

Advanced Python Programming (APP)

Course Code (CC) – CSI – 3007

Laboratory (Lab) Class Live Day To Day Assessment Assignment

Docker

Name :- Dharshan Raj P A

Register Number :- 22MIC0073

Slot :- A2 + L7 + L8 + L27 + L28

Faculty:-Dr. Prof. Sharmila Banu K

Semester:- Fall Semester (2024-2025)

1)Docker:-

Projects-J_Component\ Docker\nginx\ Dockerfile

```
FROM nginx:alpine

# Remove default nginx config
RUN rm /etc/nginx/conf.d/default.conf

# Copy custom nginx configuration
COPY nginx/nginx.conf /etc/nginx/conf.d/app.conf

# Create directory for static files
RUN mkdir -p /var/www/static

# Expose port 80
EXPOSE 80

# Health check
HEALTHCHECK --interval=30s --timeout=3s --start-period=5s --retries=3 \
    CMD wget --quiet --tries=1 --spider http://localhost/health || exit 1

# Start nginx
CMD ["nginx", "-g", "daemon off;"]
```

Projects-J_Component\ Docker\nginx\nginx.conf

```
upstream flask_app {
    server flask:5000;
```

```
# Add more Flask instances for Load balancing
# server flask:5001;
}

# Rate Limiting zones
limit_req_zone $binary_remote_addr zone=api_limit:10m rate=10r/s;
limit_req_zone $binary_remote_addr zone=login_limit:10m rate=5r/m;

server {
    listen 80;
    server_name localhost;

    # Maximum upload size
    client_max_body_size 10M;

    # Security headers
    add_header X-Frame-Options "SAMEORIGIN" always;
    add_header X-Content-Type-Options "nosniff" always;
    add_header X-XSS-Protection "1; mode=block" always;

    # Gzip compression
    gzip on;
    gzip_vary on;
    gzip_min_length 1024;
    gzip_types text/plain text/css text/xml text/javascript application/x-javascript application/xml+rss application/json;

    # Static files (if serving directly from Nginx)
    location /static {
        alias /var/www/static;
        expires 30d;
        add_header Cache-Control "public, immutable";
    }

    # Health check endpoint (no rate limiting)
    location /health {
        proxy_pass http://flask_app;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        access_log off;
    }

    # Login endpoint with stricter rate limiting
    location ~ ^/(login|register) {
        limit_req zone=login_limit burst=3 nodelay;
        proxy_pass http://flask_app;
    }
}
```

```

proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme;
}

# API endpoints with rate limiting
location /api {
    limit_req zone=api_limit burst=20 nodelay;
    proxy_pass http://flask_app;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
}

# All other requests
location / {
    limit_req zone=api_limit burst=20 nodelay;
    proxy_pass http://flask_app;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;

    # WebSocket support (if needed in future)
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";

    # Timeouts
    proxy_connect_timeout 60s;
    proxy_send_timeout 60s;
    proxy_read_timeout 60s;
}

# Error pages
error_page 500 502 503 504 /50x.html;
location = /50x.html {
    root /usr/share/nginx/html;
}
}

```

Projects-J_Component\ Docker\ .dockerignore

Git

.git
.gitignore
.gitattributes

Environment files

.env
.env.local
.env.*.local

Python

__pycache__
*.py[cod]
*\$py.class
*.so

.Python

env/
venv/
ENV/
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/

```
var/
wheels/
*.egg-info/
.installed.cfg
*.egg

# IDE
.vscode/
.idea/
*.swp
*.swo
*~

.DS_Store

# Logs
*.log
logs/
*.log.*

# Database
*.db
*.sqlite
*.sqlite3

# Docker
docker-compose.override.yml
.dockerignore
```

Documentation

README.md

***.md**

!Docker/README.md

Test files

tests/

test_*.py

***_test.py**

.pytest_cache/

.coverage

htmlcov/

Temporary files

tmp/

temp/

***.tmp**

OS files

Thumbs.db

.DS_Store

Jupyter notebooks (if any)

