**CAD PHASE 3**

**LAPTOP PRICE PREDICTOR WITH IBM CLOUD**

**WATSON STUDIO**

**INTRODUCTION**

After completing our major ideas and plans on phase 2 , here we are at phase 3 , elaborating our project with detailed explanation and building , through an completely evaluated documentation of the given phase

Developing a machine learning model deployment using IBM Cloud Watson studio involves several key steps to ensure the continuity of your business operations in the event of a disaster or service interruption. Here's a step by step guide to create a robust disaster recovery plan for IBM Cloud Watson studio environment

1. Assessment and Requirements Gathering

2. Backup and Replication

3. Failover Planning

4. Network and Connectivity

5. Testing and Validation

6. Monitoring and Alerting

7. Documentation and Communication

Let's dive deeper into the steps and considerations for developing a machine learning model deployment using IBM Cloud Watson studio:

[**Introduction**](https://github.com/Briankim254/IBM-Watson-and-Sreamlit-deployment-of-mobile-price-predictor#introduction)

The laptop Price Predictor application is designed to help users estimate the price of laptop devices based on their features. The machine learning model is trained on a dataset of laptop features and corresponding prices, providing accurate predictions for a wide range of devices. By leveraging IBM Watson services and Streamlit for the frontend, the application offers a user-friendly interface and seamless deployment.

**Backup and Replication**

In this subphase , we need to configure automated, regular backups of your virtual server instances and data using IBM Cloud services like IBM Cloud Object Storage and I Implement data replication to another geographic region using IBM Cloud's Cross Region Resiliency or similar services.

