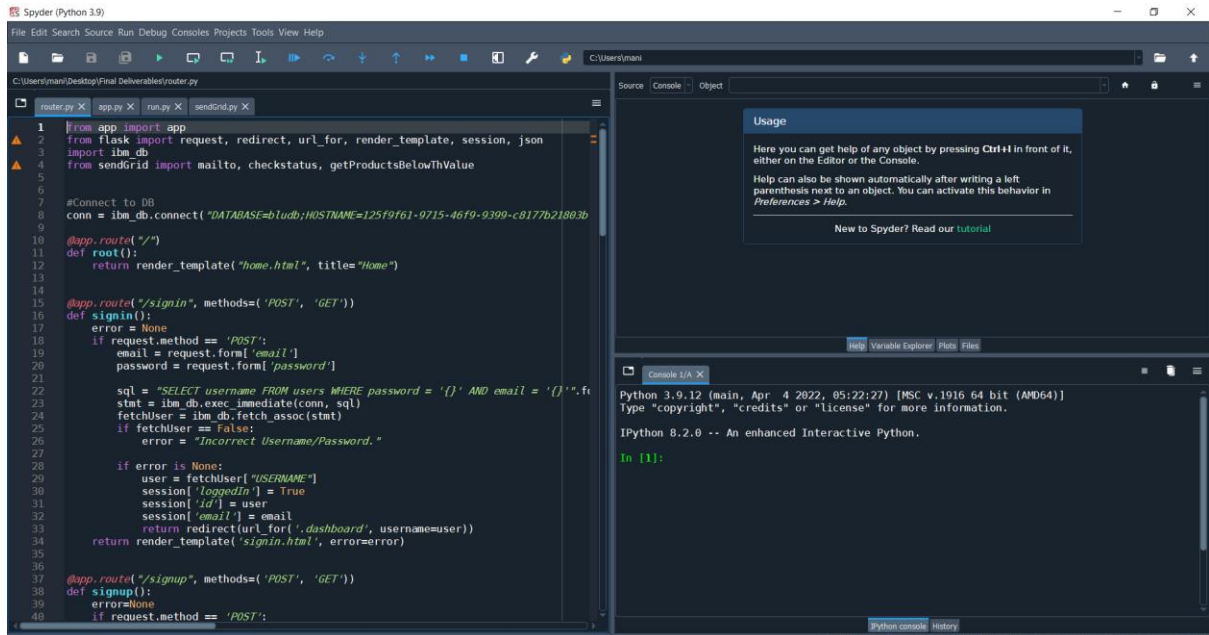


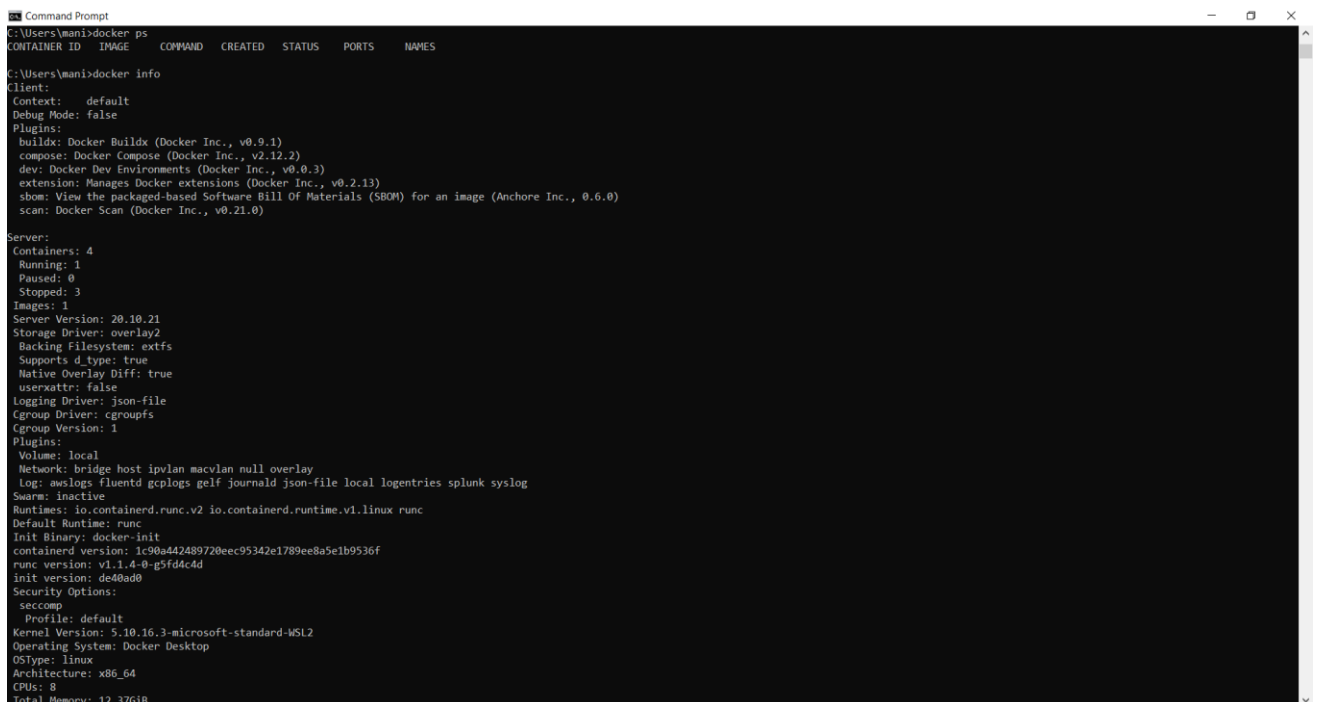
| | |
|--------------|--|
| Team ID | PNT2022TMID04548 |
| Project Name | Inventory Managment System for Retailers |

