CSE5121 - DevOps & MLOps

Activity 3

Create Level 2 of CI /CD pipeline using Docker, Jenkins and Streamlit.

Prerequisites:
VS Code
VS Code Extension – Python
GitHub
Docker
Jenkins

Step 1

Creation of ML Model and Pickle(.pkl) file

Note: After the successful completion, model.pkl will be generated.

```
import pickle
from  sklearn import  datasets
iris=datasets.load_iris()
x=iris.data
y=iris.target

#labels for iris dataset
labels ={
  0: "setosa",
  1: "versicolor",
  2: "virginica"
}

#split the data set
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.50)

#Using decision tree algorithm
from sklearn import tree
classifier=tree.DecisionTreeClassifier()
classifier.fit(x_train,y_train)
predictions=classifier.predict(x_test)

#export the model
pickle.dump(classifier, open('model.pkl','wb'))

#load the model and test with a custom input
model = pickle.load( open('model.pkl','rb'))
x = [[6.7, 3.3, 5.7, 2.1]]
predict = model.predict(x)
print("Hello Worlds")
print(labels[predict[0]])
```

Step 2

Creation of UI using Streamlit

```
import streamlit as st
from PIL import Image
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
import json
import pandas as pd

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))
labels ={
  0: "setosa",
  1: "versicolor",
  2: "virginica"
}

@app.route('/')
def welcome():
    return "Index Page"

@app.route('/predict',methods=['POST'])
def predict(sl,sw,pl,pw):
    prediction=model.predict([[sl,sw,pl,pw]])
    return labels[prediction[0]]
def main():
    st.title("IRIS Prediction")
    html_temp = """
    <div style="background-color:tomato;padding:10px">
    <h2 style="color:white;text-align:center;">Streamlit IRIS Predictor </h2>
    </div>
    """
    st.markdown(html_temp,unsafe_allow_html=True)
    sl = st.text_input("Sepal Length","Type Here")
    sw = st.text_input("Sepal Width","Type Here")
    pl = st.text_input("Petal Length","Type Here")
    pw = st.text_input("Petal Width","Type Here")
    result=""
    if st.button("Predict"):
        result=predict(sl,sw,pl,pw)
    st.success('The output is {}'.format(result))
    if st.button("About"):
        st.text("Lets LEarn")
        st.text("Built with Streamlit")
if __name__=='__main__':
    main()
```

Step 3

Creation of Docker File

FROM ubuntu

FROM python

We copy just the requirements.txt first to leverage Docker cache
COPY ./requirements.txt /app/requirements.txt

