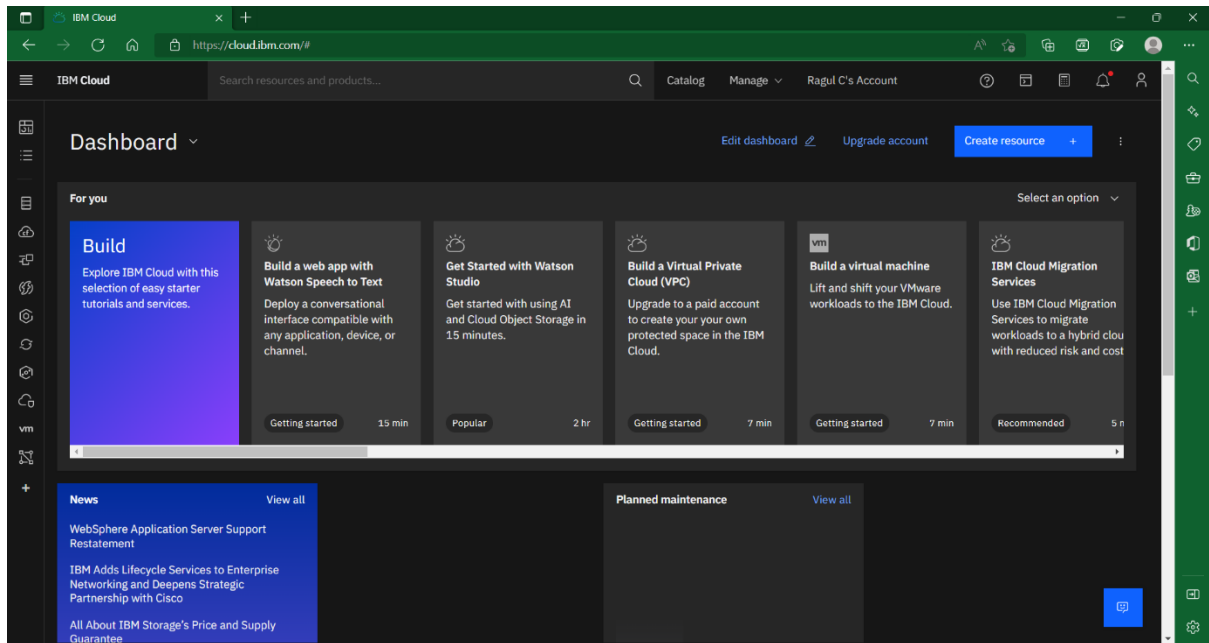


SETTING UP APPLICATION ENVIRONMENT

1.CREATE IBM CLOUD ACCOUNT



2.INSTALL IBM CLOUD CLI

```
Microsoft Windows [Version 10.0.22000.1098]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>ibmcloud

NAME:
  ibmcloud - A command line tool to interact with IBM Cloud
  Find more information at: https://ibm.biz/cli-docs

USAGE:
  [environment variables] ibmcloud [global options] command [arguments...] [command options]

VERSION:
  2.12.1+b8488a1-2022-10-31T15:08:10+00:00

COMMANDS:
  account      Manage accounts, users, orgs and spaces
  api          Set or view target API endpoint
  billing      Retrieve usage and billing information
  catalog      Manage catalog
  cf           Run Cloud Foundry CLI with IBM Cloud CLI context
  config       Write default values to the config
  cr           Manage IBM Cloud Container Registry content and configuration.
  dev          Create, develop, deploy, and monitor applications
  enterprise   Manage enterprise, account groups and accounts.
  iam          Manage identities and access to resources
  login        Log user in
  logout       Log user out
  plugin       Manage plug-ins and plug-in repositories
  regions      List all the regions
  resource     Manage resource groups and resources
  resources    List all resources
```

3.DOCKER CLI INSTALLATION

```
Command Prompt

C:\Users\HP>docker

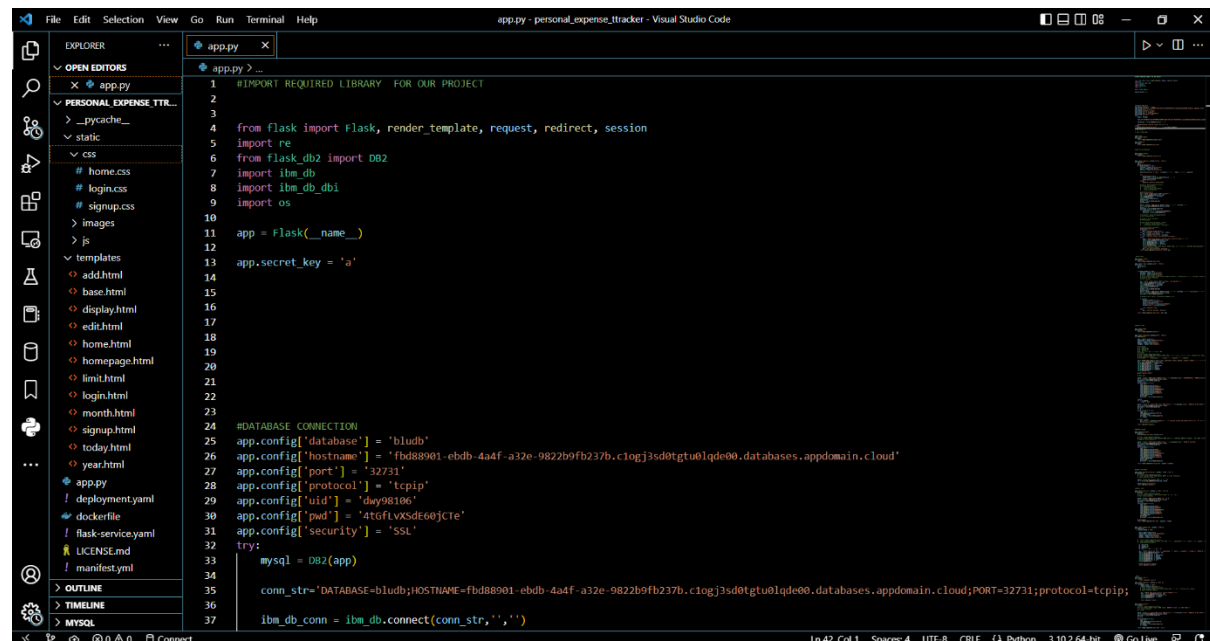
Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default
                        "C:\\Users\\HP\\.docker")
  -c, --context string  Name of the context to use to connect to the
                        daemon (overrides DOCKER_HOST env var and
                        default context set with "docker context use")
  -D, --debug           Enable debug mode
  -H, --host list       Daemon socket(s) to connect to
  -l, --log-level string Set the logging level
                        ("debug"|"info"|"warn"|"error"|"fatal")
                        (default "info")
  --tls                Use TLS; implied by --tlsverify
  --tlscacert string    Trust certs signed only by this CA (default
                        "C:\\Users\\HP\\.docker\\ca.pem")
  --tlscert string      Path to TLS certificate file (default
                        "C:\\Users\\HP\\.docker\\cert.pem")
  --tlskey string       Path to TLS key file (default
                        "C:\\Users\\HP\\.docker\\key.pem")
  --tlsverify           Use TLS and verify the remote
  -v, --version         Print version information and quit

Management Commands:
  builder      Manage builds
  buildx*      Docker Buildx (Docker Inc., v0.9.1)
```

4.CREATE FLASK PROJECT



```
app.py - personal_expense_tracker - Visual Studio Code

1  #IMPORT REQUIRED LIBRARY FOR OUR PROJECT
2
3
4  from flask import Flask, render_template, request, redirect, session
5  import re
6  from flask_db2 import DB2
7  import ibm_db
8  import ibm_db_dbi
9  import os
10
11 app = Flask(__name__)
12
13 app.secret_key = 'a'
14
15
16
17
18
19
20
21
22
23
24 #DATABASE CONNECTION
25 app.config['database'] = 'bludb'
26 app.config['hostname'] = 'fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgu1qde00.databases.appdomain.cloud'
27 app.config['port'] = '32731'
28 app.config['protocol'] = 'tcpip'
29 app.config['uid'] = 'duy98106'
30 app.config['pwd'] = '4tGfLVxsde60jCte'
31 app.config['security'] = 'SSL'
32 try:
33     mysql = DB2(app)
34
35     conn_str='DATABASE=bludb;HOSTNAME=fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgu1qde00.databases.appdomain.cloud;PORT=32731;protocol=tcpip;
36
37     ibm_db_conn = ibm_db.connect(conn_str, '', '')
```