DevOps

Assignment 3

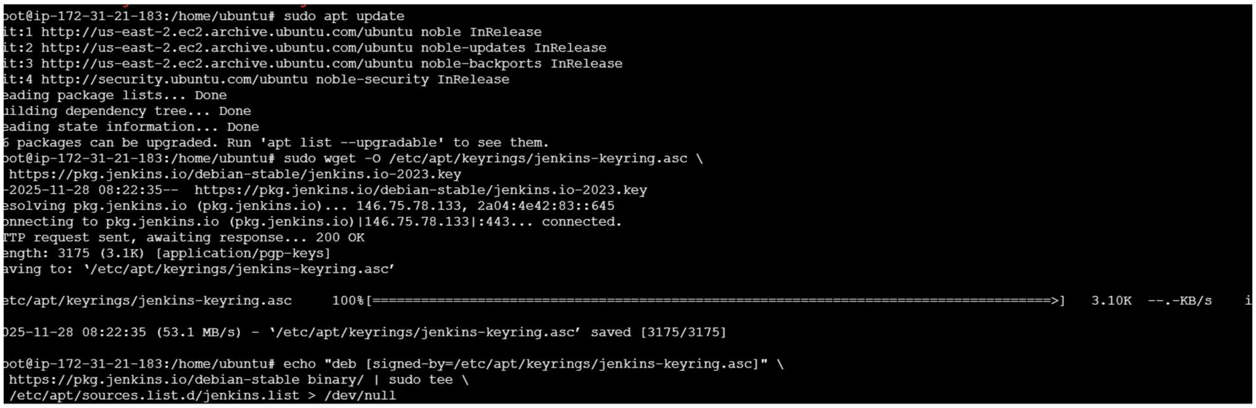
Submitted by: Irum Imtiaz

Reg no: FA22-BCT-011

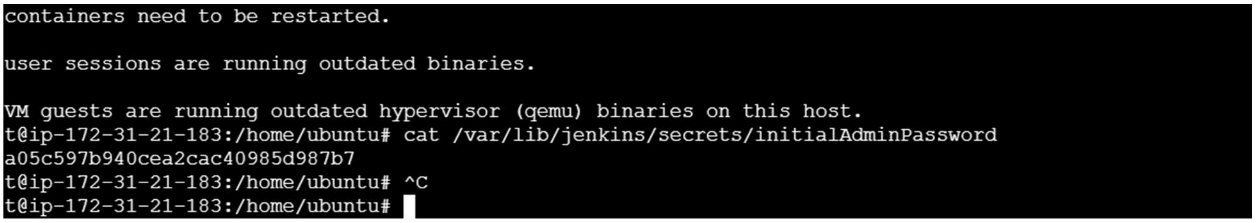
Table of Contents

Setup

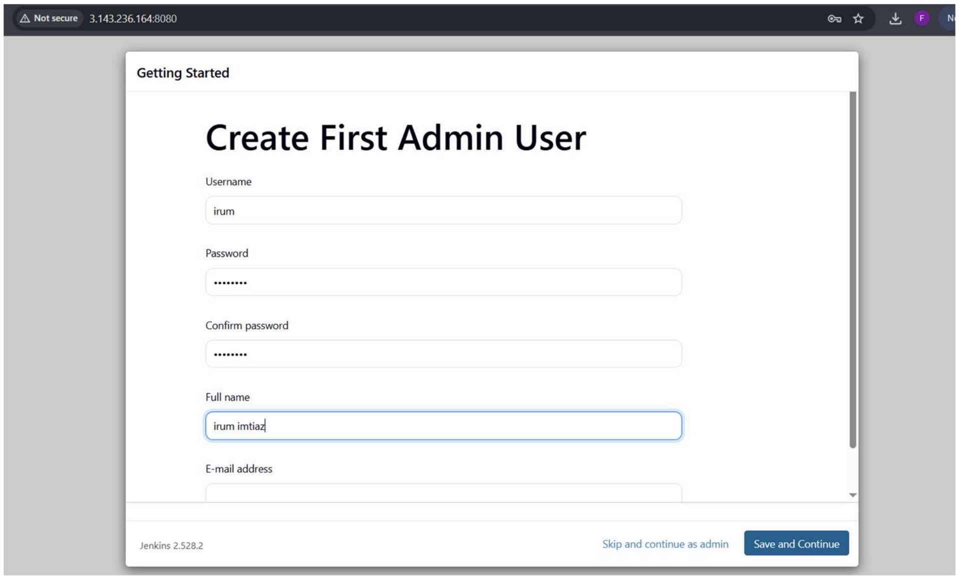
Installed docker,git,python and jenkins



```
oot@ip-172-31-21-183:/home/ubuntu# sudo apt update
it:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble InRelease
it:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
it:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
it:4 http://security.ubuntu.com/ubuntu noble-security InRelease
eading package lists... Done
uilding dependency tree... Done
eading state information... Done
6 packages can be upgraded. Run 'apt list --upgradable' to see them.
oot@ip-172-31-21-183:/home/ubuntu# sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc \
 https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
-2025-11-28 08:22:35--  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
esolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.78.133, 2a04:4e42:83::645
onnecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.78.133|:443... connected.
TTP request sent, awaiting response... 200 OK
ength: 3175 (3.1K) [application/pgp-keys]
aving to: '/etc/apt/keyrings/jenkins-keyring.asc'

etc/apt/keyrings/jenkins-keyring.asc     100%[==================================================================================>]   3.10K  --.-KB/s    i

025-11-28 08:22:35 (53.1 MB/s) - '/etc/apt/keyrings/jenkins-keyring.asc' saved [3175/3175]

oot@ip-172-31-21-183:/home/ubuntu# echo "deb [signed-by=/etc/apt/keyrings/jenkins-keyring.asc]" \
 https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
 /etc/apt/sources.list.d/jenkins.list > /dev/null
```



```
containers need to be restarted.

user sessions are running outdated binaries.

VM guests are running outdated hypervisor (qemu) binaries on this host.
t@ip-172-31-21-183:/home/ubuntu# cat /var/lib/jenkins/secrets/initialAdminPassword
a05c597b940cea2cac40985d987b7
t@ip-172-31-21-183:/home/ubuntu# ^C
t@ip-172-31-21-183:/home/ubuntu#
```



Not secure 3.143.236.164:8080

Getting Started

Create First Admin User

Username: irum

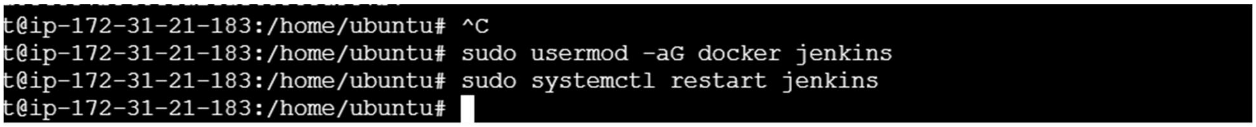
Password: ••••••••

Confirm password: ••••••••

Full name: irum imtiaz

E-mail address:

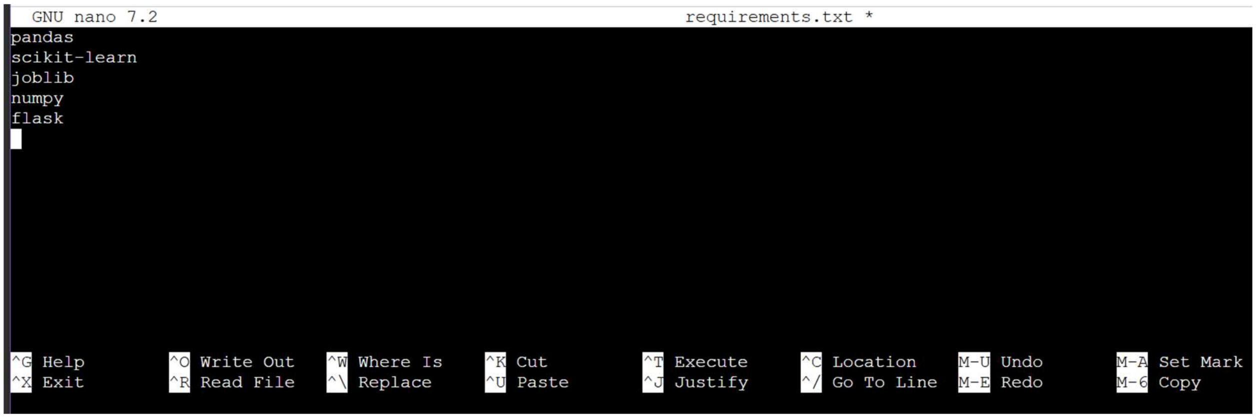
Give permission to jenkins to run docker



```
t@ip-172-31-21-183:/home/ubuntu# ^C
t@ip-172-31-21-183:/home/ubuntu# sudo usermod -aG docker jenkins
t@ip-172-31-21-183:/home/ubuntu# sudo systemctl restart jenkins
t@ip-172-31-21-183:/home/ubuntu#
```

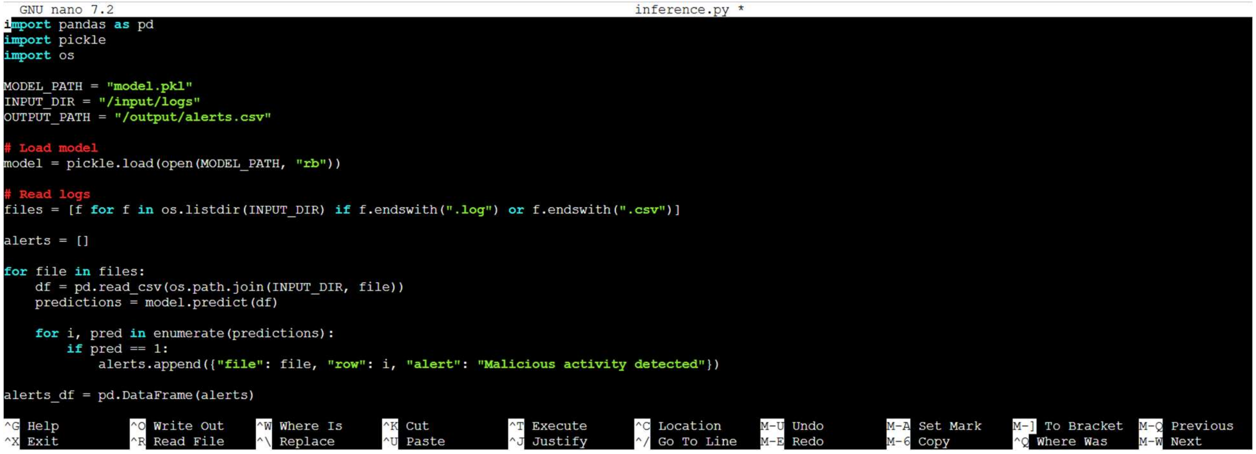
a) Write a Dockerfile that packages the entire malware detection application, including installing dependencies using pip install.

Create requirements.txt



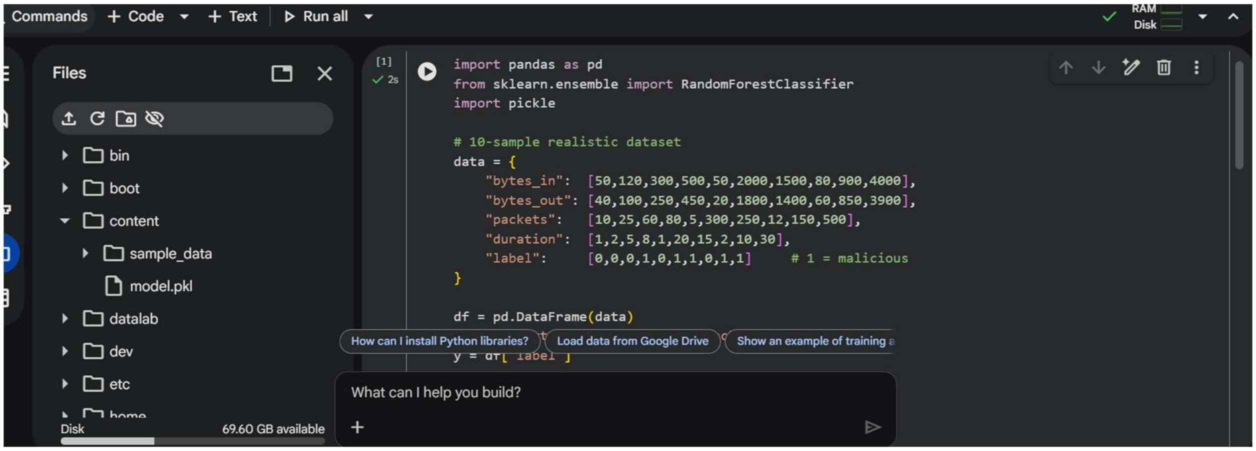
```
  GNU nano 7.2                                      requirements.txt *
pandas
scikit-learn
joblib
numpy
flask



^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location     M-U Undo        M-A Set Mark
^X Exit         ^R Read File    ^\ Replace      ^U Paste        ^J Justify      ^/ Go To Line   M-E Redo        M-6 Copy
```

Create inference.py

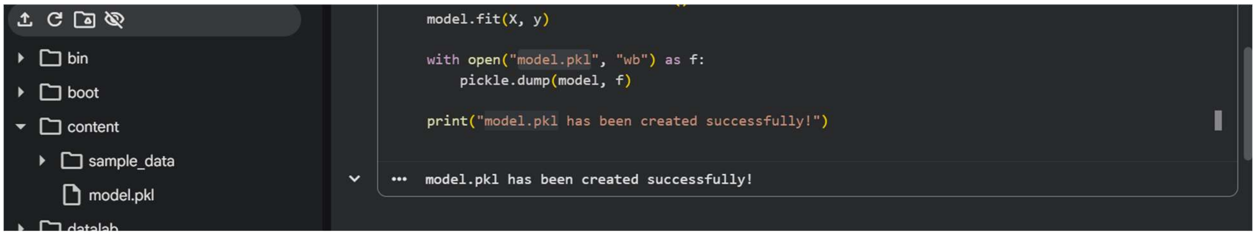


```
  GNU nano 7.2                                      inference.py *
import pandas as pd
import pickle
import os

MODEL_PATH = "model.pkl"
INPUT_DIR = "/input/logs"
OUTPUT_PATH = "/output/alerts.csv"

# Load model
model = pickle.load(open(MODEL_PATH, "rb"))

# Read logs
files = [f for f in os.listdir(INPUT_DIR) if f.endswith(".log") or f.endswith(".csv")]

alerts = []

for file in files:
    df = pd.read_csv(os.path.join(INPUT_DIR, file))
    predictions = model.predict(df)

    for i, pred in enumerate(predictions):
        if pred == 1:
            alerts.append({"file": file, "row": i, "alert": "Malicious activity detected"})

alerts_df = pd.DataFrame(alerts)

^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket  M-Q Previous
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^Q Where Was    M-W Next
```

