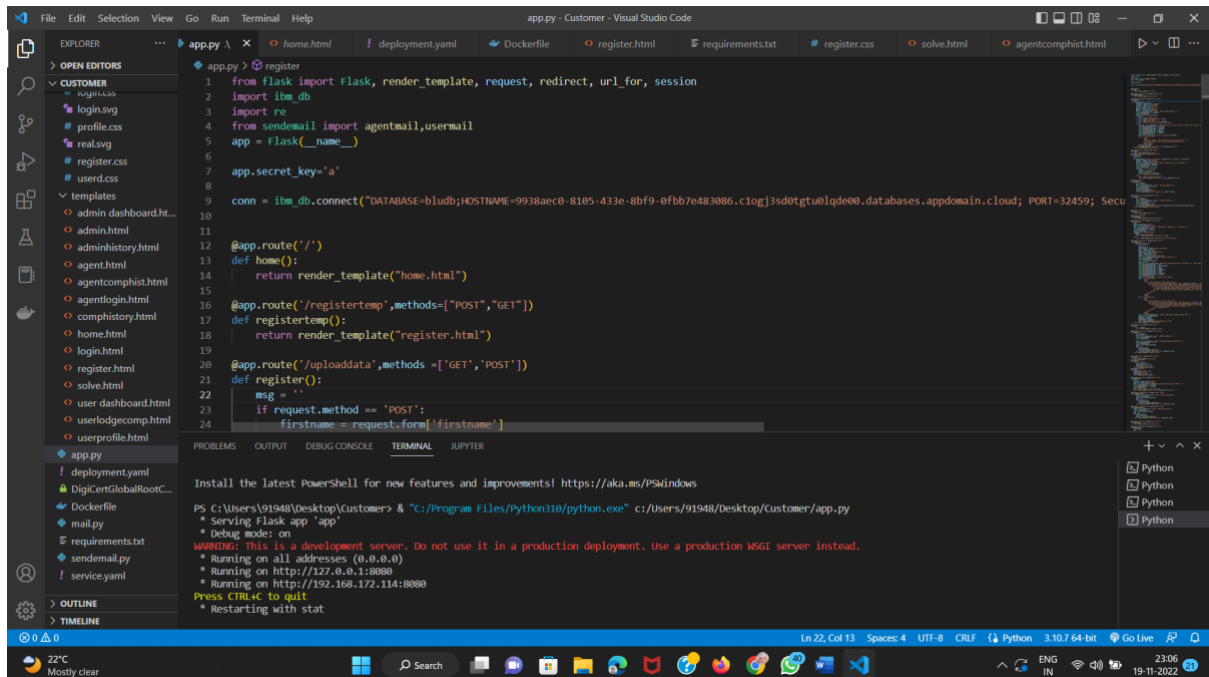


## SETTING UP APPLICATION ENVIRONMENT:

### CREATE FLASK PROJECT:

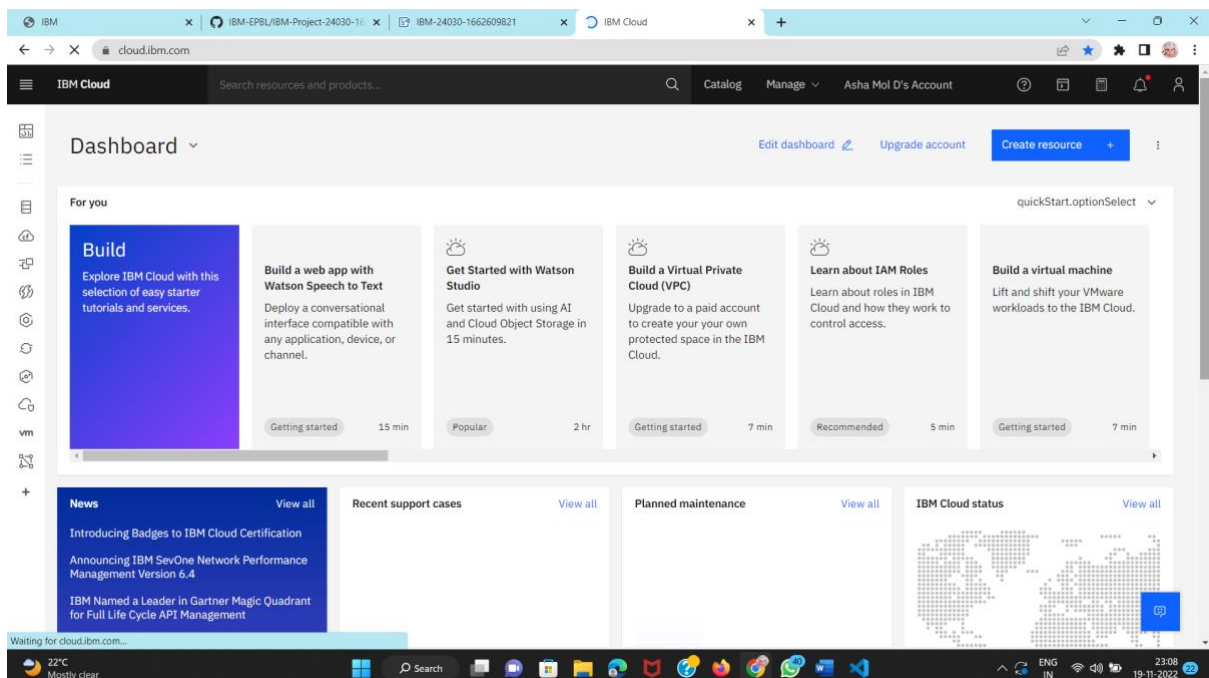


```
app.py > register
1 from flask import Flask, render_template, request, redirect, url_for, session
2 import ibm_db
3 import re
4 from sendmail import agentmail, usermail
5 app = Flask(__name__)
6
7 app.secret_key = 'a'
8
9 conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=9938aec0-8105-433e-8bf9-0fb7e483086.c1ogj3sd0tgu0lqde00.databases.appdomain.cloud; PORT=32459; Secu
10
11
12 @app.route('/')
13 def home():
14     return render_template("home.html")
15
16 @app.route('/registertemp', methods=["POST", "GET"])
17 def registertemp():
18     return render_template("register.html")
19
20 @app.route('/uploaddata', methods=["GET", "POST"])
21 def register():
22     msg = ''
23     if request.method == "POST":
24         firstname = request.form['firstname']
```

