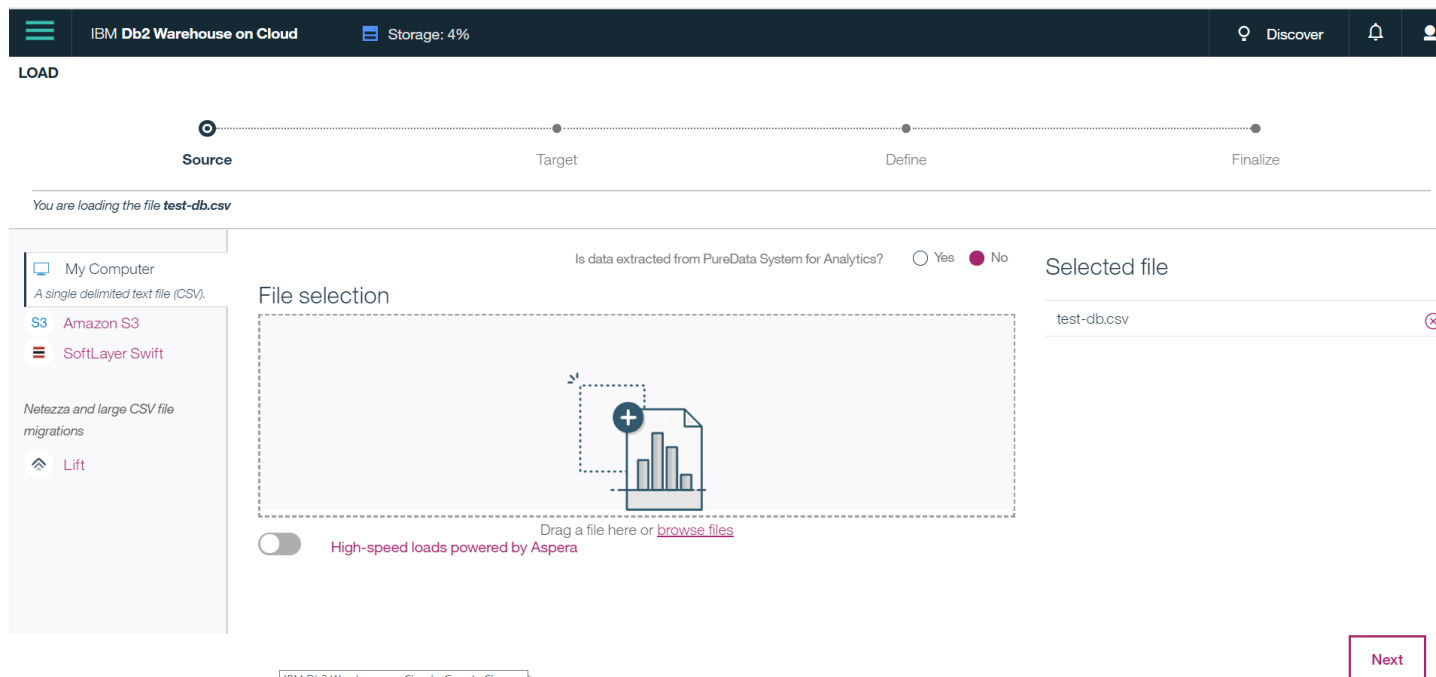
File upload dialog box showing the selection of the 'obama-speech' file.

Name	Date modified	Type	Size
public	2/11/2019 5:14 PM	File folder	
.cfignore	4/16/2016 4:04 AM	CFIGNORE File	1 KB
app	2/19/2019 10:47 PM	JavaScript File	3 KB
manifest	2/15/2019 2:46 PM	YML File	1 KB
obama-speech	4/16/2016 4:04 AM	Text Document	37 KB
package	2/11/2019 5:35 PM	JSON File	1 KB



"word_count: 6535"

7. Download the test-db.csv (contains data of students in a school) from the course web page. Upload the test-db.csv file as a new table in the same database you created in **Example 1** in the hands-on exercise. (please include screenshots) (1 marks)



LOAD

Load details

COMPLETE

My computer

test-db.csv

Target

DASH102034.TEST-DB

View Table

COMPLETE

My computer

test-db.csv

Target

DASH102034.TEST-DB

Clear All

Status

Settings

20

20

0

Rows read

Rows loaded

Rows rejected

Start time

02/17/2019 10:56:50 PM

End time

02/17/2019 10:56:52 PM

The data load job succeeded.

You can now work with your data.

Errors 0

No errors

IBM Db2 Warehouse on Cloud

Storage: 4%

Discover

Quick stats

Storage usage %

Tue, 12 Feb 2019 14:44:54 GMT

3%

Connect to IBM Db2 Warehouse on Cloud

Select a client

Select a client to connect your applications to IBM Db2 Warehouse on Cloud.