Create model.py to train model in google collab and upload that model in github

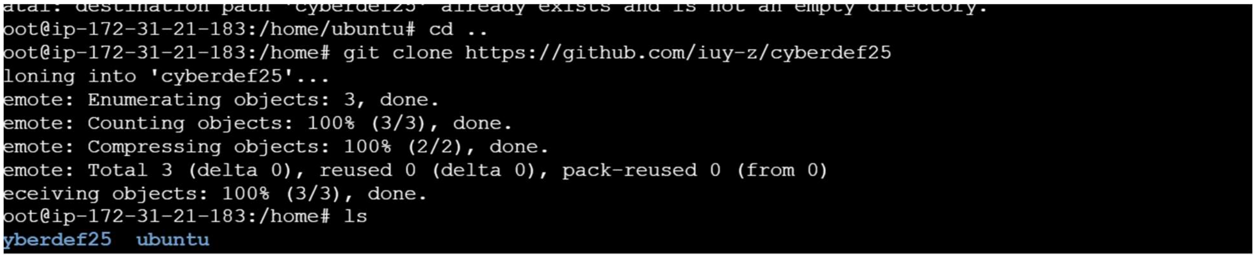

Commands
+ Code
+ Text
Run all
RAM
Disk
Files
bin
boot
content
sample_data
model.pkl
datalab
dev
etc
home
Disk
69.60 GB available
[1]
2s
How can I install Python libraries?
Load data from Google Drive
Show an example of training a
What can I help you build?

```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
import pickle

# 10-sample realistic dataset
data = {
    "bytes_in":  [50,120,300,500,50,2000,1500,80,900,4000],
    "bytes_out": [40,100,250,450,20,1800,1400,60,850,3900],
    "packets":   [10,25,60,80,5,300,250,12,150,500],
    "duration":  [1,2,5,8,1,20,15,2,10,30],
    "label":     [0,0,0,1,0,1,1,0,1,1]     # 1 = malicious
}

df = pd.DataFrame(data)

y = df["label"]
```



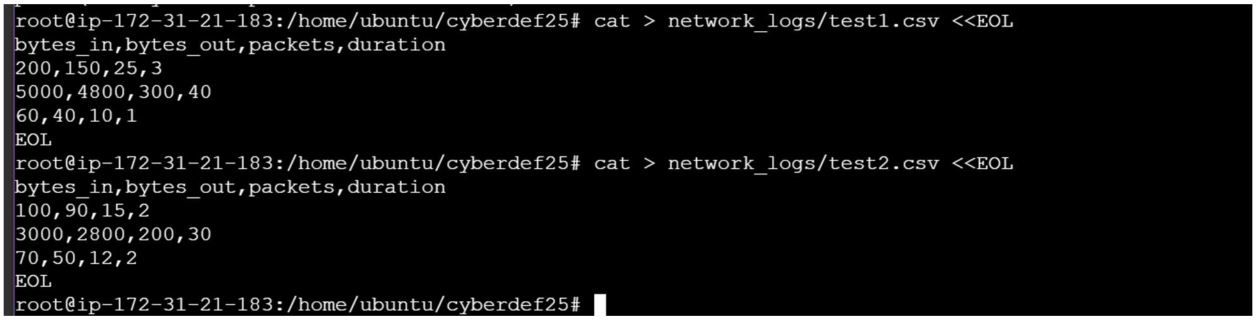
```
model.fit(X, y)

with open("model.pkl", "wb") as f:
    pickle.dump(model, f)

print("model.pkl has been created successfully!")
```

```
model.pkl has been created successfully!
```



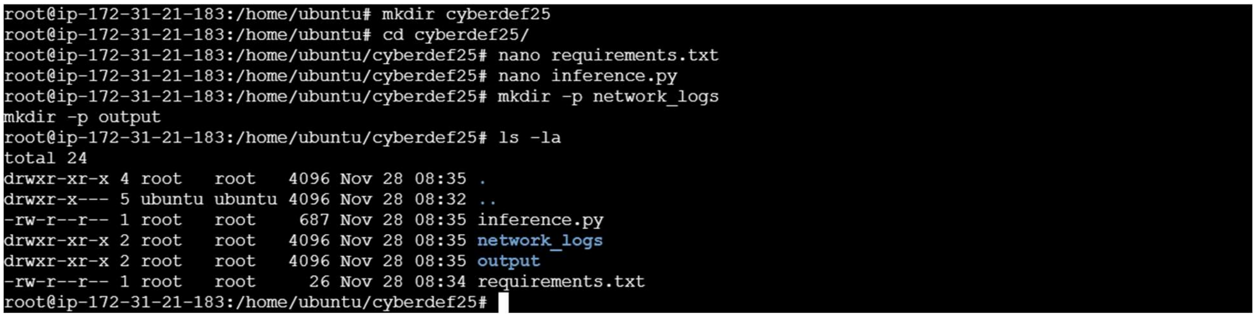
```
atal: destination path 'cyberdef25' already exists and is not an empty directory.
oot@ip-172-31-21-183:/home/ubuntu# cd ..
oot@ip-172-31-21-183:/home# git clone https://github.com/iuy-z/cyberdef25
loning into 'cyberdef25'...
emote: Enumerating objects: 3, done.
emote: Counting objects: 100% (3/3), done.
emote: Compressing objects: 100% (2/2), done.
emote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
eceiving objects: 100% (3/3), done.
oot@ip-172-31-21-183:/home# ls
yberdef25  ubuntu
```