***.ipynb**

.ipynb_checkpoints/

Flutter/Weather app (not needed for Flask)

weather_app/

Flutter/

Projects-J_Component\ Docker\ docker-compose.prod.yml

```
version: '3.8'

# Production overrides for docker-compose.yml
# Usage: docker-compose -f docker-compose.yml -f docker-compose.prod.yml up -d

services:
  flask:
    environment:
      FLASK_ENV: production
      FLASK_DEBUG: "false"
    # Override command for production with more workers
    command: ["gunicorn", "--bind", "0.0.0.0:5000", "--workers", "8", "--worker-class", "sync", "--timeout", "120", "--keep-alive", "5", "--max-requests", "1000", "--max-requests-jitter", "50", "--access-logfile", "-", "--error-logfile", "-", "app:app"]
    # Resource limits
    deploy:
      resources:
        limits:
          cpus: '2'
          memory: 2G
        reservations:
          cpus: '1'
          memory: 1G
    # Restart policy
    restart: always

  postgres:
    # Resource limits
    deploy:
      resources:
        limits:
          cpus: '1'
          memory: 1G
        reservations:
          cpus: '0.5'
          memory: 512M
    # Restart policy
    restart: always
    # Production PostgreSQL settings
    command: >
```

```

postgres
  -c shared_buffers=256MB
  -c max_connections=100
  -c effective_cache_size=1GB
  -c maintenance_work_mem=64MB
  -c checkpoint_completion_target=0.9
  -c wal_buffers=16MB
  -c default_statistics_target=100
  -c random_page_cost=1.1
  -c effective_io_concurrency=200
  -c work_mem=4MB
  -c min_wal_size=1GB
  -c max_wal_size=4GB

redis:
  # Resource limits
  deploy:
    resources:
      limits:
        cpus: '0.5'
        memory: 512M
      reservations:
        cpus: '0.25'
        memory: 256M
    # Restart policy
    restart: always
    # Production Redis settings
    command: >
      redis-server
      --appendonly yes
      --requirepass ${REDIS_PASSWORD}
      --maxmemory 256mb
      --maxmemory-policy allkeys-lru

nginx:
  # Resource limits
  deploy:
    resources:
      limits:
        cpus: '0.5'
        memory: 256M
      reservations:
        cpus: '0.25'
        memory: 128M
    # Restart policy
    restart: always
    # Uncomment for SSL/HTTPS
    # ports:

```

```

#      - "80:80"
#      - "443:443"
# volumes:
#      - ./Flask/static:/var/www/static:ro
#      - ./nginx/ssl:/etc/nginx/ssl:ro

```

Projects-J_Component\ Docker\ docker-compose.yml

```

version: '3.8'

services:
  # PostgreSQL Database
  postgres:
    image: postgres:15-alpine
    container_name: foodapp_postgres
    environment:
      POSTGRES_USER: ${DB_USER:-foodapp_user}
      POSTGRES_PASSWORD: ${DB_PASSWORD:-foodapp_pass}
      POSTGRES_DB: ${DB_NAME:-food_order_db}
      PGDATA: /var/lib/postgresql/data/pgdata
    volumes:
      - postgres_data:/var/lib/postgresql/data
    ports:
      - "${DB_PORT:-5432}:5432"
    networks:
      - foodapp_network
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U ${DB_USER:-foodapp_user} -d ${DB_NAME:-food_order_db}"]
      interval: 10s
      timeout: 5s
      retries: 5
      start_period: 10s
    restart: unless-stopped

  # Redis Cache
  redis:
    image: redis:7-alpine
    container_name: foodapp_redis
    command: redis-server --appendonly yes --requirepass ${REDIS_PASSWORD:-redis_pass}
    volumes:
      - redis_data:/data
    ports:
      - "${REDIS_PORT:-6379}:6379"
    networks:

```

```

- foodapp_network
healthcheck:
  test: ["CMD", "redis-cli", "-a", "${REDIS_PASSWORD:-redis_pass}", "ping"]
    interval: 10s
    timeout: 3s
    retries: 5
    start_period: 10s
  restart: unless-stopped

# Flask Application
flask:
  build:
    context: ..
    dockerfile: Docker/Dockerfile
  container_name: foodapp_flask
  environment:
    # Database configuration
    DB_HOST: postgres
    DB_PORT: 5432
    DB_NAME: ${DB_NAME:-food_order_db}
    DB_USER: ${DB_USER:-foodapp_user}
    DB_PASSWORD: ${DB_PASSWORD:-foodapp_pass}
    # Redis configuration
    REDIS_HOST: redis
    REDIS_PORT: 6379
    REDIS_DB: ${REDIS_DB:-0}
    REDIS_PASSWORD: ${REDIS_PASSWORD:-redis_pass}
    # Flask configuration
    SECRET_KEY: ${SECRET_KEY:-dev-secret-key-change-in-production}
    FLASK_ENV: ${FLASK_ENV:-production}
    FLASK_DEBUG: ${FLASK_DEBUG:-false}
    FLASK_HOST: 0.0.0.0
    FLASK_PORT: 5000
    INIT_DB: "true"
  volumes:
    - flask_logs:/app/logs
    - ./Flask/static:/app/static:ro
  depends_on:
    postgres:
      condition: service_healthy
    redis:
      condition: service_healthy
  networks:
    - foodapp_network
  healthcheck:
    test: ["CMD", "python", "-c", "import urllib.request; urllib.request.urlopen('http://localhost:5000/health').read()"]

```

```

    interval: 30s
    timeout: 10s
    retries: 3
    start_period: 40s
    restart: unless-stopped

# Nginx Reverse Proxy
nginx:
  build:
    context: .
    dockerfile: nginx/Dockerfile
  container_name: foodapp_nginx
  ports:
    - "${NGINX_PORT:-80}:80"
    # Uncomment for HTTPS
    # - "${NGINX_HTTPS_PORT:-443}:443"
  volumes:
    - ./Flask/static:/var/www/static:ro
    # Uncomment for SSL certificates
    # - ./nginx/ssl:/etc/nginx/ssl:ro
  depends_on:
    flask:
      condition: service_healthy
  networks:
    - foodapp_network
  healthcheck:
    test: ["CMD", "wget", "--quiet", "--tries=1", "--spider",
"http://localhost/health"]
    interval: 30s
    timeout: 3s
    retries: 3
    start_period: 10s
    restart: unless-stopped

# Named volumes for data persistence
volumes:
  postgres_data:
    driver: local
  redis_data:
    driver: local
  flask_logs:
    driver: local

# Network for service communication
networks:
  foodapp_network:
    driver: bridge

```

Projects-J_Component\ Docker\ Dockerfile

```
# Multi-stage build for Flask application
FROM python:3.11-slim as builder

# Set working directory
WORKDIR /app

# Install system dependencies
RUN apt-get update && apt-get install -y --no-install-recommends \
    gcc \
    postgresql-client \
    && rm -rf /var/lib/apt/lists/*

# Copy requirements and install Python dependencies
COPY Flask/requirements.txt .
RUN pip install --no-cache-dir --user -r requirements.txt

# Final stage
FROM python:3.11-slim

# Set working directory
WORKDIR /app

# Install runtime dependencies
RUN apt-get update && apt-get install -y --no-install-recommends \
    postgresql-client \
    && rm -rf /var/lib/apt/lists/*

# Create non-root user for security
RUN useradd -m -u 1000 appuser && chown -R appuser:appuser /app

# Copy Python dependencies from builder
COPY --from=builder /root/.local /home/appuser/.local

# Copy application code
COPY Flask/ /app/

# Set environment variables
ENV PATH=/home/appuser/.local/bin:$PATH
ENV PYTHONUNBUFFERED=1
ENV FLASK_APP=app.py
ENV INIT_DB=true

# Change ownership
RUN chown -R appuser:appuser /app
```

```

# Switch to non-root user
USER appuser

# Expose port
EXPOSE 5000

# Health check
HEALTHCHECK --interval=30s --timeout=10s --start-period=40s --retries=3 \
  CMD python -c "import urllib.request;
urllib.request.urlopen('http://localhost:5000/health').read()" || exit 1

# Run with gunicorn for production
CMD ["gunicorn", "--bind", "0.0.0.0:5000", "--workers", "4", "--timeout",
"120", "--access-logfile", "-", "--error-logfile", "-", "app:app"]