WORKDIR /app

RUN pip install -r requirements.txt

COPY . /app

CMD python /app/model.py && python /app/server.py

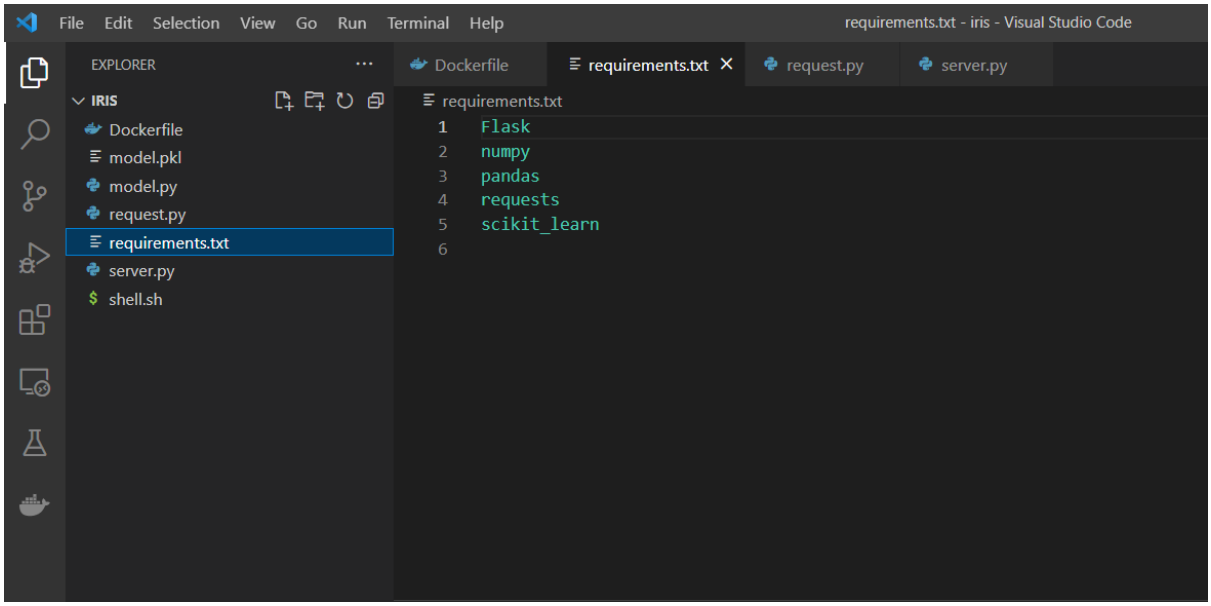
Step 4

Generate requirement file

Note
a. Run the below script in VS Code terminal window of your project directory.
b. After the successful completion, it generates a requirements.txt file.

i.pip install pipreqs

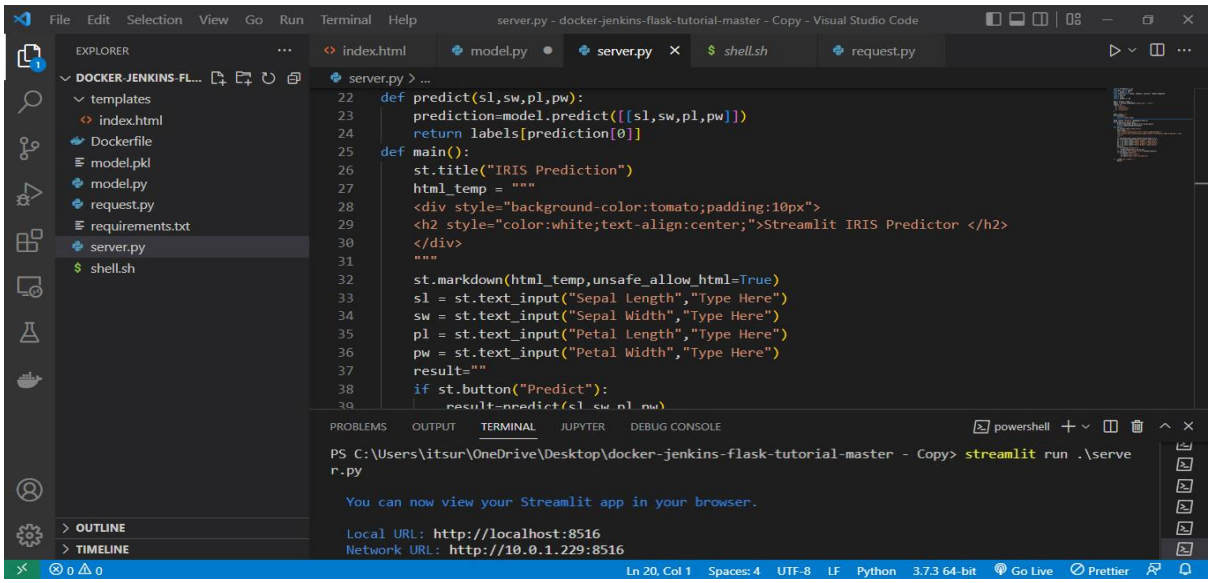
ii. pipreqs .