Created fake network logs



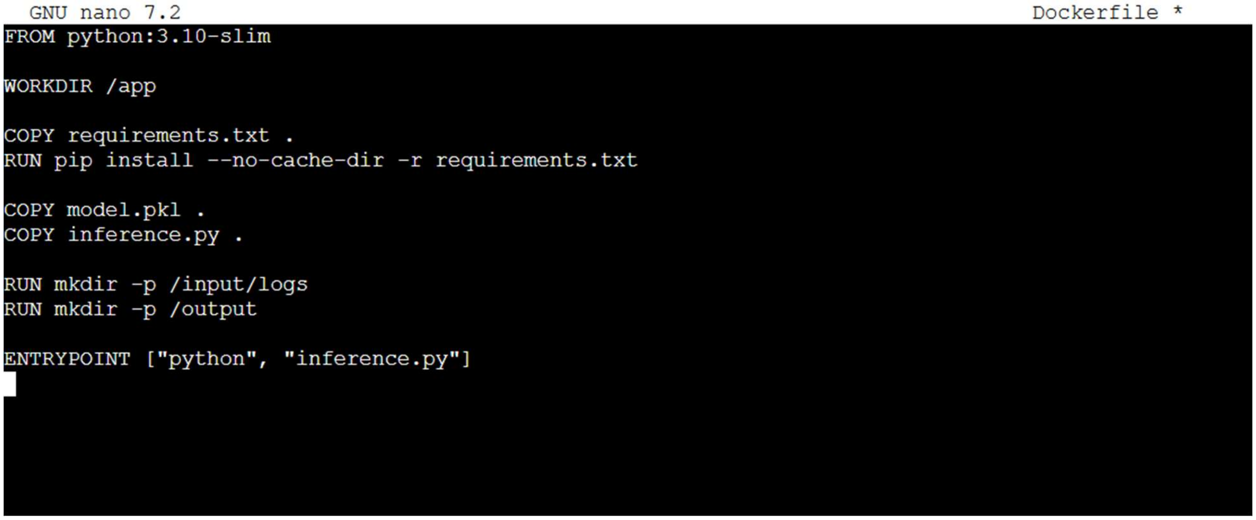
```
root@ip-172-31-21-183:/home/ubuntu/cyberdef25# cat > network_logs/test1.csv <<EOL
bytes_in,bytes_out,packets,duration
200,150,25,3
5000,4800,300,40
60,40,10,1
EOL
root@ip-172-31-21-183:/home/ubuntu/cyberdef25# cat > network_logs/test2.csv <<EOL
bytes_in,bytes_out,packets,duration
100,90,15,2
3000,2800,200,30
70,50,12,2
EOL
root@ip-172-31-21-183:/home/ubuntu/cyberdef25#
```

Create directories output folder



```
root@ip-172-31-21-183:/home/ubuntu# mkdir cyberdef25
root@ip-172-31-21-183:/home/ubuntu# cd cyberdef25/
root@ip-172-31-21-183:/home/ubuntu/cyberdef25# nano requirements.txt
root@ip-172-31-21-183:/home/ubuntu/cyberdef25# nano inference.py
root@ip-172-31-21-183:/home/ubuntu/cyberdef25# mkdir -p network_logs
mkdir -p output
root@ip-172-31-21-183:/home/ubuntu/cyberdef25# ls -la
total 24
drwxr-xr-x 4 root   root   4096 Nov 28 08:35 .
drwxr-x--- 5 ubuntu ubuntu 4096 Nov 28 08:32 ..
-rw-r--r-- 1 root   root    687 Nov 28 08:35 inference.py
drwxr-xr-x 2 root   root   4096 Nov 28 08:35 network_logs
drwxr-xr-x 2 root   root   4096 Nov 28 08:35 output
-rw-r--r-- 1 root   root     26 Nov 28 08:34 requirements.txt
root@ip-172-31-21-183:/home/ubuntu/cyberdef25#
```

Create dockerfile



```
FROM python:3.10-slim

WORKDIR /app

COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt

COPY model.pkl .
COPY inference.py .

RUN mkdir -p /input/logs
RUN mkdir -p /output

ENTRYPOINT ["python", "inference.py"]
```

FROM python:3.10-slim

WORKDIR /app

COPY requirements.txt .

RUN pip install --no-cache-dir -r requirements.txt

COPY model.pkl .

COPY inference.py .

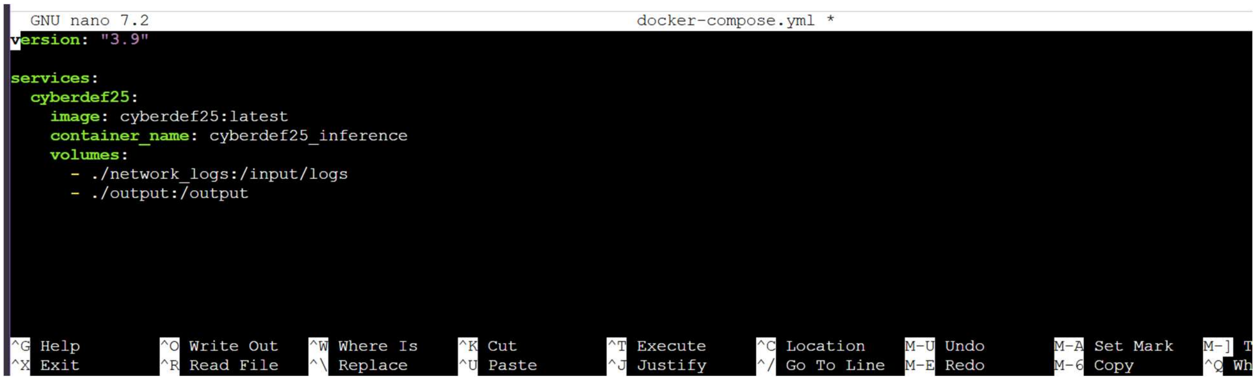
RUN mkdir -p /input/logs

RUN mkdir -p /output

ENTRYPOINT ["python", "inference.py"]

b) The CYBER-DEF25 Challenge requires you to submit a docker-compose file that mounts the host folder ./network_logs/ to provide test log files. Write such a docker-compose file.

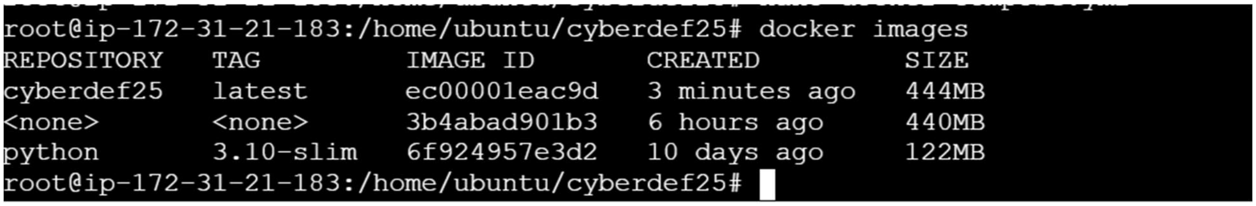
Create docker-compose.yml file



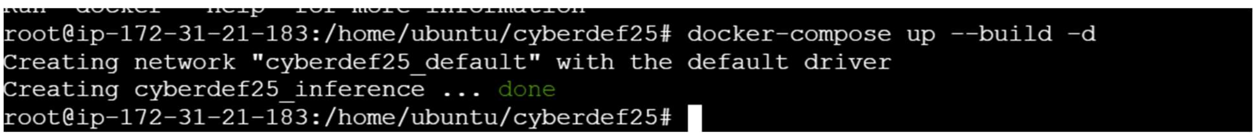
```
version: "3.9"

services:
  cyberdef25:
    image: cyberdef25:latest
    container_name: cyberdef25_inference
    volumes:
      - ./network_logs:/input/logs
      - ./output:/output
```

```
version: "3.9"


services:

 cyberdef25:

   image: cyberdef25:latest

   container_name: cyberdef25_inference

   volumes:

    - ./network_logs:/input/logs

    - ./output:/output
```