```

Projects-J_Component\ Docker\ start.bat

```

@echo off
REM Quick start script for Docker setup on Windows
REM Usage: start.bat [dev/prod]

set MODE=%1
if "%MODE%"=="" set MODE=dev

echo Starting Flask Food Delivery App in %MODE% mode...

REM Check if .env file exists
if not exist .env (
    echo Creating .env file from env.example...
    copy env.example .env
    echo.
    echo WARNING: Please edit .env file and update SECRET_KEY, DB_PASSWORD,
and REDIS_PASSWORD before continuing!
    echo Press any key to continue or Ctrl+C to exit...
    pause >nul
)

REM Start services
if "%MODE%"=="prod" (
    echo Starting in PRODUCTION mode...
    docker-compose -f docker-compose.yml -f docker-compose.prod.yml up -d --
build
) else (
    echo Starting in DEVELOPMENT mode...
    docker-compose up -d --build
)

```

```

REM Wait for services to be healthy
echo.
echo Waiting for services to be healthy...
timeout /t 10 /nobreak >nul

REM Check service status
echo.
echo Service Status:
docker-compose ps

echo.
echo Application should be available at: http://localhost
echo Health check: http://localhost/health
echo.
echo View logs with: docker-compose logs -f
echo Stop services with: docker-compose down

```

Projects-J_Component\ Docker\start.sh

```

#!/bin/bash

# Quick start script for Docker setup
# Usage: ./start.sh [dev/prod]

MODE=${1:-dev}

echo "Starting Flask Food Delivery App in $MODE mode..."

# Check if .env file exists
if [ ! -f .env ]; then
    echo "Creating .env file from env.example..."
    cp env.example .env
    echo "⚠ Please edit .env file and update SECRET_KEY, DB_PASSWORD, and
REDIS_PASSWORD before continuing!"
    echo "Press Enter to continue or Ctrl+C to exit..."
    read
fi

# Start services
if [ "$MODE" = "prod" ]; then
    echo "Starting in PRODUCTION mode..."
    docker-compose -f docker-compose.yml -f docker-compose.prod.yml up -d --
build
else
    echo "Starting in DEVELOPMENT mode..."

```

```

    docker-compose up -d --build
fi

# Wait for services to be healthy
echo "Waiting for services to be healthy..."
sleep 10

# Check service status
echo ""
echo "Service Status:"
docker-compose ps

echo ""
echo "☒ Application should be available at: http://localhost"
echo "☒ Health check: http://localhost/health"
echo ""
echo "View logs with: docker-compose logs -f"
echo "Stop services with: docker-compose down"