Setting up application environment

Flask Project



Docker Cli Installation



IBM cloud Cli Installation

```
Microsoft Windows [Version 10.0.19044.2193]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mani>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]

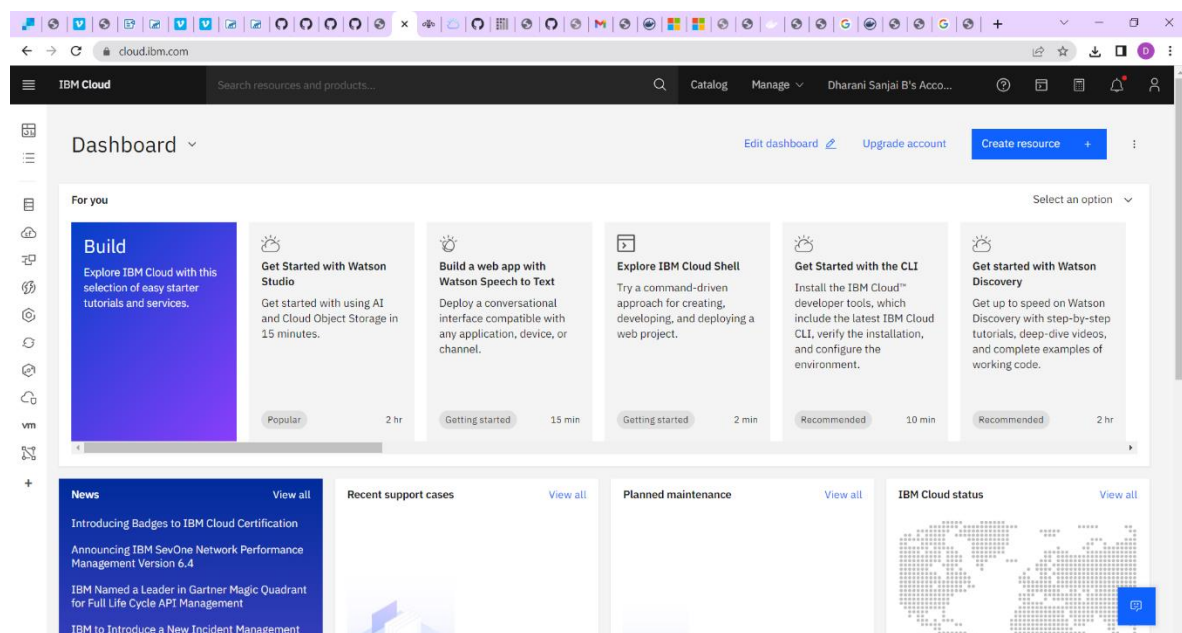
VERSION:
  2.12.1+b8488a1-2022-10-31T15:08:10+00:00

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
  sl         Manage Classic infrastructure services
  target      Set or view the targeted region, account, resource group, org or space
  update      Update CLI to the latest version
  version     Print the version
  help, h     Show help

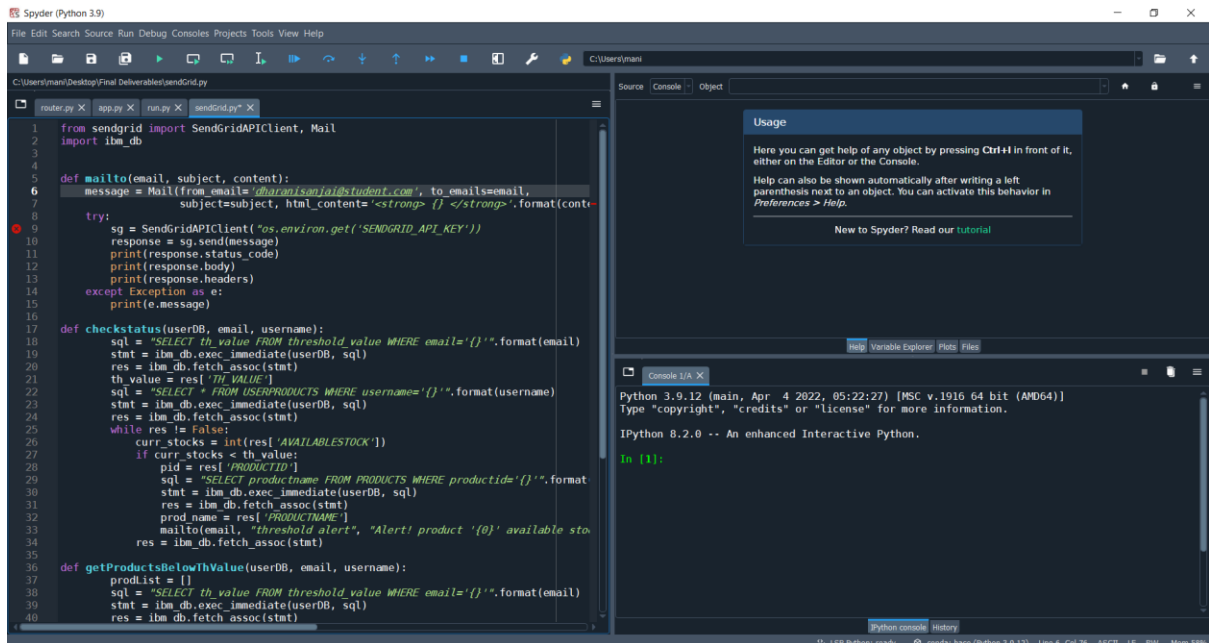
Enter 'ibmcloud help [command]' for more information about a command.

ENVIRONMENT VARIABLES:
  IBM_CLOUD_COLOR=false          Do not colorize output
  IBM_CLOUD_VERSION_CHECK=false  Do not check latest version for update
  IBM_CLOUD_HTTP_TIMEOUT=5       A time limit for HTTP requests
  IBM_CLOUD_API_KEY=api_key_value API Key used for login
  IBM_CLOUD_CR_VPC_URL=ur1_value The custom server URL to use when obtaining an instance identity token and IAM token as a VPC VSI compute resource. This value will replace the default server endpoint of the VPC VSI Instance Identity token service.
  IBM_CLOUD_CR_TOKEN=cr_token_value Compute resource token used for login. Can either be a token string or a path to a @file.
  IBM_CLOUD_CR_PROFILE=profile_value The name, ID, or CRN of the linked trusted IAM profile to be used when obtaining the IAM access token. If authenticating as a VPC VSI compute resource, only specifying a trusted profile CRN or ID is supported.
  IBM_CLOUD_TRACE=true          Print API request diagnostics to stdout
```

