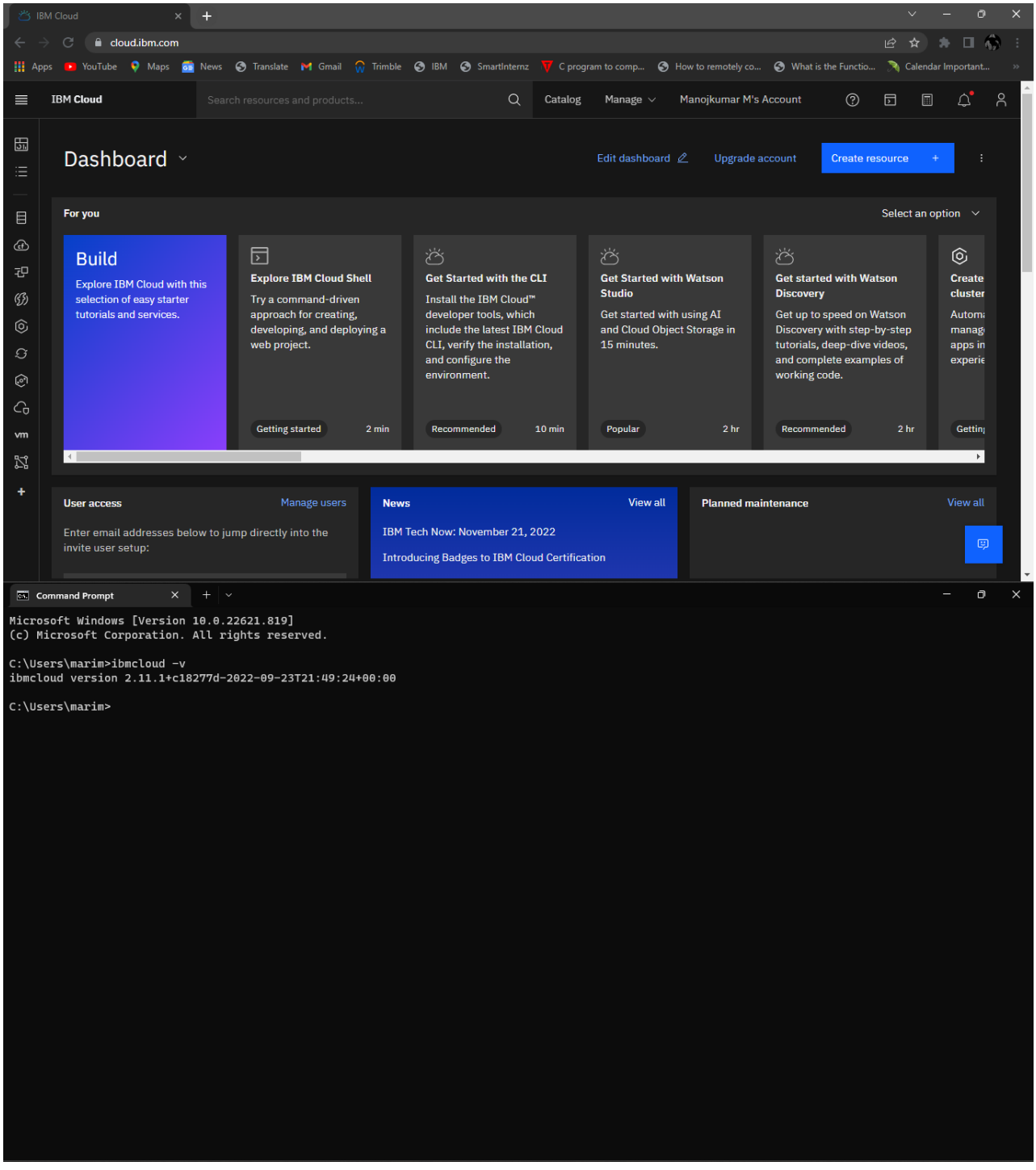
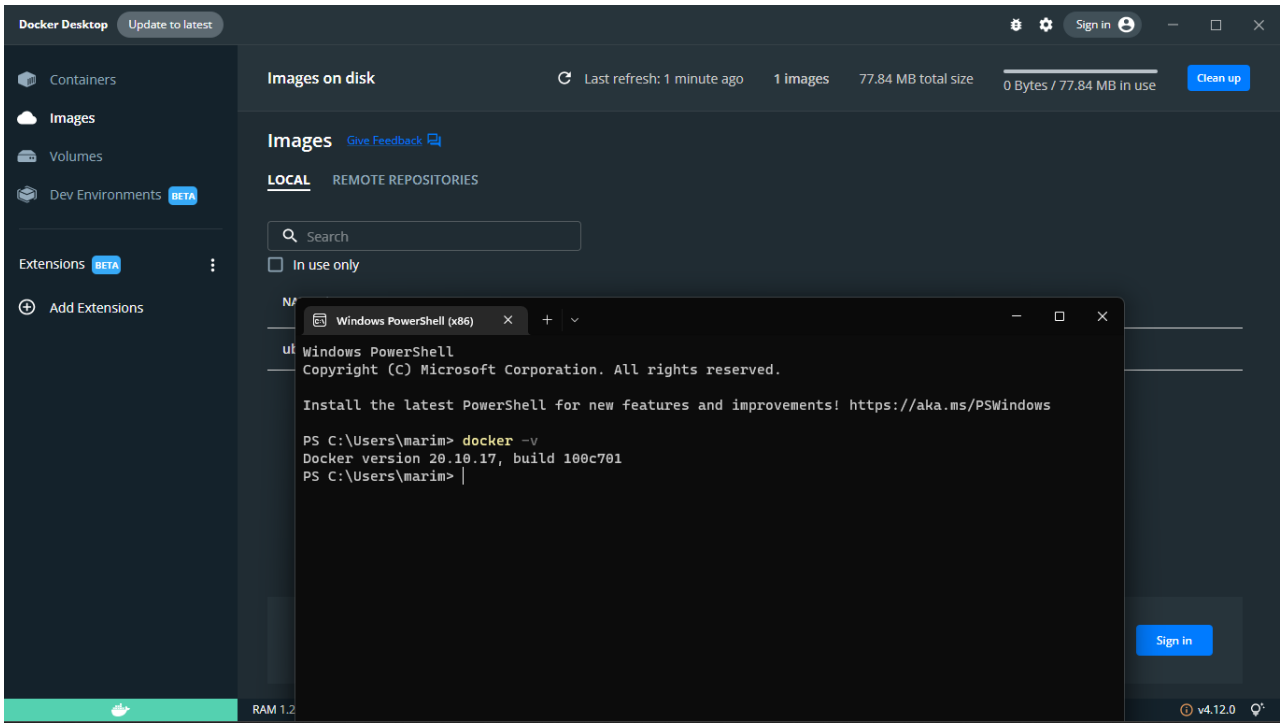
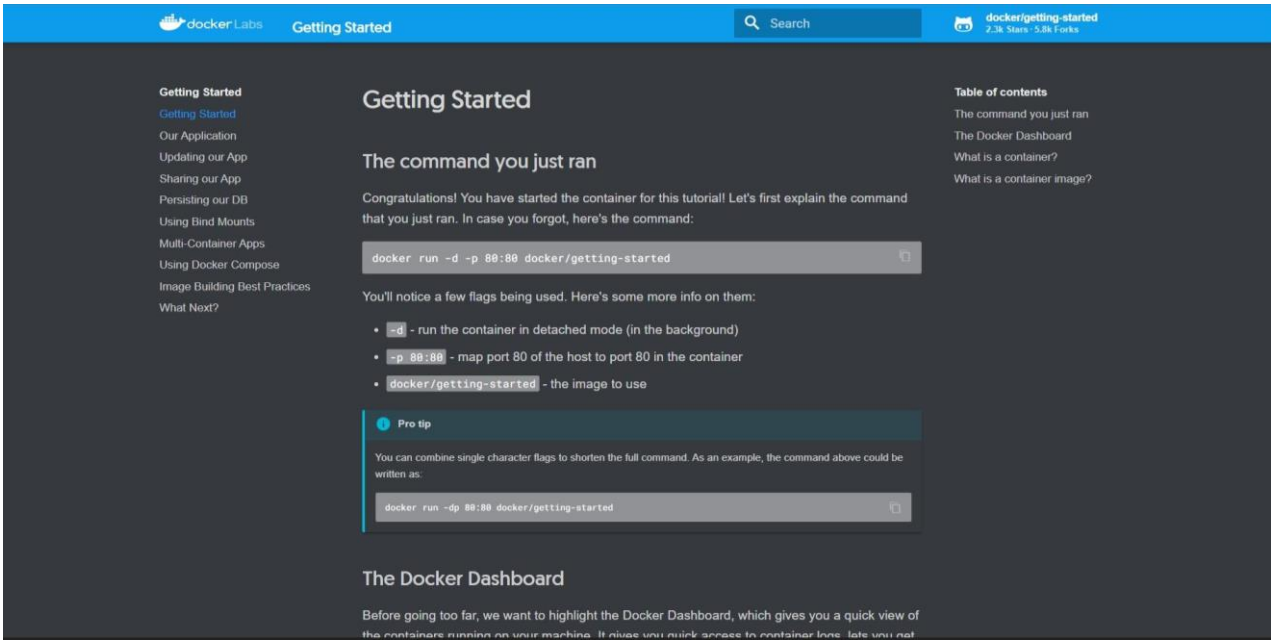
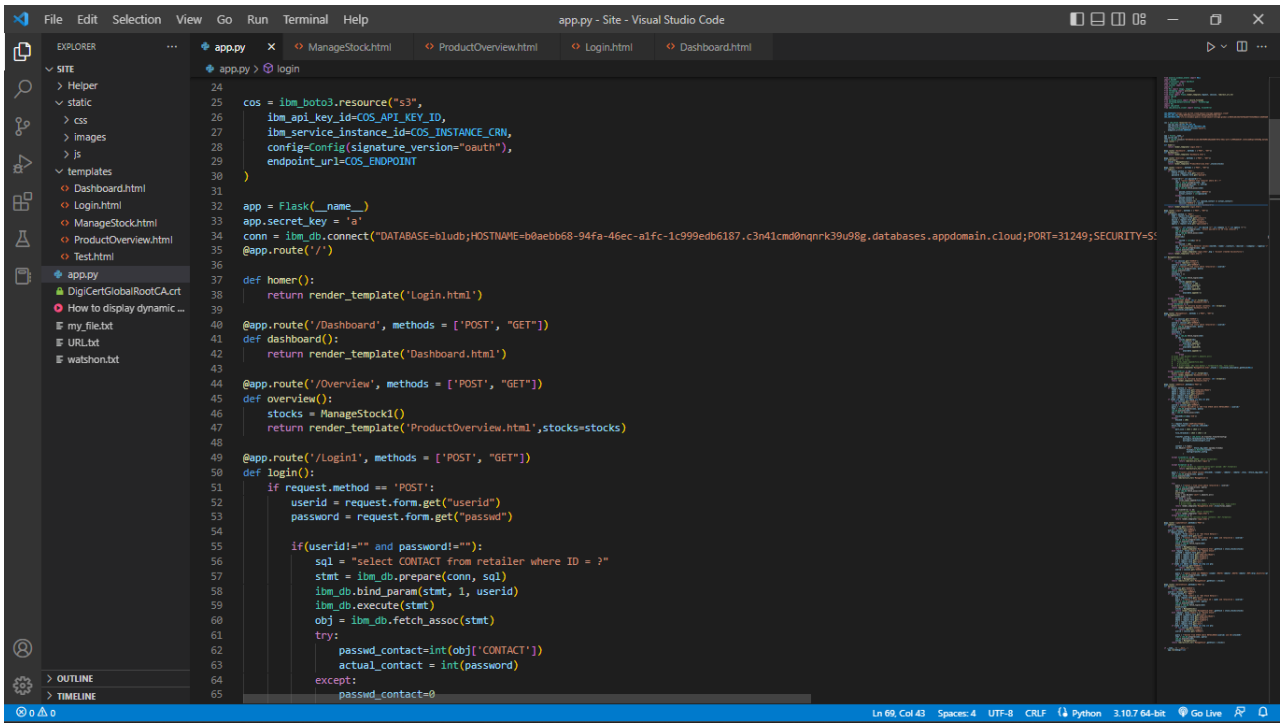
Deploy the App in IBM Cloud

Date	03 November 2022
Team ID	PNT2022TMID17815
Project Name	Inventory Management System for Retailers.




Docker Desktop
Update to latest
Sign in
Containers
Images
Volumes
Dev Environments
BETA
Extensions
BETA
Add Extensions
Images on disk
Last refresh: 1 minute ago
1 images
77.84 MB total size
0 Bytes / 77.84 MB in use
Clean up
Images
Give Feedback
LOCAL
REMOTE REPOSITORIES
Search
In use only
Windows PowerShell (x86)
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\marim> docker -v
Docker version 20.10.17, build 100c701
PS C:\Users\marim>
Sign in
RAM 1.2
v4.12.0


docker Labs
Getting Started
Search
docker/getting-started
2.3k Stars · 5.8k Forks
Getting Started
Getting Started
Our Application
Updating our App
Sharing our App
Persisting our DB
Using Bind Mounts
Multi-Container Apps
Using Docker Compose
Image Building Best Practices
What Next?
Getting Started
The command you just ran
Congratulations! You have started the container for this tutorial! Let's first explain the command that you just ran. In case you forgot, here's the command:
docker run -d -p 80:80 docker/getting-started
You'll notice a few flags being used. Here's some more info on them:
-d - run the container in detached mode (in the background)
-p 80:80 - map port 80 of the host to port 80 in the container
docker/getting-started - the image to use
Pro tip
You can combine single character flags to shorten the full command. As an example, the command above could be written as:
docker run -dp 80:80 docker/getting-started
The Docker Dashboard
Before going too far, we want to highlight the Docker Dashboard, which gives you a quick view of
Table of contents
The command you just ran
The Docker Dashboard
What is a container?
What is a container image?


File Edit Selection View Go Run Terminal Help
app.py - Site - Visual Studio Code
EXPLORER
SITE
Helper
static
css
images
js
templates
Dashboard.html
Login.html
ManageStock.html
ProductOverview.html
Test.html
app.py
DigiCertGlobalRootCA.crt
How to display dynamic ...
my_file.txt
URL.txt
watshon.txt
OUTLINE
TIMELINE
app.py
ManageStock.html
ProductOverview.html
Login.html
Dashboard.html
app.py > login
cos = ibm_boto3.resource("s3",
ibm_api_key_id=COS_API_KEY_ID,
ibm_service_instance_id=COS_INSTANCE_CRN,
config=Config(signature_version="oauth"),
endpoint_url=COS_ENDPOINT
)
app = Flask(__name__)
app.secret_key = 'a'
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=31249;SECURITY=S
@app.route('/')
def homer():
return render_template('Login.html')
@app.route('/Dashboard', methods = ['POST', "GET"])
def dashboard():
return render_template('Dashboard.html')
@app.route('/Overview', methods = ['POST', "GET"])
def overview():
stocks = ManageStock1()
return render_template('ProductOverview.html',stocks=stocks)
@app.route('/Login1', methods = ['POST', "GET"])
def login():
if request.method == 'POST':
userid = request.form.get("userid")
password = request.form.get("passwd")
if(userid!="" and password!=""):
sql = "select CONTACT from retailer where ID = ?"
stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt, 1, userid)
ibm_db.execute(stmt)
obj = ibm_db.fetch_assoc(stmt)
try:
passwd_contact=int(obj['CONTACT'])
actual_contact = int(password)
except:
passwd_contact=0
Ln 69, Col 43 Spaces: 4 UTF-8 CRLF Python 3.10.7 64-bit Go Live