```

Output Screenshots Snip Snap Terminal Bash Command Prompt Shell:-

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\Dharshan Raj P A\College\Laboratory\Advanced_Python_Programming_Laboratory\Projects-J_Component\Docker> .\start.bat
Starting Flask Food Delivery App in dev mode...
time=2025-11-07T00:46:42+05:30" level=warning msg="C:\\\\Dharshan Raj P A\\\\College\\\\Laboratory\\\\Advanced_Python_Programming_Laboratory\\\\Projects-J_Component\\\\Docker\\\\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 0/2
- redis Pulled
  ✓ d75b8bec998 Pull complete
  ✓ 380e8aa8b1fd Pull complete
  ✓ 232f7549c9b0 Pull complete
  ✓ 4f4fb700ef50 Pull complete
  ✓ 60c57c0072ef Pull complete
  ✓ c78aae7b5e0f Pull complete
  ✓ fc4343b4accd Pull complete
  ✓ f637881d1138 Pull complete
  ✓ postgres Pulled
    ✓ 9239d8c236e2 Pull complete
    ✓ d7cf304fb91a Pull complete
    ✓ e7dbb5a29e19 Pull complete
    ✓ 0a297d5cb757 Pull complete
    ✓ 01a8e1e8a6d3 Pull complete
    ✓ 78753f2bcd40 Pull complete
    ✓ 8e7af31e0abd Pull complete
    ✓ 665850181beb Pull complete
    ✓ b7a5bbdd8445b Pull complete
    ✓ 2d35ebdb57d9 Pull complete
    ✓ c8d6a201b2e Pull complete
[+] Building 172.7s (27/27) FINISHED
=> [internal] load local bake definitions
=> reading from stdin 1.25kB
=> [flask internal] load build definition from Dockerfile
=> transferring dockerfile: 1.62kB
=> WARN: FromAsCasing: 'as' and 'FROM' keywords' casing do not match (line 2)
=> [nginx internal] load build definition from Dockerfile
=> transferring dockerfile: 538B
=> [nginx internal] load metadata for docker.io/library/nginx:alpine
=> [flask internal] load metadata for docker.io/library/python:3.11-slim
=> [nginx internal] load .dockerrignore
=> transferring context: 895B
=> [nginx 1/4] FROM docker.io/library/nginx:alpine@sha256:b3c656d55d7ad751196f21b7fd2e8d4da9cb430e32f646adcfc924
=> => resolve docker.io/library/nginx:alpine@sha256:b3c656d55d7ad751196f21b7fd2e8d4da9cb430e32f646adcfc92441b72f8 0.0s
=> => sha256:bdabb0d442710d667f4fd871b5fd215cc2a430a95b192bc598bf945b8e60999b 16.97MB / 16.97MB 12.9s
=> => sha256:ff8a36d5502a57c3fc8eff48e578ab433a03bdd528992ba0d966ddf853309a 1.48kB / 1.48kB 1.7s

```

```

Windows PowerShell x + ~

>> [flask builder 3/5] RUN apt-get update && apt-get install -y --no-install-recommends gcc postgresql- 77.7s
>> [flask stage-1 4/7] RUN useradd -m -u 1000 appuser && chown -R appuser:appuser /app 0.8s
>> [flask builder 4/5] COPY Flask/requirements.txt . 2.6s
>> [flask builder 5/5] RUN pip install --no-cache-dir --user -r requirements.txt 46.8s
>> [flask stage-1 5/7] COPY --from=builder /root/.local /home/appuser/.local 0.4s
>> [flask stage-1 6/7] COPY Flask/ /app/ 0.1s
>> [flask stage-1 7/7] RUN chown -R appuser:appuser /app 0.5s
>> [flask] exporting to image 6.3s
>> exporting layers 4.6s
>> exporting manifest sha256:97bb6f037085cd24025513defb93dfc6d6a7406feelce7378beb0333ab0f1ce 0.0s
>> exporting config sha256:9816bd619b9bc941799a68e87bfa511658de33435615b997a93d98bae326b6d9 0.0s
>> exporting attestation manifest sha256:87b091d423b5c43b6e46937dd2796f2c89e93eaa40cf3798097952feebdb186 0.0s
>> exporting manifest list sha256:75fcfd3025504f6ddd7aeaf3f2f27395435b10d56c9d7654a2d7acd719fdc61c93 0.0s
>> naming to docker.io/library/docker-flask:latest 0.0s
>> unpacking to docker.io/library/docker-flask:latest 0.0s
>> [flask] resolving provenance for metadata file 1.5s
[+] Running 10/18
  ✓ docker-flask          Built          0.0s
  ✓ docker-nginx           Built          0.0s
  ✓ Network docker-foodapp_network Created      0.1s
  ✓ Volume "docker_redis_data" Created      0.0s
  ✓ Volume "docker_flask_logs" Created      0.0s
  ✓ Volume "docker_postgres_data" Created      0.0s
  ✓ Container foodapp_redis   Healthy       0.0s
  ✓ Container foodapp_postgres Healthy       0.0s
  ✓ Container foodapp_flask    Healthy       13.0s
  ✓ Container foodapp_nginx   Started       13.2s