Build docker image

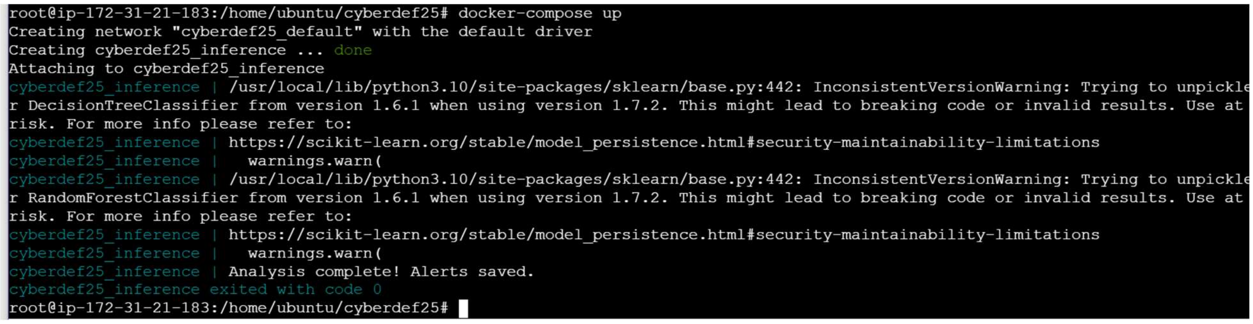


```
root@ip-172-31-21-183:/home/ubuntu/cyberdef25# docker images
REPOSITORY   TAG         IMAGE ID       CREATED         SIZE
cyberdef25   latest      ec00001eac9d   3 minutes ago   444MB
<none>       <none>      3b4abad901b3   6 hours ago     440MB
python       3.10-slim   6f924957e3d2   10 days ago     122MB
root@ip-172-31-21-183:/home/ubuntu/cyberdef25# 
```

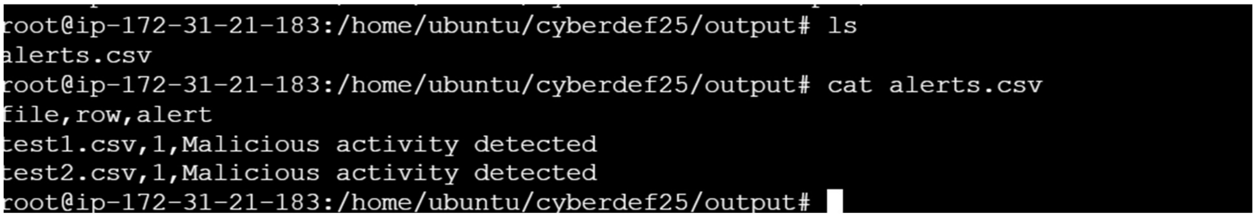


```
Run 'docker --help' for more information
root@ip-172-31-21-183:/home/ubuntu/cyberdef25# docker-compose up --build -d
Creating network "cyberdef25_default" with the default driver
Creating cyberdef25_inference ... done
root@ip-172-31-21-183:/home/ubuntu/cyberdef25#
```

Docker compose

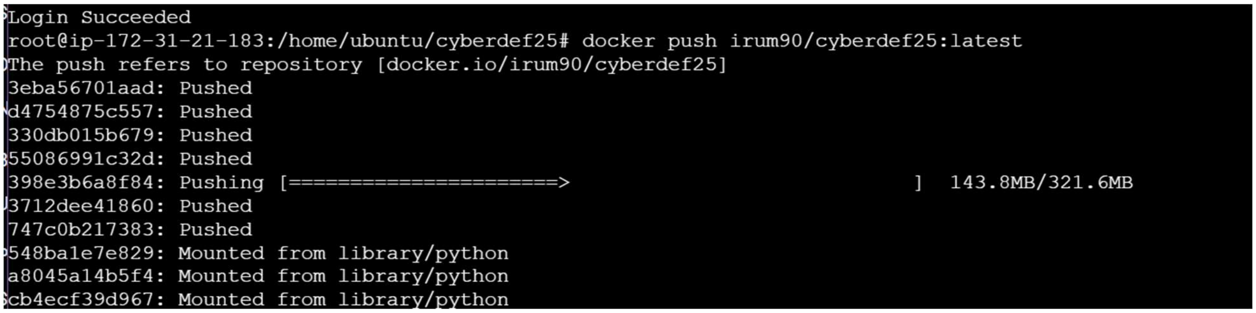


```
root@ip-172-31-21-183:/home/ubuntu/cyberdef25# docker-compose up
Creating network "cyberdef25_default" with the default driver
Creating cyberdef25_inference ... done
Attaching to cyberdef25_inference
cyberdef25_inference | /usr/local/lib/python3.10/site-packages/sklearn/base.py:442: InconsistentVersionWarning: Trying to unpickle
r DecisionTreeClassifier from version 1.6.1 when using version 1.7.2. This might lead to breaking code or invalid results. Use at
risk. For more info please refer to:
cyberdef25_inference | https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
cyberdef25_inference |   warnings.warn(
cyberdef25_inference | /usr/local/lib/python3.10/site-packages/sklearn/base.py:442: InconsistentVersionWarning: Trying to unpickle
r RandomForestClassifier from version 1.6.1 when using version 1.7.2. This might lead to breaking code or invalid results. Use at
risk. For more info please refer to:
cyberdef25_inference | https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
cyberdef25_inference |   warnings.warn(
cyberdef25_inference | Analysis complete! Alerts saved.
cyberdef25_inference exited with code 0
root@ip-172-31-21-183:/home/ubuntu/cyberdef25#
```

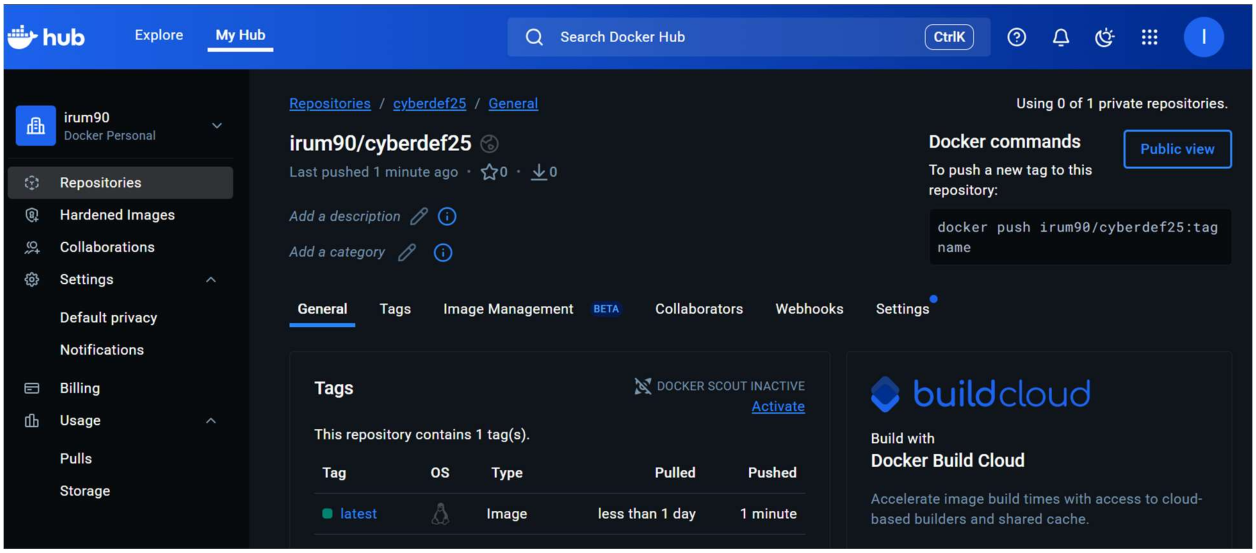


```
root@ip-172-31-21-183:/home/ubuntu/cyberdef25/output# ls
alerts.csv
root@ip-172-31-21-183:/home/ubuntu/cyberdef25/output# cat alerts.csv
file,row,alert
test1.csv,1,Malicious activity detected
test2.csv,1,Malicious activity detected
root@ip-172-31-21-183:/home/ubuntu/cyberdef25/output#
```

