🔷 Real-Time Task: Update Inventory or Analytics in Background

Scenario:

Whenever an order is placed, your system **updates inventory counts and analytics dashboards** asynchronously.

This prevents the main checkout page from waiting for database-heavy operations.

Create project folder and navigate into it

mkdir myproject
cd myproject

Create virtual environment

python -m venv venv

Temporarily allow script execution (PowerShell only)

Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope Process

Activate virtual environment

venv\Scripts\activate

Install Django and Celery

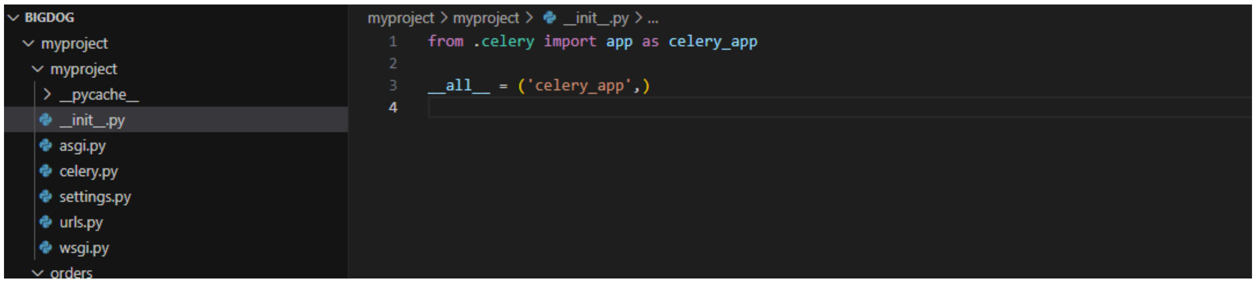
pip install django celery

Start Django project (use a different name if folder name conflicts)

python -m django startproject projectname .

Start a Django app

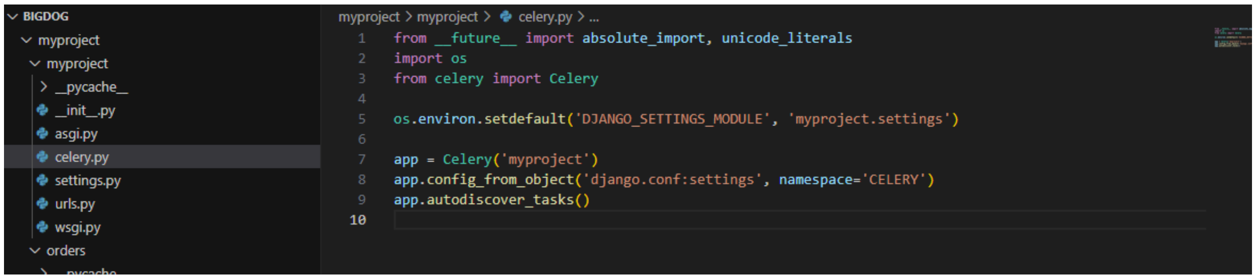
python manage.py startapp orders



BIGDOG
- myproject
  - myproject
    - __pycache__
    - __init__.py
    - asgi.py
    - celery.py
    - settings.py
    - urls.py
    - wsgi.py
  - orders

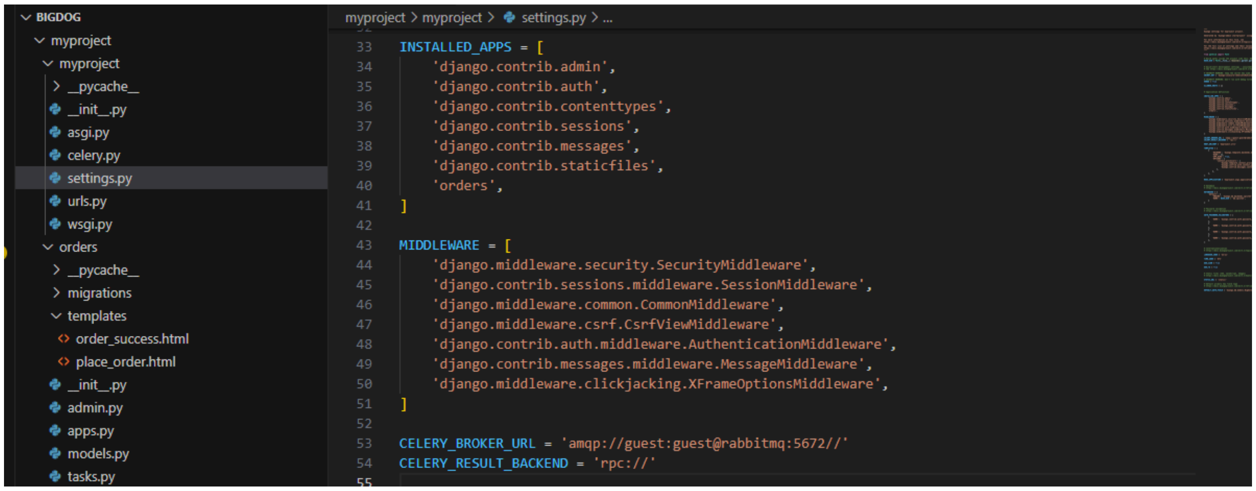
myproject > myproject > __init__.py > ...

```python
from .celery import app as celery_app

__all__ = ('celery_app',)

```



myproject > myproject > celery.py > ...

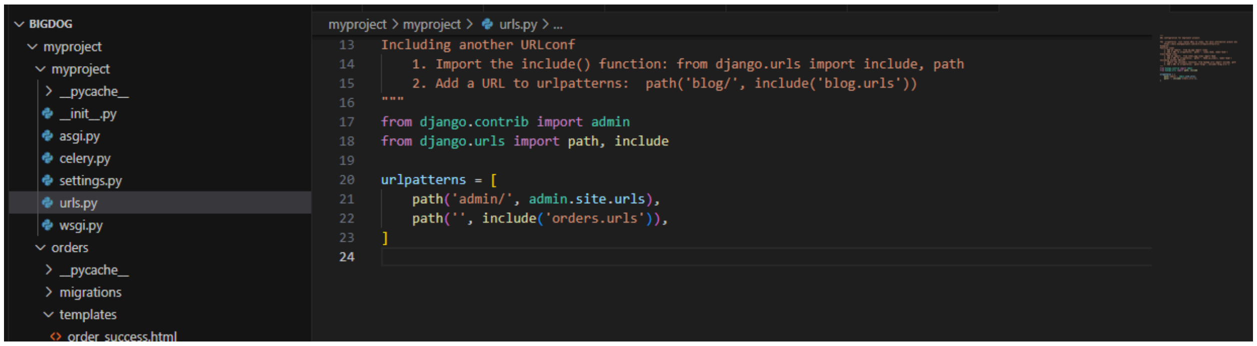
```python
from __future__ import absolute_import, unicode_literals
import os
from celery import Celery

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'myproject.settings')

app = Celery('myproject')
app.config_from_object('django.conf:settings', namespace='CELERY')
app.autodiscover_tasks()

```



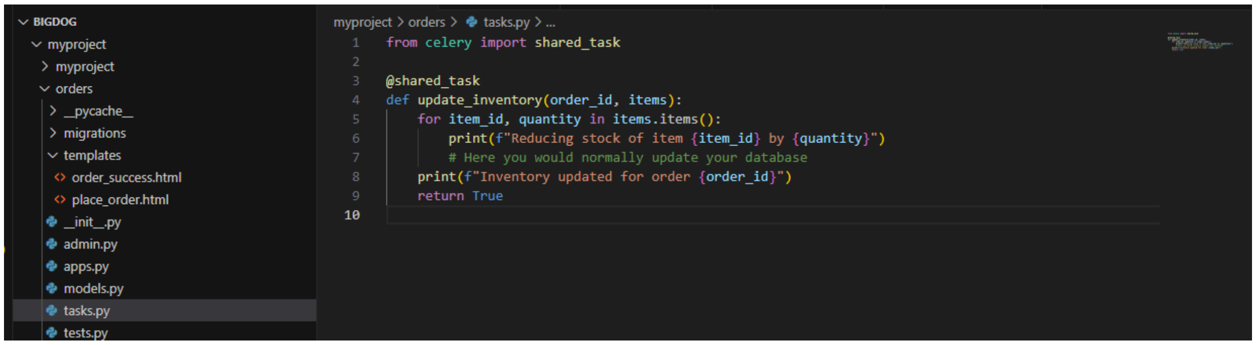
BIGDOG
- myproject
  - myproject
    - __pycache__
    - __init__.py
    - asgi.py
    - celery.py
    - settings.py
    - urls.py
    - wsgi.py
  - orders
    - __pycache__
    - migrations
    - templates
      - order_success.html
      - place_order.html
    - __init__.py
    - admin.py
    - apps.py
    - models.py
    - tasks.py

myproject > myproject > settings.py > ...