Waiting for services to be healthy...

Service Status:
time="2025-11-07T00:57:20+05:30" level=warning msg="C:\Dharshan Raj P A\College\Laboratory\Advanced_Python_Programming_Laboratory\Projects-J_Component\\Docker\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
NAME           IMAGE            COMMAND          SERVICE     CREATED        STATUS          PORTS
foodapp_flask   docker-flask   "gunicorn --bind 0.0..." flask        23 seconds ago Up 16 seconds (healthy)  5000/tcp
foodapp_nginx   docker-nginx   "/dockerc-entrypoint.s..." nginx      23 seconds ago Up 10 seconds (health: starting)  0.0.0.0:80->80/tcp, [::]:80-
>80/tcp
foodapp_postgres postgres:15-alpine "docker-entrypoint.s..." postgres    24 seconds ago Up 23 seconds (healthy)  0.0.0.0:5432->5432/tcp, [:]
:5432->5432/tcp
foodapp_redis   redis:7-alpine "dockerc-entrypoint.s..." redis      24 seconds ago Up 23 seconds (healthy)  0.0.0.0:6379->6379/tcp, [:]
:6379->6379/tcp

Application should be available at: http://localhost
Health check: http://localhost/health

View logs with: docker-compose logs -f
Stop services with: docker-compose down
PS C:\Dharshan Raj P A\College\Laboratory\Advanced_Python_Programming_Laboratory\Projects-J_Component\Docker>

```

The screenshot shows the Docker Desktop application interface. On the left, there's a sidebar with various sections like Ask Gordon, Containers, Images, Volumes, Kubernetes, Models, MCP Toolkit, Docker Hub, Docker Scout, and Extensions. The 'Containers' section is currently selected.

In the main area, there are two charts: 'Container CPU usage' (36.84% / 1200% (12 CPUs available)) and 'Container memory usage' (521.27MB / 7.42GB). Below these charts is a search bar and a filter option 'Only show running containers'. A table lists the running containers:

	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	real-estate-asset-tokenizer	-	-	-	0.67%	19 minutes ago	⋮ trash
<input type="checkbox"/>	realestate-mongo-express	97195b176ec6	mongo-express:1.0.0	8081:8081	0%	19 minutes ago	⋮ trash
<input type="checkbox"/>	realestate-mongodb	69160b221999	mongo:7.0	27017:27017	0.67%	19 minutes ago	⋮ trash
<input type="checkbox"/>	docker	-	-	-	0.88%	9 minutes ago	⋮ trash
<input type="checkbox"/>	foodapp_redis	a217654daa44	redis:7-alpine	6379:6379	0.83%	9 minutes ago	⋮ trash
<input type="checkbox"/>	foodapp_postgres	83ccc1e2fad0	postgres:15-alpine	5432:5432	0.01%	9 minutes ago	⋮ trash
<input type="checkbox"/>	foodapp_flask	39766962bc81	docker-flask	-	0.04%	9 minutes ago	⋮ trash
<input type="checkbox"/>	foodapp_nginx	6c4f207b9c27	docker-nginx	80:80	0%	9 minutes ago	⋮ trash

Below the table, it says 'Showing 8 items'. Under the 'Walkthroughs' section, there are two cards: 'Multi-container applications' (8 mins) and 'Containerize your application' (\$ docker init 3 mins). At the bottom, there are status indicators for Engine running, RAM usage, CPU usage, Disk usage, and a link to the Learning center.

docker desktop PERSONAL

Ask Gordon BETA

Containers

Images

Volumes

Kubernetes