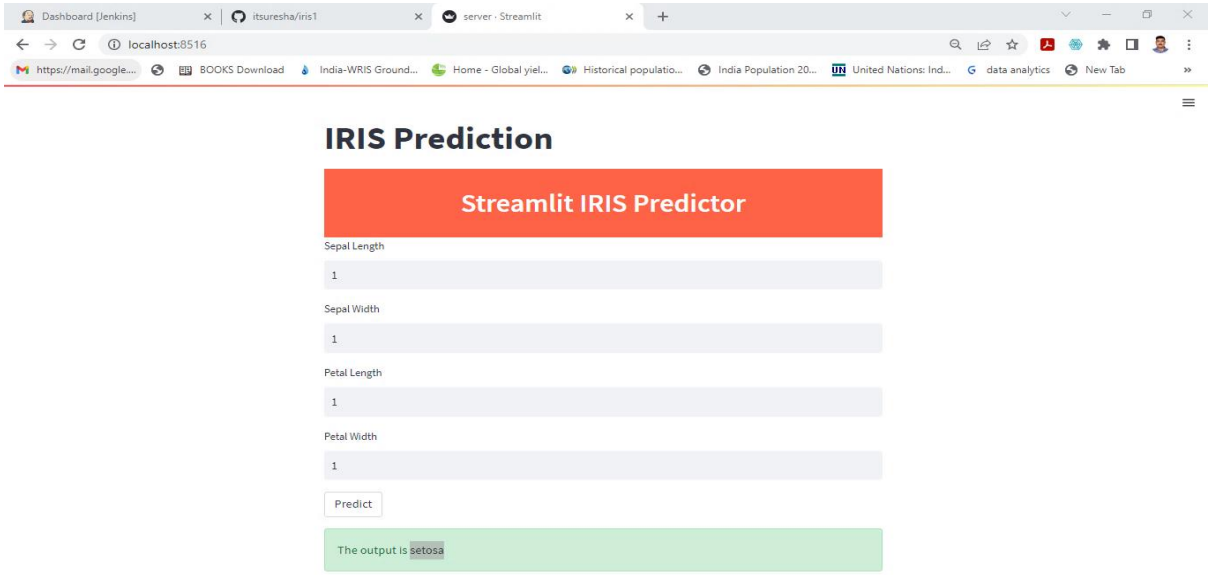


Step 5

Run the following script in VS Code terminal

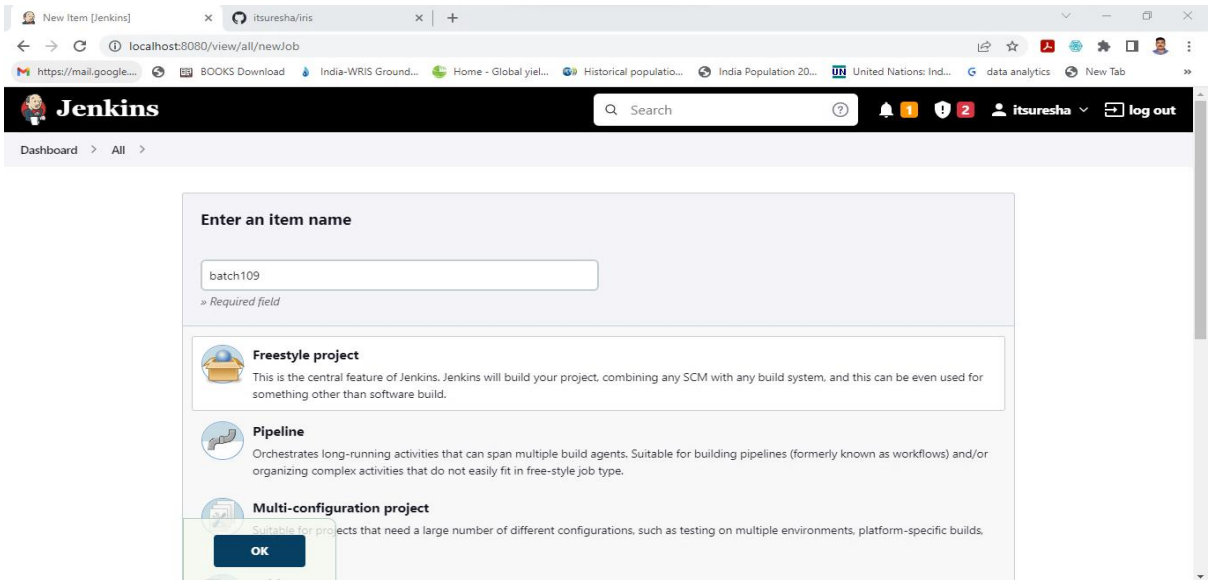
streamlit run server.py

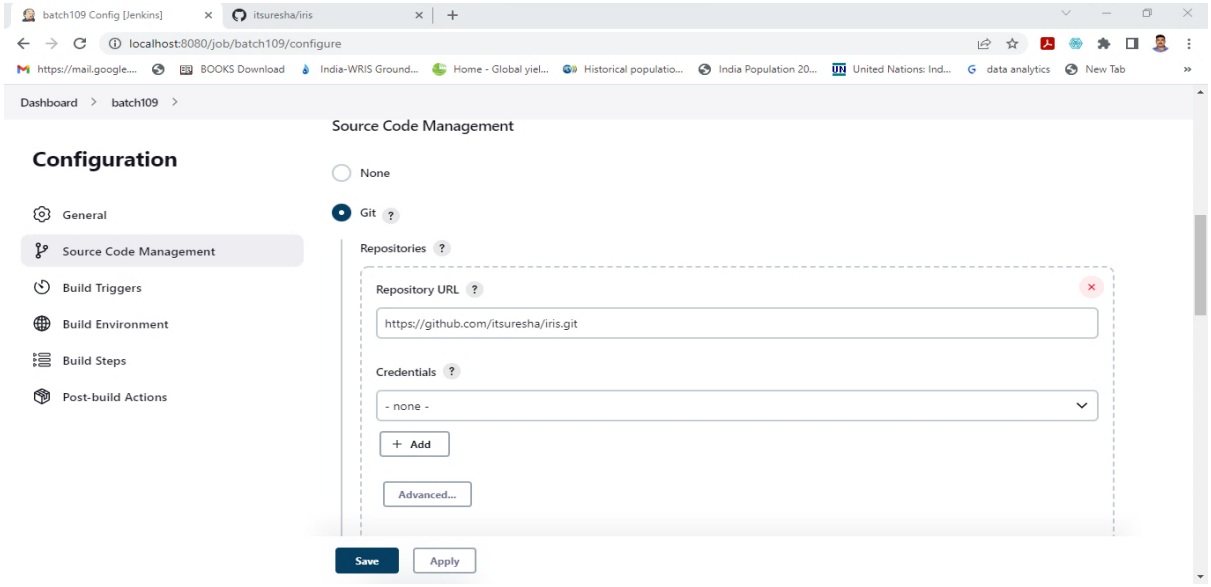


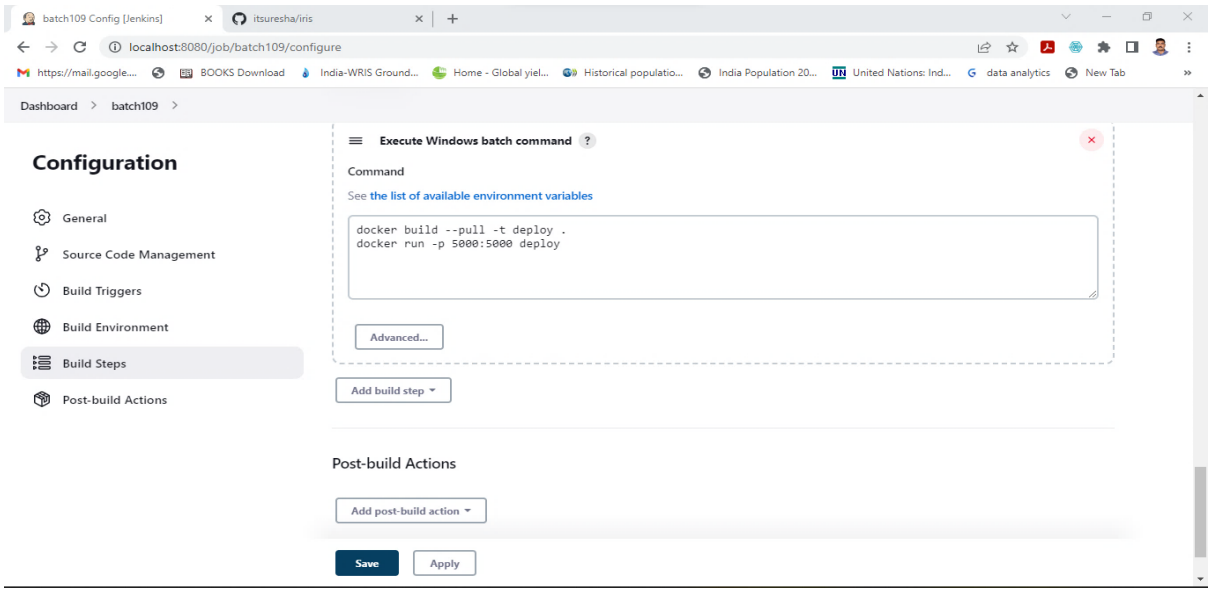


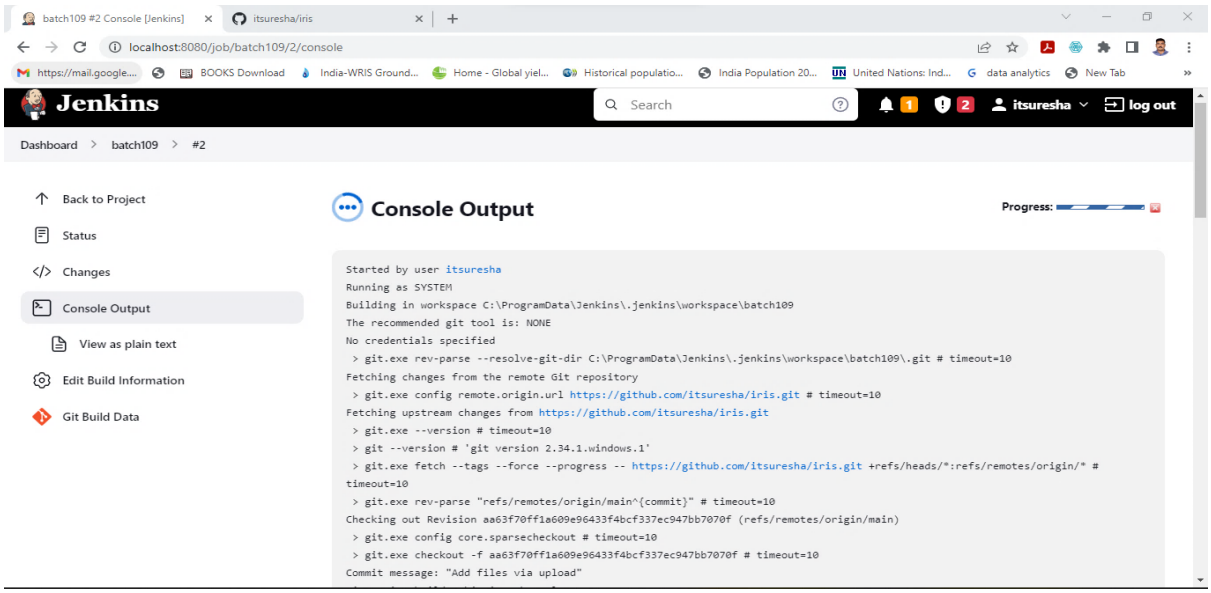
Step 6

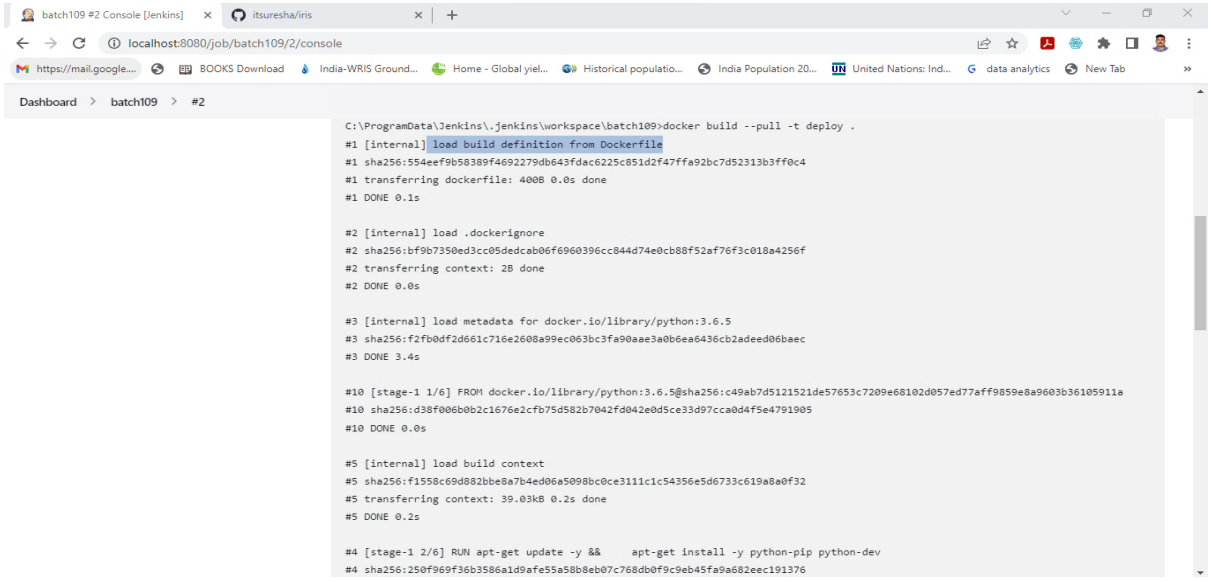
Jenkins Configuration

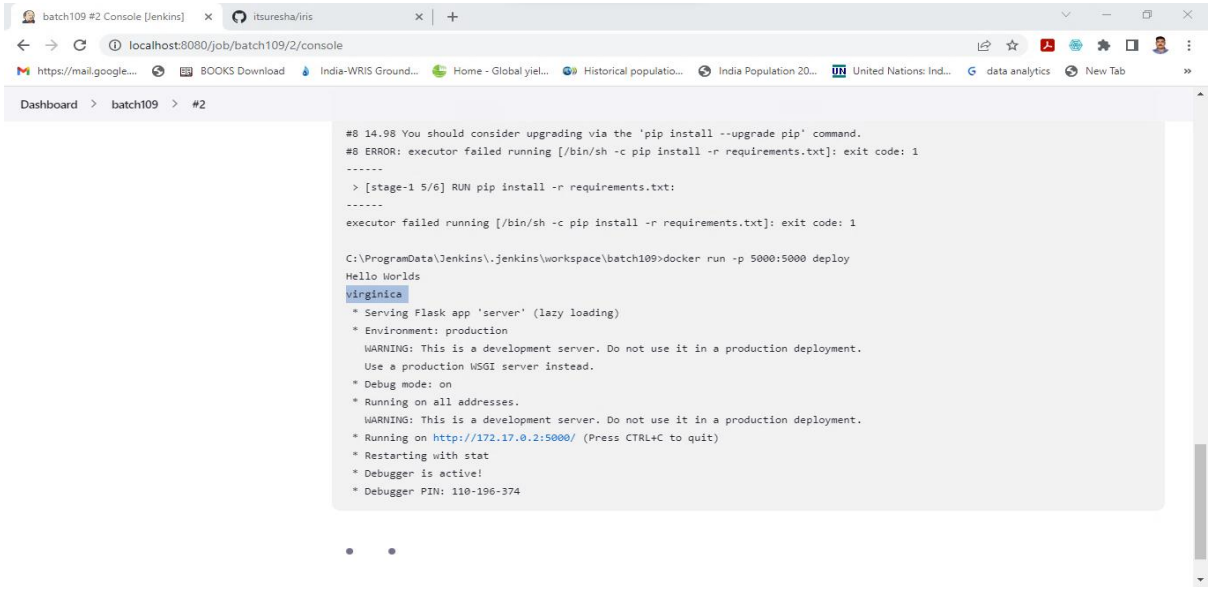

New Item [Jenkins]
itsuresha/iris
localhost:8080/view/all/newJob
Jenkins
Search
itsuresha
log out
Dashboard
All
Enter an item name
batch109
» Required field
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
Multi-configuration project
OK

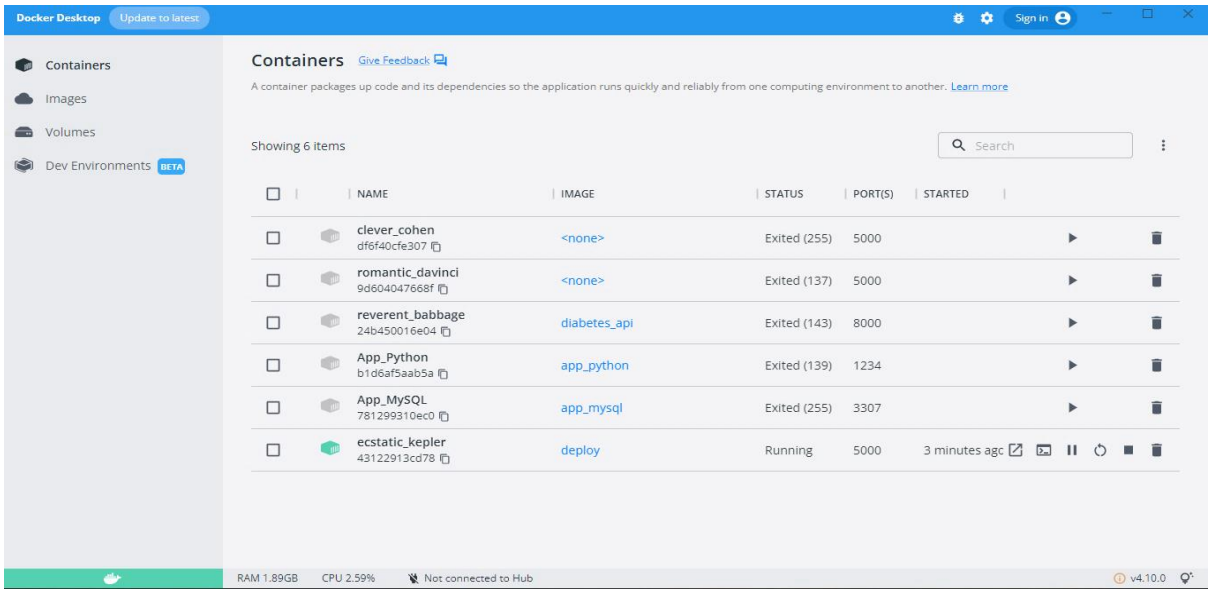

batch109 Config [Jenkins]
itsuresha/iris
localhost:8080/job/batch109/configure
Dashboard
batch109
Configuration
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Source Code Management
None
Git
Repositories
Repository URL
https://github.com/itsuresha/iris.git
Credentials
- none -
Add
Advanced...
Save
Apply


batch109 Config [Jenkins]
itsuresha/iris
localhost:8080/job/batch109/configure
Dashboard
batch109
Configuration
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Execute Windows batch command
Command
See the list of available environment variables
docker build --pull -t deploy .
docker run -p 5000:5000 deploy
Advanced...
Add build step
Post-build Actions
Add post-build action
Save
Apply


batch109 #2 Console [Jenkins]
itsuresha/iris
localhost:8080/job/batch109/2/console
Jenkins
Search
itsuresha
log out
Dashboard > batch109 > #2
Back to Project
Status
Changes
Console Output
View as plain text
Edit Build Information
Git Build Data
Console Output
Progress:
Started by user itsuresha
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\batch109
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\batch109\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/itsuresha/iris.git # timeout=10
Fetching upstream changes from https://github.com/itsuresha/iris.git
> git.exe --version # timeout=10
> git --version # 'git version 2.34.1.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/itsuresha/iris.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision aa63f70ff1a609e96433f4bcf337ec947bb7070f (refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f aa63f70ff1a609e96433f4bcf337ec947bb7070f # timeout=10
Commit message: "Add files via upload"


batch109 #2 Console [Jenkins]
itsuresha/iris
localhost:8080/job/batch109/2/console
Dashboard > batch109 > #2
C:\ProgramData\Jenkins\.jenkins\workspace\batch109>docker build --pull -t deploy .
#1 [internal] load build definition from Dockerfile
#1 sha256:554eef9b58389f4692279db643fdac6225c851d2f47ffa92bc7d52313b3ff0c4
#1 transferring dockerfile: 400B 0.0s done
#1 DONE 0.1s
#2 [internal] load .dockerignore
#2 sha256:bf9b7350ed3cc05dedcab06f6960396cc844d74e0cb88f52af76f3c018a4256f
#2 transferring context: 2B done
#2 DONE 0.0s
#3 [internal] load metadata for docker.io/library/python:3.6.5
#3 sha256:f2fb0df2d661c716e2608a99ec063bc3fa90aae3a0b6ea6436cb2adeed06baec
#3 DONE 3.4s
#10 [stage-1 1/6] FROM docker.io/library/python:3.6.5@sha256:c49ab7d5121521de57653c7209e68102d057ed77aff9859e8a9603b36105911a
#10 sha256:d38f006b0b2c1676e2cfb75d582b7042fd042e0d5ce33d97cca0d4f5e4791905
#10 DONE 0.0s
#5 [internal] load build context
#5 sha256:f1558c69d882bbe8a7b4ed06a5098bc0ce3111c1c54356e5d6733c619a8a0f32
#5 transferring context: 39.03kB 0.2s done
#5 DONE 0.2s
#4 [stage-1 2/6] RUN apt-get update -y &&   apt-get install -y python-pip python-dev
#4 sha256:250f969f36b3586a1d9afe55a58b8eb07c768db0f9c9eb45fa9a682eec191376


batch109 #2 Console [Jenkins]
itsuresha/iris
localhost:8080/job/batch109/2/console
Dashboard > batch109 > #2
#8 14.98 You should consider upgrading via the 'pip install --upgrade pip' command.
#8 ERROR: executor failed running [/bin/sh -c pip install -r requirements.txt]: exit code: 1
> [stage-1 5/6] RUN pip install -r requirements.txt:
executor failed running [/bin/sh -c pip install -r requirements.txt]: exit code: 1
C:\ProgramData\Jenkins\.jenkins\workspace\batch109>docker run -p 5000:5000 deploy
Hello Worlds
virginica
* Serving Flask app 'server' (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Running on all addresses.
WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://172.17.0.2:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 110-196-374


Docker Desktop
Update to latest
Sign in
Containers
Images
Volumes
Dev Environments
BETA
Containers
Give Feedback
A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more
Showing 6 items
Search
NAME
IMAGE
STATUS
PORT(S)
STARTED
clever_cohen
df6f40cfe307
<none>
Exited (255)
5000
romantic_davinci
9d604047668f
<none>
Exited (137)
5000
reverent_babbage
24b450016e04
diabetes_api
Exited (143)
8000
App_Python
b1d6af5aab5a
app_python
Exited (139)
1234
App_MySQL
781299310ec0
app_mysql
Exited (255)
3307
ecstatic_kepler
43122913cd78
deploy
Running
5000
3 minutes ago
RAM 1.89GB
CPU 2.59%
Not connected to Hub
v4.10.0