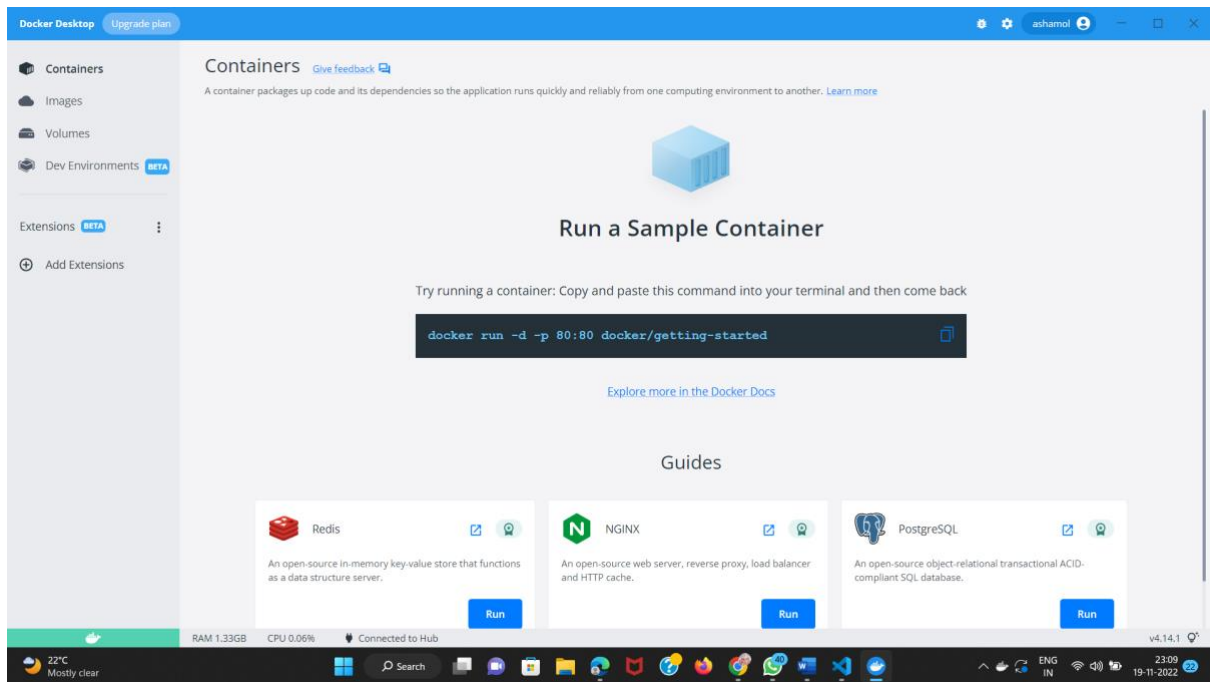
Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\91948\Desktop\Customer> & "C:\Program Files\Python310\python.exe" c:\Users\91948\Desktop\Customer\app.py  
\* Serving Flask app "app"  
\* Debug mode: on  
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.  
\* Running on all addresses (0.0.0.0)  
\* Running on http://127.0.0.1:8080  
\* Running on http://192.168.172.114:8080  
Press CTRL+C to quit  
\* Restarting with stat

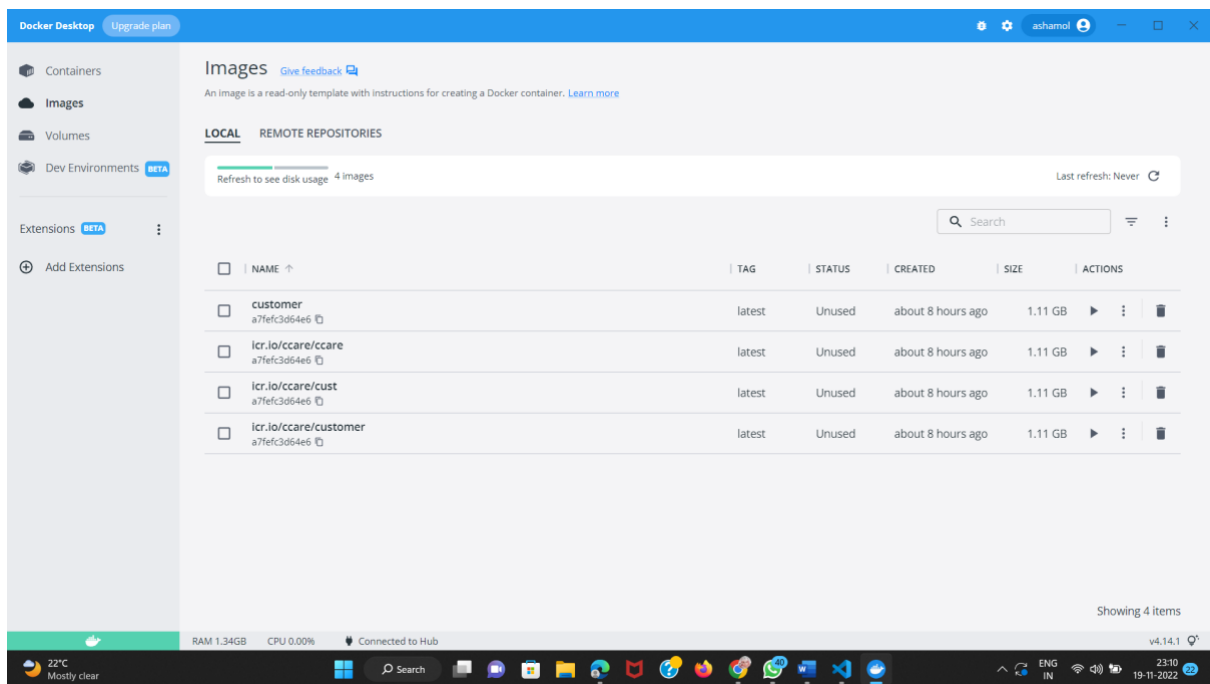
### CREATE IBM CLOUD ACCOUNT:



### DOCKER AND CLI INSTALLATION :



## IMAGES:



## INSTALLATION:

```
Command Prompt
Microsoft Windows [Version 10.0.22000.1219]
(c) Microsoft Corporation. All rights reserved.

C:\Users\91948>Docker

Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default
                        "C:\Users\91948\.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\91948\.docker\ca.pem")
  --tlscert string      Path to TLS certificate file (default
                        "C:\Users\91948\.docker\cert.pem")
  --tlskey string        Path to TLS key file (default
                        "C:\Users\91948\.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)
  compose*             Docker Compose (Docker Inc., v2.12.2)
  config               Manage Docker configs
  container             Manage containers
  context              Manage contexts
  dev*                Docker Dev Environments (Docker Inc., v0.0.3)
  extension*           Manages Docker extensions (Docker Inc., v0.2.13)
  image                Manage images
  manifest              Manage Docker image manifests and manifest lists
  network              Manage networks
  node                 Manage Swarm nodes
  plugin               Manage plugins
  sbom*                View the packaged-based Software Bill Of Materials (SBOM) for an image (Anchore Inc., 0.6.0)
  scan*               Docker Scan (Docker Inc., v0.21.0)
  secret               Manage Docker secrets
  service              Manage services
  stack                Manage Docker stacks
```

```
Command Prompt

system      Manage Docker
trust       Manage trust on Docker images
volume      Manage volumes

Commands:
  attach      Attach local standard input, output, and error streams to a running container
  build       Build an image from a Dockerfile
  commit      Create a new image from a container's changes
  cp          Copy files/folders between a container and the local filesystem
  create      Create a new container
  diff        Inspect changes to files or directories on a container's filesystem
  events      Get real time events from the server
  exec        Run a command in a running container
  export      Export a container's filesystem as a tar archive
  history     Show the history of an image
  images      List images
  import      Import the contents from a tarball to create a filesystem image
  info        Display system-wide information
  inspect     Return low-level information on Docker objects
  kill        Kill one or more running containers
  load        Load an image from a tar archive or STDIN
  login       Log in to a Docker registry
  logout      Log out from a Docker registry
  logs        Fetch the logs of a container
  pause       Pause all processes within one or more containers
  port        List port mappings or a specific mapping for the container
  ps          List containers
  pull        Pull an image or a repository from a registry
  push        Push an image or a repository to a registry
  rename      Rename a container
  restart     Restart one or more containers
  rm          Remove one or more containers
  rmi         Remove one or more images
  run         Run a command in a new container
  save        Save one or more images to a tar archive (streamed to STDOUT by default)
  search      Search the Docker Hub for images
  start       Start one or more stopped containers
  stats       Display a live stream of container(s) resource usage statistics
  stop        Stop one or more running containers
  tag         Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
  top         Display the running processes of a container
  unpause     Unpause all processes within one or more containers
  update      Update configuration of one or more containers
  version     Show the Docker version information
  wait        Block until one or more containers stop, then print their exit codes

Run 'docker COMMAND --help' for more information on a command.

To get more help with docker, check out our guides at https://docs.docker.com/go/guides/
```

## DOCKER CLI:

