Lab: Deploying New Relic in a Containerized Python + Redis Application

# Objective

- Deploy a Python Flask app with Redis in containers  
- Integrate New Relic APM agent with Python  
- Add Redis instrumentation  
- View application and Redis performance in New Relic dashboard

# Step 1: Prerequisites

- Docker & Docker Compose installed  
- A New Relic account (Free tier works)  
- Your New Relic License Key (from New Relic → Account Settings → API Keys)

# Step 2: Setup Project Structure

Run the following commands:

mkdir newrelic-lab  
cd newrelic-lab  
touch docker-compose.yml app.py requirements.txt Dockerfile

# Step 3: Python Flask App (with Redis)

app.py

from flask import Flask, request  
import redis  
  
app = Flask(\_\_name\_\_)  
  
# Connect to Redis  
r = redis.Redis(host="redis", port=6379)  
  
@app.route('/')  
def home():  
 count = r.incr('hits')  
 return f"Hello from Flask + Redis! This page has been visited {count} times."  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 app.run(host="0.0.0.0", port=5000)

requirements.txt

flask  
redis  
newrelic

# Step 4: Dockerfile for Python App

FROM python:3.9  
  
WORKDIR /app  
  
COPY requirements.txt .  
RUN pip install --no-cache-dir -r requirements.txt  
  
COPY . .  
  
EXPOSE 5000  
  
CMD ["newrelic-admin", "run-program", "python", "app.py"]

# Step 5: Docker Compose Setup

docker-compose.yml

version: "3.8"  
  
services:  
 app:  
 build: .  
 ports:  
 - "5000:5000"  
 environment:  
 - NEW\_RELIC\_LICENSE\_KEY=<YOUR\_NEW\_RELIC\_LICENSE\_KEY>  
 - NEW\_RELIC\_APP\_NAME=Flask-Redis-App  
 - NEW\_RELIC\_LOG=stdout  
 depends\_on:  
 - redis  
  
 redis:  
 image: redis:6.2  
 container\_name: redis  
 ports:  
 - "6379:6379"

# Step 6: Add New Relic Config File

Generate a config file:  
newrelic-admin generate-config $NEW\_RELIC\_LICENSE\_KEY newrelic.ini  
  
Copy newrelic.ini into the container (or mount it). Update:  
- app\_name = Flask-Redis-App  
- license\_key = <YOUR\_NEW\_RELIC\_LICENSE\_KEY>  
  
Modify Dockerfile to include:  
COPY newrelic.ini /app/

# Step 7: Run the Lab

Run:

docker-compose up --build

Open in browser: http://localhost:5000

# Step 8: Verify in New Relic

- Login to New Relic → APM → Applications  
- You should see Flask-Redis-App  
- Check Transactions for Flask routes  
- Check Datastores → Redis for Redis performance  
- Create dashboards with custom charts

# Step 9: Lab Extensions

1. Add Custom Transaction Traces in Flask:  
  
import newrelic.agent  
  
@app.route('/slow')  
@newrelic.agent.background\_task()  
def slow():  
 import time; time.sleep(3)  
 return "This is a slow endpoint"  
  
2. Add Error Monitoring by forcing exceptions  
  
3. Deploy with Kubernetes manifests instead of Docker Compose