Pushing to docker hub



```
Login Succeeded
root@ip-172-31-21-183:/home/ubuntu/cyberdef25# docker push irum90/cyberdef25:latest
The push refers to repository [docker.io/irum90/cyberdef25]
3eba56701aad: Pushed
d47548 75c557: Pushed
330db015b679: Pushed
55086991c32d: Pushed
398e3b6a8f84: Pushing [======================>                              ]  143.8MB/321.6MB
3712dee41860: Pushed
747c0b217383: Pushed
548ba1e7e829: Mounted from library/python
a8045a14b5f4: Mounted from library/python
cb4ecf39d967: Mounted from library/python
```



Repositories / cyberdef25 / General

Using 0 of 1 private repositories.

irum90/cyberdef25

Last pushed 1 minute ago · ☆0 · ⤓0

Add a description

Add a category

Docker commands

To push a new tag to this repository:

```
docker push irum90/cyberdef25:tagname
```

Tags

DOCKER SCOUT INACTIVE Activate

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	less than 1 day	1 minute

buildcloud

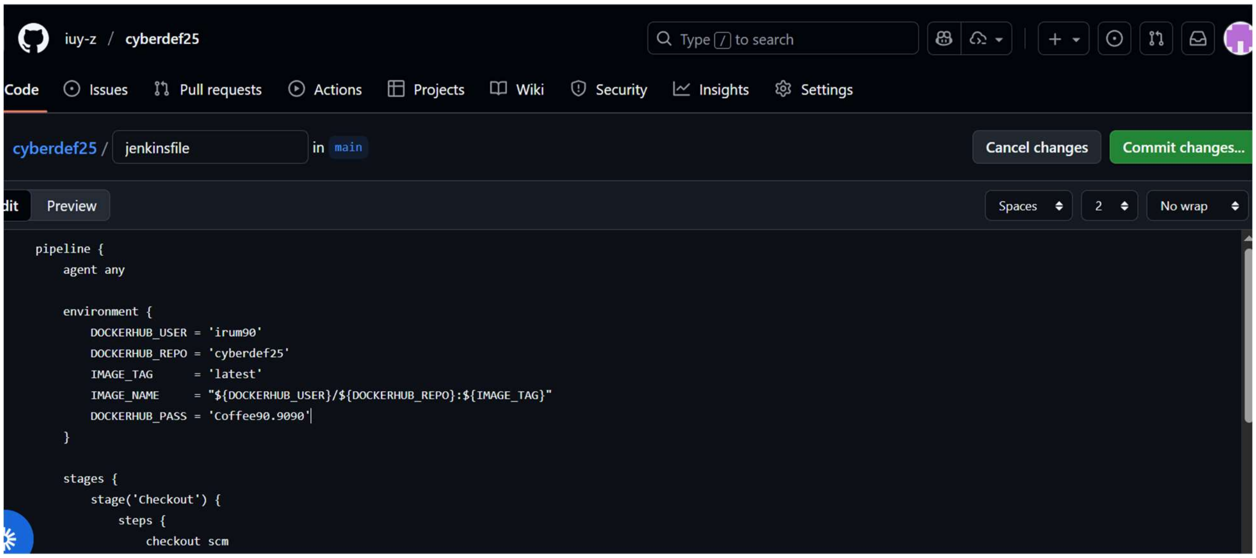
Build with

Docker Build Cloud

Accelerate image build times with access to cloud-based builders and shared cache.

c) Write a Jenkins pipeline script to build the image, push it to Docker Hub and run it using docker compose file.

Created a Jenkins file



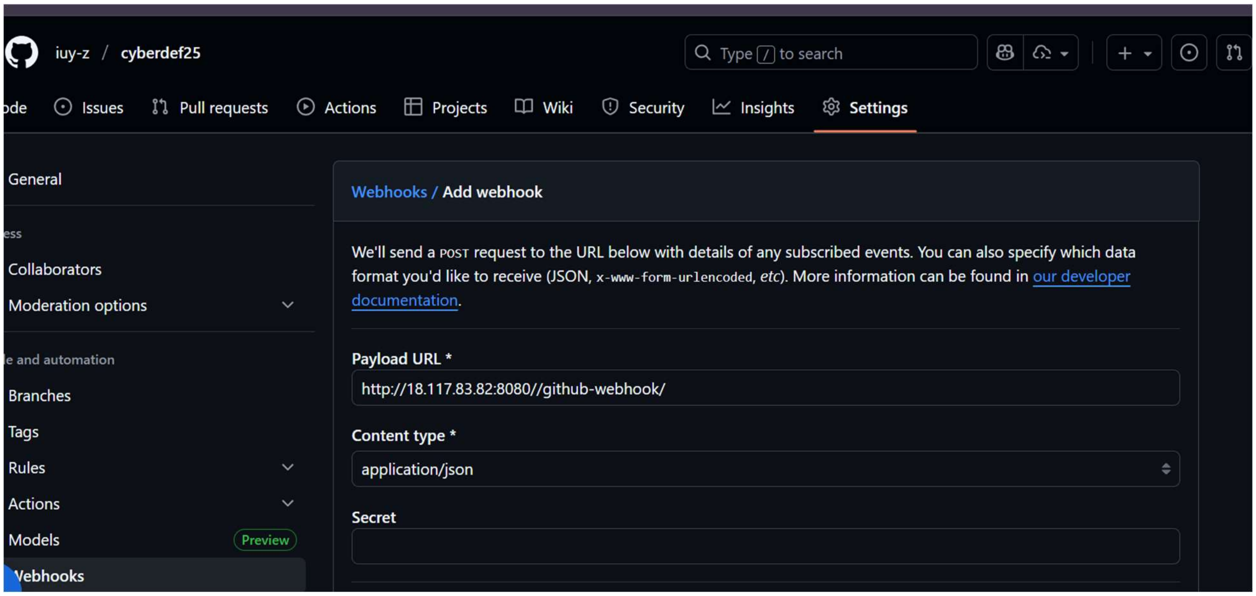
cyberdef25 / jenkinsfile in main

```
pipeline {
    agent any

    environment {
        DOCKERHUB_USER = 'irum90'
        DOCKERHUB_REPO = 'cyberdef25'
        IMAGE_TAG      = 'latest'
        IMAGE_NAME     = "${DOCKERHUB_USER}/${DOCKERHUB_REPO}:${IMAGE_TAG}"
        DOCKERHUB_PASS = 'Coffee90.9090'
    }

    stages {
        stage('Checkout') {
            steps {
                checkout scm
```

pipeline {

```
  agent any

  environment {
    DOCKERHUB_USER = 'irum90'
    DOCKERHUB_REPO = 'cyberdef25'
    IMAGE_TAG      = 'latest'
    IMAGE_NAME     = "${DOCKERHUB_USER}/${DOCKERHUB_REPO}:${IMAGE_TAG}"
  }

  stages {
    stage('Build Docker Image') {
      steps {
        echo "Building Docker image ${IMAGE_NAME}..."
        sh "docker build -t ${IMAGE_NAME} ."
      }
    }

    stage('Login to Docker Hub') {
      steps {
        // Using Jenkins credentials with ID 'dockerhub_creds'
        echo "Logging in to Docker Hub..."
        withCredentials([usernamePassword(credentialsId: 'dockerhub_creds', usernameVariable: 'DOCKER_USER', passwordVariable: 'DOCKER_PASS')]) {
          sh 'echo "$DOCKER_PASS" | docker login -u "$DOCKER_USER" --password-stdin'
        }
      }
    }

    stage('Push Docker Image') {
      steps {
        echo "Pushing Docker image to Docker Hub..."
        sh "docker push ${IMAGE_NAME}"
      }
    }

    stage('Run Container with Docker Compose') {
      steps {
        echo "Running container using docker-compose..."
        sh '''
          docker-compose down || true
          docker-compose up --build -d
        '''
      }
    }
  }

  post {
    always {
      echo "Jenkins pipeline finished successfully."
    }
  }
}
```

Added a webhook



Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, *etc*). More information can be found in our developer documentation.

Payload URL *

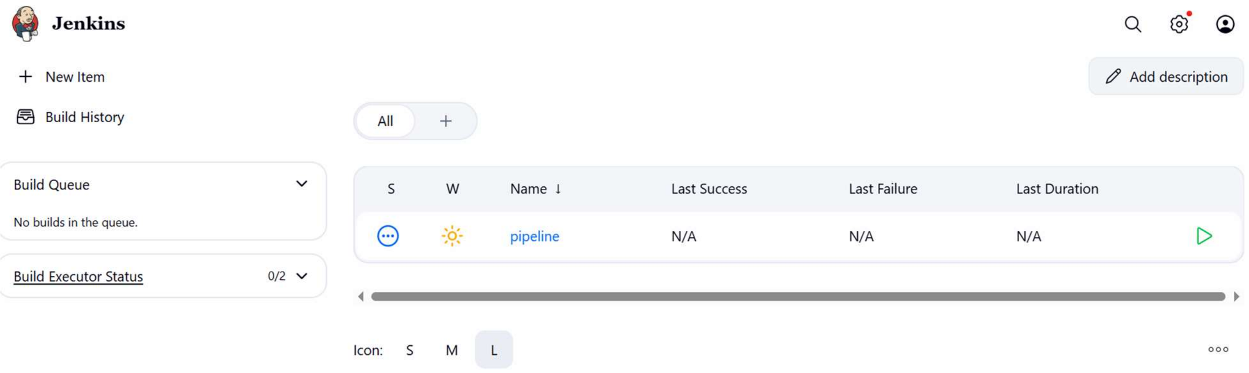
http://18.117.83.82:8080//github-webhook/

Content type *

application/json

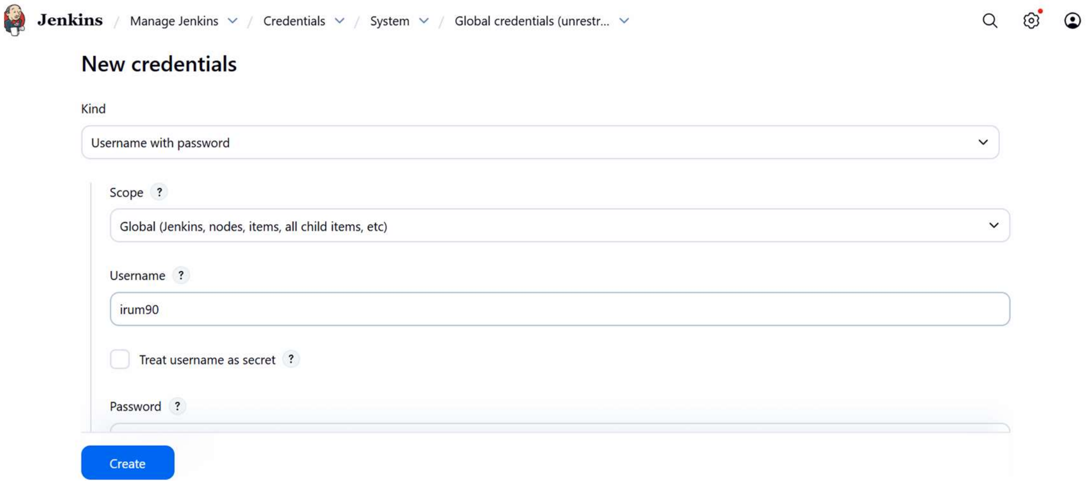
Secret

Created a pipeline



S	W	Name ↓	Last Success	Last Failure	Last Duration
⋯	☀	pipeline	N/A	N/A	N/A

Added dockerhub credentials to Jenkins


Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr...

New credentials

Kind

Username with password

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

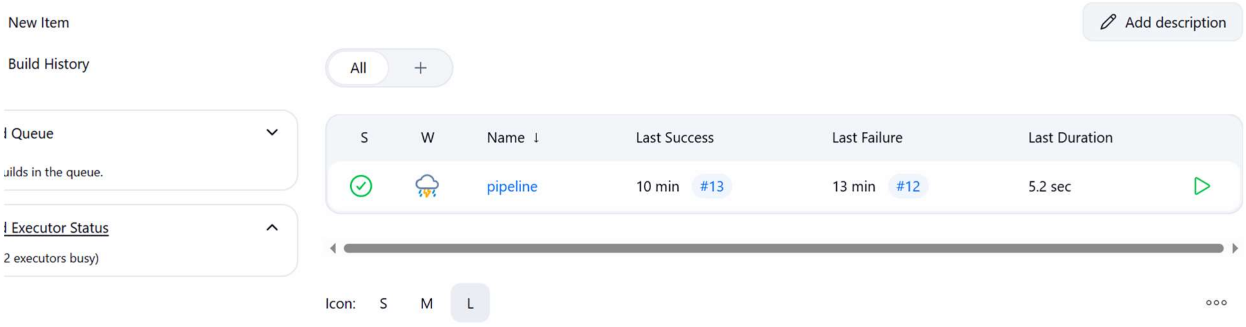
Username ?

irum90

□ Treat username as secret ?

Password ?

Executed successfully

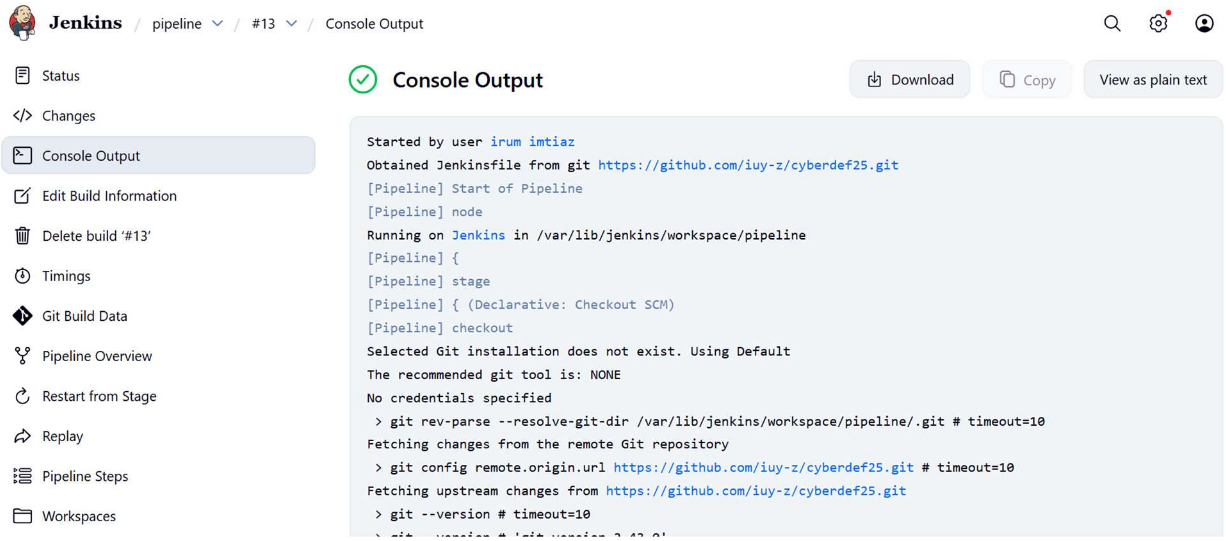


uilds in the queue.

2 executors busy)

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓		pipeline	10 min #13	13 min #12	5.2 sec

Console output



```
Started by user irum imtiaz
Obtained Jenkinsfile from git https://github.com/iuy-z/cyberdef25.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/pipeline/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/iuy-z/cyberdef25.git # timeout=10
Fetching upstream changes from https://github.com/iuy-z/cyberdef25.git
 > git --version # timeout=10
```

Started by user irum imtiaz

Obtained Jenkinsfile from git https://github.com/iuy-z/cyberdef25.git

[Pipeline] Start of Pipeline

[Pipeline] node

Running on Jenkins in /var/lib/jenkins/workspace/pipeline

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Declarative: Checkout SCM)

[Pipeline] checkout

Selected Git installation does not exist. Using Default

The recommended git tool is: NONE

No credentials specified

> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/pipeline/.git # timeout=10

Fetching changes from the remote Git repository

> git config remote.origin.url https://github.com/iuy-z/cyberdef25.git # timeout=10

Fetching upstream changes from https://github.com/iuy-z/cyberdef25.git

> git --version # timeout=10

> git --version # 'git version 2.43.0'

```
 > git fetch --tags --force --progress -- https://github.com/iuy-z/cyberdef25.git
+refs/heads/*:refs/remotes/origin/* # timeout=10

 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10

Checking out Revision 2d10f6ca82c526df3d55e2060b80bb9d6c304ecf (refs/remotes/origin/main)

 > git config core.sparsecheckout # timeout=10

 > git checkout -f 2d10f6ca82c526df3d55e2060b80bb9d6c304ecf # timeout=10

Commit message: "Fix Docker Compose command syntax in Jenkinsfile"

 > git rev-list --no-walk 2d10f6ca82c526df3d55e2060b80bb9d6c304ecf # timeout=10

[Pipeline] }

[Pipeline] // stage

[Pipeline] withEnv

[Pipeline] {

[Pipeline] withEnv

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Build Docker Image)

[Pipeline] echo

Building Docker image irum90/cyberdef25:latest...

[Pipeline] sh

+ docker build -t irum90/cyberdef25:latest .

DEPRECATED: The legacy builder is deprecated and will be removed in a future release.

            Install the buildx component to build images with BuildKit:

            https://docs.docker.com/go/buildx/


Sending build context to Docker daemon  184.3kB


Step 1/9 : FROM python:3.10-slim

 ---> 6f924957e3d2

Step 2/9 : WORKDIR /app
```

```
 ---> Using cache
 ---> 629741d90278
Step 3/9 : COPY requirements.txt .
 ---> Using cache
 ---> 0c58d6f8fa98
Step 4/9 : RUN pip install --no-cache-dir -r requirements.txt
 ---> Using cache
 ---> e783fa046220
Step 5/9 : COPY model.pkl .
 ---> Using cache
 ---> e1ed09660bb0
Step 6/9 : COPY inference.py .
 ---> Using cache
 ---> a38e45a4941a
Step 7/9 : RUN mkdir -p /input/logs
 ---> Using cache
 ---> 0ceef277f335
Step 8/9 : RUN mkdir -p /output
 ---> Using cache
 ---> c5d0cf71fbae
Step 9/9 : ENTRYPOINT ["python", "inference.py"]
 ---> Using cache
 ---> ec00001eac9d
Successfully built ec00001eac9d
Successfully tagged irum90/cyberdef25:latest
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Login to Docker Hub)
```

```
[Pipeline] echo
Logging in to Docker Hub...
[Pipeline] withCredentials
Masking supported pattern matches of $DOCKER_PASS
[Pipeline] {
[Pipeline] sh
+ docker login -u irum90 --password-stdin
+ echo ****
Login Succeeded
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Push Docker Image)
[Pipeline] echo
Pushing Docker image to Docker Hub...
[Pipeline] sh
+ docker push irum90/cyberdef25:latest
The push refers to repository [docker.io/irum90/cyberdef25]
3eba56701aad: Preparing
d4754875c557: Preparing
330db015b679: Preparing
55086991c32d: Preparing
398e3b6a8f84: Preparing
3712dee41860: Preparing
747c0b217383: Preparing
548ba1e7e829: Preparing
a8045a14b5f4: Preparing
```

```
cb4ecf39d967: Preparing
70a290c5e58b: Preparing
548ba1e7e829: Waiting
a8045a14b5f4: Waiting
cb4ecf39d967: Waiting
70a290c5e58b: Waiting
3712dee41860: Waiting
747c0b217383: Waiting
55086991c32d: Layer already exists
3eba56701aad: Layer already exists
398e3b6a8f84: Layer already exists
330db015b679: Layer already exists
d4754875c557: Layer already exists
3712dee41860: Layer already exists
747c0b217383: Layer already exists
548ba1e7e829: Layer already exists
a8045a14b5f4: Layer already exists
cb4ecf39d967: Layer already exists
70a290c5e58b: Layer already exists
latest: digest:
sha256:a0145f5d74262e3bd7b78e76728fbcedd1cf141f4a0e0c330e487c47c9363ab6 size: 2612
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Run Container with Docker Compose)
[Pipeline] echo
Running container using docker-compose...
[Pipeline] sh
+ docker-compose down
```

```
Removing network pipeline_default

+ docker-compose up --build -d

Creating network "pipeline_default" with the default driver

Creating cyberdef25_inference ...

Creating cyberdef25_inference ... done

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Declarative: Post Actions)

[Pipeline] echo

Jenkins pipeline finished successfully.

[Pipeline] }

[Pipeline] // stage

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] }

[Pipeline] // node

[Pipeline] End of Pipeline

Finished: SUCCESS
```