Load activity

Load Data

Refresh

	STATUS	SOURCE	FILENAME	TARGET	REQUESTED BY	ROWS LOADED	ROWS REJECTED
✓	Success	My computer	test-db.csv	DASH102034.TEST-DB	dash102034	20	0
✓	Success	My computer	stats.csv	DASH102034.EDSTATS	dash102034	241	0
✓	Success	My computer	stats.csv	DASH102034.STATS	dash102034	241	0

8. Using an R script, draw the class distribution bar chart, where x-axis denotes the class and y-axis denote the number of students. (please include screenshots) (1 marks)

Ans. There are 4 students in Class 6, 5 students in Class 7, 4 students in Class 8, 7 students in Class 9. Below bar chart depicts the same.

```
In [23]: library(ibmdbR)

# @hidden_cell
# This connection object is used to access your data and contains your credentials.
# You might want to remove those credentials before you share your notebook.

con.f2ffa5caa634cacb9419b413b35d632 <- idbConnect("DASHDB;DATABASE=BLUD8;HOSTNAME=dashdb-entry-yp-dal109-10.services.dal1.bluemix.net;PORT=50000;PROTOCOL=TCP;uid = \"dash102034\", pwd = \"OkqK_01_Artj\", conType = \"odbc\")
idbInit(con.f2ffa5caa634cacb9419b413b35d632)

data.df.1 <- as.data.frame(idb.data.frame('DASH102034.TEST'))
head(data.df.1)

# You can close the connection with the following code:
# idbClose(con.f2ffa5caa634cacb9419b413b35d632)
```

ID	FirstName	LastName	Class	Module1	Module2	Module3
1124A	Jamie	Mathews	8	History	Mathematics	Biology
3325R	Antony	Marks	9	English	Mathematics	History
3325F	Nicolas	Bourdine	6	Mathematics	English	Physics
4454Y	Sansa	Williams	9	Mathematics	Physics	Chemistry
1475Y	Samantha	Peters	7	History	Mathematics	Physics
1425F	Andy	Boucher	7	Mathematics	English	History

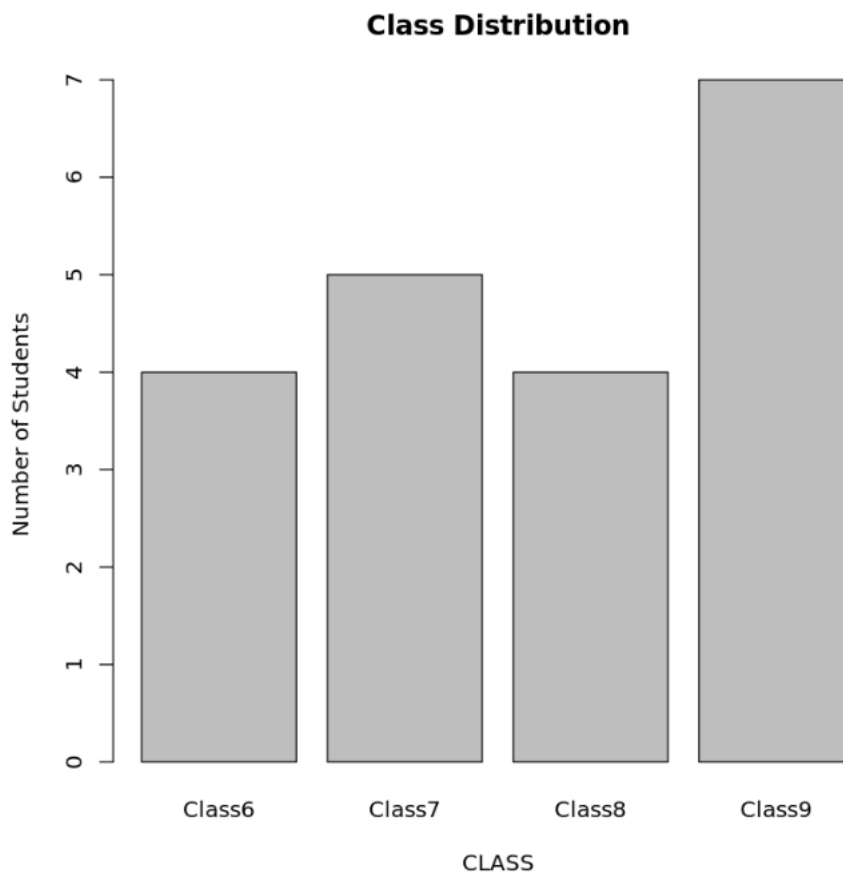
```
In [24]: library(ggplot2)

In [25]: data_frame1 <- as.data.frame(idb.data.frame('DASH102034.TEST'))[
,c('Class','FirstName')]

In [26]: data_frame1$'Class' <- factor(data_frame1$'Class',levels=c("6","7","8","9"),
labels=c("Class6","Class7","Class8","Class9"))

In [27]: counts <- table(data_frame1$'Class')

In [28]: barplot(counts, main="Class Distribution",xlab="CLASS",ylab="Number of Students")
```



9. Using an SQL script, find the number of students that are in grade 9 and studying history.
(please include screenshots) (1 marks)

Ans. There are 4 students from class 9 studying history which can be seen in result set

The screenshot displays the IBM Db2 Warehouse on Cloud interface. At the top, the header shows 'IBM Db2 Warehouse on Cloud' and 'Storage: 5%'. Below the header, the 'RUN SQL' section contains buttons for 'Run', 'Script', 'Edit', 'Favorites', and 'New tab'. The SQL editor shows a query: `1 Select Count('FirstName') from DASH102034.TEST where ("Module1" Like 'History%' or "Module2" Like 'History%' or "Module3" Like 'History%') and "Class"=9`. Below the editor, the 'Result' tab is active, showing a table with one row and one column. The row contains the value '4', indicating the count of students.

Result set
4