Builds

Models

MCP Toolkit BETA

Docker Hub

Docker Scout

Extensions

Images Give feedback

Local My Hub

1.74 GB / 1.46 GB in use 6 Images

Last refresh: 14 hours ago

Search

Name	Tag	Image ID	Created	Size	Actions	
mongo	7.0	c258b26dbb77	1 month ago	1.12 GB		
mongo-express	1.0.0	52f18378afac	2 years ago	340.02 MB		
redis	7-alpine	ee64a64eaab6	3 days ago	60.66 MB		
postgres	15-alpine	64583b3cb4f2	22 days ago	390.48 MB		
docker-nginx	latest	84d6e7089115	11 minutes ago	79.87 MB		
docker-flask	latest	75fc4d302554	9 minutes ago	327.62 MB		

Showing 6 items

Engine running RAM 4.98 GB CPU 0.17% Disk: 6.61 GB used (limit 1006.85 GB)

Terminal New version available

docker desktop PERSONAL

Ask Gordon BETA

Containers

Images

Volumes

Kubernetes

Builds

Models

MCP Toolkit BETA

Docker Hub

Docker Scout

Extensions

Volumes Give feedback

Create

Search

Name ↑	Created	Size	Actions	
091bebe9039d1ada2790dad8905c07f1cae857584ec8d0cf6e29af48e4e73a8d	4 days ago	0 Bytes		
docker_flask_logs	9 minutes ago	0 Bytes		
docker_postgres_data	9 minutes ago	21.1 MB		
docker_redis_data	9 minutes ago	176 Bytes		
e2cffcb037f16aac5ace6052eea4ec99c74fa2259e16239d002892a68d5cc04f	14 days ago	0 Bytes		
real-estate-asset-tokenizer-using-ethereum-blockchain-main_mongodb_data	14 days ago	306 MB		

Showing 6 items

Engine running RAM 4.97 GB CPU 1.58% Disk: 6.61 GB used (limit 1006.85 GB)

Terminal New version available

Docker Desktop PERSONAL

Search Ctrl+K

Selected builder desktop-linux Import builds Builder settings

Builds Give feedback

Build large and multi-platform Docker images faster in Docker Build Cloud. To improve build speeds for you, your team, and even your CI, try Docker Build Cloud now.

Build history Active builds

Search Show only my builds

	Name	ID	Builder	Duration	Created	Author
Projects-J_Component	oy7zl7	default	2m 50s	12 minutes ago	N/A	
Projects-J_Component/Docker	rmpj6y	default	22.6s	12 minutes ago	N/A	

Rows per page: 10 1–2 of 2

Engine running RAM 4.99 GB CPU 4.50% Disk: 6.61 GB used (limit 1006.85 GB)

Terminal New version available

A screenshot of the Docker Desktop application interface. The left sidebar contains links for Ask Gordon (Beta), Containers, Images, Volumes, Kubernetes, Builds (selected), Models, MCP Toolkit (Beta), Docker Hub, Docker Scout, and Extensions. The main area is titled 'Builds' with a 'Give feedback' button. A purple banner at the top right encourages using Docker Build Cloud for faster builds. Below this, there are two tabs: 'Build history' (selected) and 'Active builds'. A search bar and a 'Show only my builds' filter are present. A table lists two builds: 'Projects-J_Component' with ID 'oy7zl7' and 'Builder default', and 'Projects-J_Component/Docker' with ID 'rmpj6y' and 'Builder default'. Both builds completed 12 minutes ago. At the bottom, system status shows 'Engine running' with resource usage: RAM 4.99 GB, CPU 4.50%, and Disk usage of 6.61 GB (limit 1006.85 GB). A terminal icon and a 'New version available' notification are also visible.