Create ibm cloud account :



Sendgrid



The image shows the Spyder Python IDE interface. The main editor window displays a Python script with the following code:

```
1 from sendgrid import SendGridAPIClient, Mail
2 import ibm_db
3
4
5 def mailto(email, subject, content):
6     message = Mail(from_email='dharanishan@student.com', to_emails=email,
7                   subject=subject, html_content='<strong> {} </strong>'.format(cont=
8
9     try:
10         sg = SendGridAPIClient(os.environ.get('SENDGRID_API_KEY'))
11         response = sg.send(message)
12         print(response.status_code)
13         print(response.body)
14         print(response.headers)
15     except Exception as e:
16         print(e.message)
17
18 def checkstatus(userDB, email, username):
19     sql = "SELECT th.value FROM threshold_value WHERE email='{}'".format(email)
20     stmt = ibm_db.exec_immediate(userDB, sql)
21     res = ibm_db.fetch_assoc(stmt)
22     th_value = res['TH_VALUE']
23     sql = "SELECT * FROM USERPRODUCTS WHERE username='{}'".format(username)
24     stmt = ibm_db.exec_immediate(userDB, sql)
25     res = ibm_db.fetch_assoc(stmt)
26     while res != False:
27         curr_stocks = int(res['AVAILABLESTOCK'])
28         if curr_stocks < th_value:
29             pid = res['PRODUCTID']
30             sql = "SELECT productname FROM PRODUCTS WHERE productid='{}'".format
31             stmt = ibm_db.exec_immediate(userDB, sql)
32             res = ibm_db.fetch_assoc(stmt)
33             prod_name = res['PRODUCTNAME']
34             mailto(email, "threshold alert", "Alert! product '{}' available sto
35             res = ibm_db.fetch_assoc(stmt)
36
37 def getProductsBelowThValue(userDB, email, username):
38     productList = []
39     sql = "SELECT th.value FROM threshold_value WHERE email='{}'".format(email)
40     stmt = ibm_db.exec_immediate(userDB, sql)
41     res = ibm_db.fetch_assoc(stmt)
```

The right-hand pane shows the 'Usage' section for the 'mailto' function, providing help text and a link to the tutorial. Below this, the 'Console' pane displays the output of the script, showing the Python version (3.9.12) and the IPython version (8.2.0).