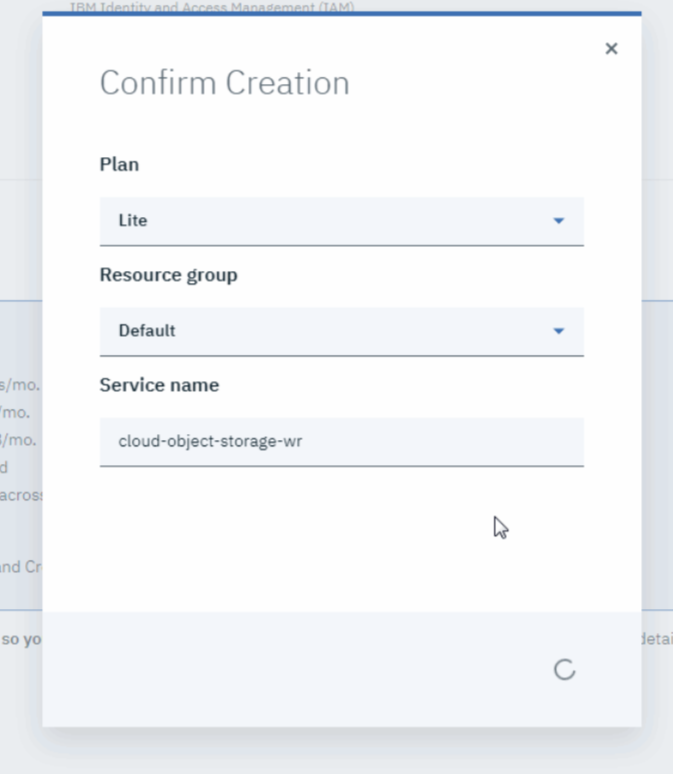
**Network and Connectivity**

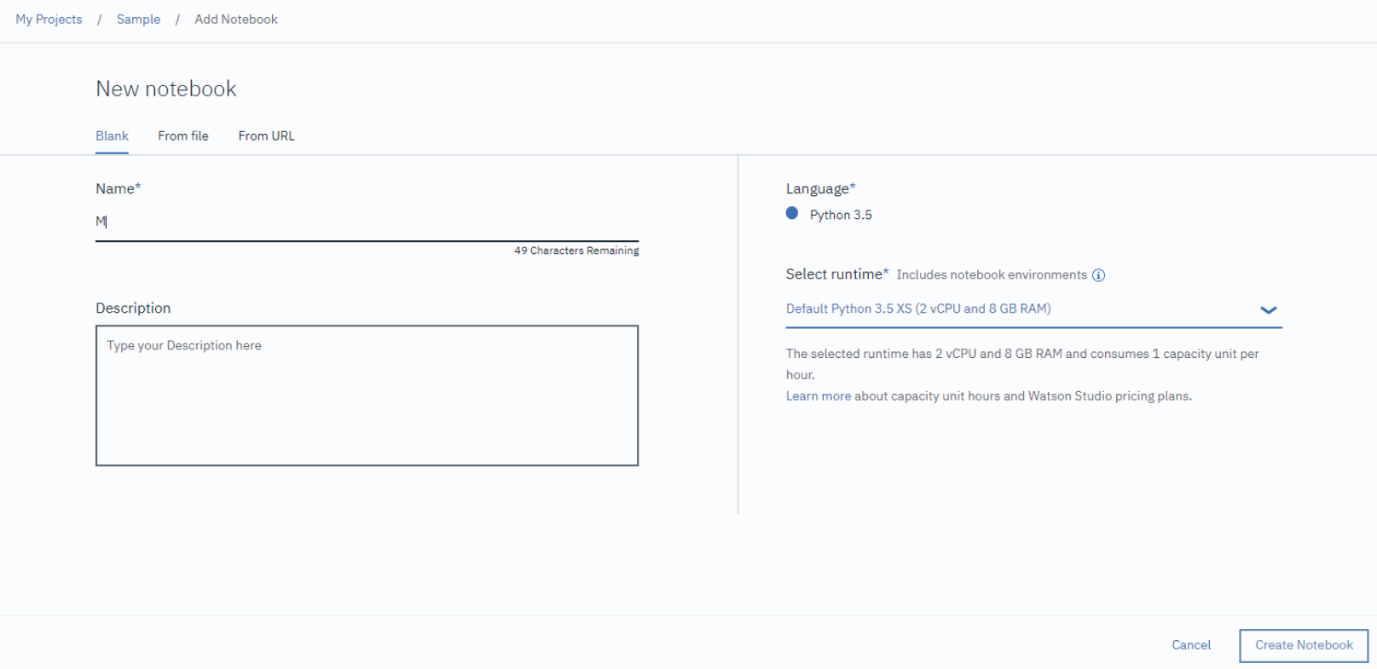
Network and connectivity is an important role in this phase. Establish a secure and reliable network connection between your primary and failover data centers using VPN, Direct Link, or similar technologies. Also ,test network configurations to ensure that applications and services will function seamlessly during a failover event.

[**Features**](https://github.com/Briankim254/IBM-Watson-and-Sreamlit-deployment-of-mobile-price-predictor#features)

* Intuitive web-based user interface powered by Streamlit
* Machine learning model deployment via IBM Watson services
* API integration for easy access to the deployed model
* **Let's break down the project into more detail starting from setting up the IBM Cloud account and provisioning Watson studio:**

**Setting Up an IBM Cloud Account :**

1. Once you are on the [account creation](https://cloud.ibm.com/registration?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMCC0100ENSkillsNetwork841-2023-01-01) page, follow the below instructions to create an IBM cloud trial account.
2. Enter your **Email** address [preferably use Gmail ID or Yahoo ID] and a strong **Password**, as per criteria and then click the **Next** button.
3. Check your email and copy and paste **Verification code**.
4. Please ensure you click on the **"Open Tool"** button
5. Open tool to get a feature code and activate the trial account .
6. Create a project in IBM Watson Studio Dashboard and assign a Cloud object Storage service to manage datasets
7. **Note: Cloud Object Storage**is a storage service in IBM Cloud. We use this service to manage our datasets for training the ML Model and store required files.
8. 
9. Add a jupyter notebook instance in your project to Develop and Deploy Machine Learning Model.
10. **Note:**You can either create a blank notebook or import from existing file or URL.



Build a Machine Learning model using jupyter notebook instance.

[**Installation & Setup**](https://github.com/Briankim254/IBM-Watson-and-Sreamlit-deployment-of-mobile-price-predictor#installation--setup)

To set up the laptop Price Predictor application on your local machine, follow these steps:

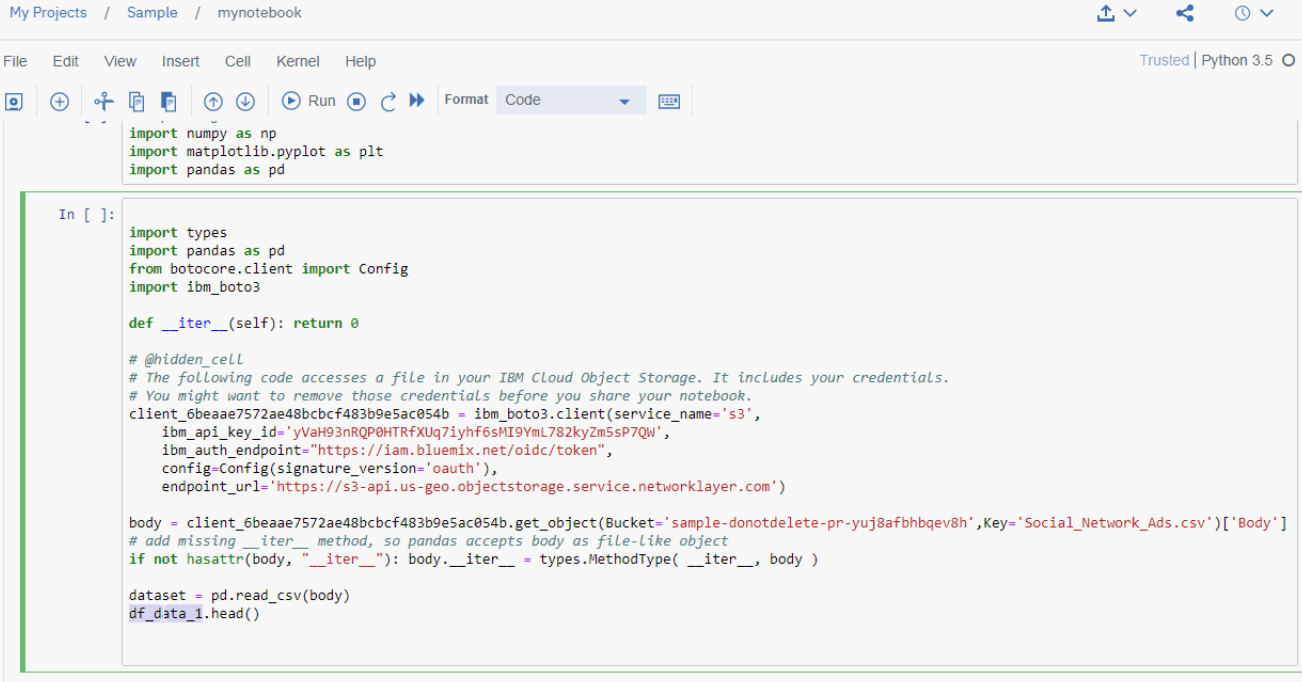
1. Clone the repository using
2. Navigate to the project directory using
3. Create a virtual environment and activate it
4. Install the required dependencies using

pip install -r requirements.txt

Let’s start Building :

**# Importing the libraries**import numpy as np  
import matplotlib.pyplot as plt  
import pandas as pd

next step we are going to upload and insert a laptop dataset for training our Machine Learning model as pandas dataframe.



Once the dataset is imported you can proceed further with pre-processing steps and building the model .

**#Check Missing Values**dataset.isnull().any()

**#Spilt Dependent and Independent Variables**

X = dataset.iloc[:, [2, 3]].values  
y = dataset.iloc[:, 4].values

**# Splitting the dataset into the Training set and Test set**

from sklearn.model\_selection import train\_test\_split  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size= 0.25, random\_state = 0)

**# Fitting Decision Tree Classification to the Training set**

from sklearn.tree import DecisionTreeClassifier  
classifier = DecisionTreeClassifier(criterion = 'entropy', random\_state = 0)  
classifier.fit(X\_train, y\_train)

**# Predicting the Test set results**

y\_pred = classifier.predict(X\_test)

**#Finding the accuracy score**

from sklearn.metrics import accuracy\_score  
print("Accuracy Score: ",accuracy\_score(y\_test,y\_pred)\*100,"%")

create Machine Learning service in AI Category.

Import **WatsonMachineLearningAPIClient**library. Watson studio uses Watson Machine Learning service credentials to access WML service.

from watson\_machine\_learning\_client import  
WatsonMachineLearningAPIClient

wml\_credentials={  
 "url": "xxxxxxxxxxxxxxxxxx",  
 "apikey": "xxxxxxxxxxxxxxxxxxxxxxx",  
 "username": "xxxxxxxxxxxxxxxxxxxx",  
 "password": "xxxxxxxxxxxxxxxxxxxxxxx",  
 "instance\_id": "xxxxxxxxxxxxxxxxxxxxxx"  
}

client = WatsonMachineLearningAPIClient(wml\_credentials)

we have to specify our machine learning model properties and store the model in WML repository.

#Specify the Properties   
model\_props = {client.repository.ModelMetaNames.AUTHOR\_NAME: "Abhi",   
client.repository.ModelMetaNames.AUTHOR\_EMAIL: "",   
client.repository.ModelMetaNames.NAME: "MyModel"}  
#Store the Machine Learning Model  
model\_artifact=client.repository.store\_model(**classifier**, meta\_props=model\_props)

Now ,we can find the list of saved models

client.repository.list()

**So , what is an application here ??**

* In the context of the project we're discussing, an "application" refers to the software or program that is hosted on your virtual server within IBM Cloud. It's the software that provides specific functionality, whether it's a website, web service, database, or any other type of program.

**Here are a few examples to illustrate what an "application" could be:**

**1. Website:** Your application could be a simple website that provides information about a topic, a blog, or an e-commerce site.

**2. Web Service:** It could be a RESTful API or a web service that provides functionality to other software applications.

**3. Database:** The application could be a database server that stores and manages data, such as customer records, product information, or financial transactions.

**4. Content Management System (CMS):** It might be a CMS like WordPress that powers a blog or content-driven website.

**5. Custom Software:** The application could be custom software that you or your team have developed for a specific purpose, such as inventory management, order processing, or customer relationship management.