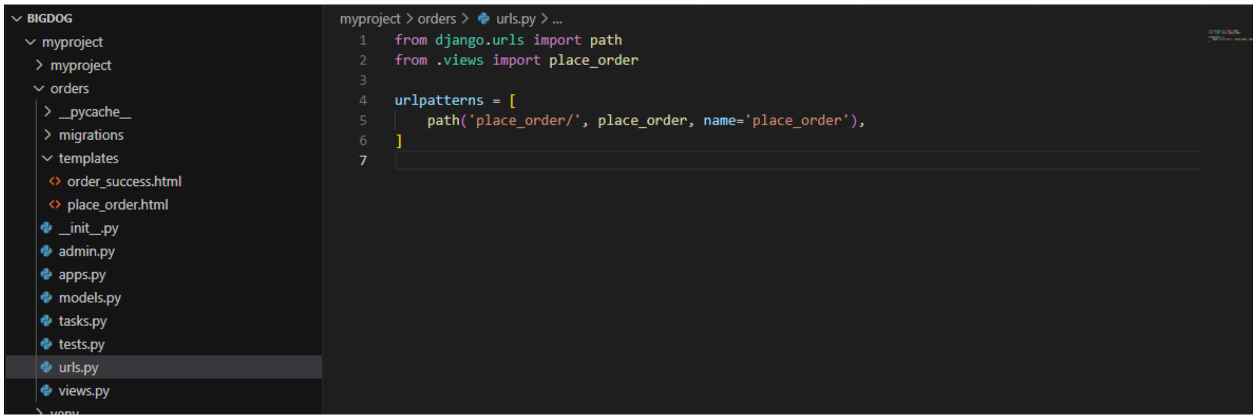
```python
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'orders',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

CELERY_BROKER_URL = 'amqp://guest:guest@rabbitmq:5672//'
CELERY_RESULT_BACKEND = 'rpc://'
```



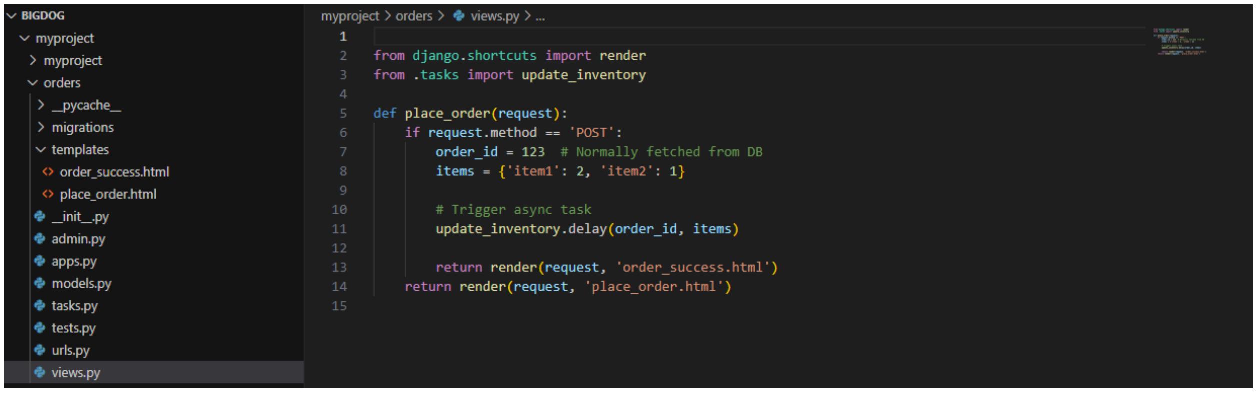
```python
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('orders.urls')),
]

```



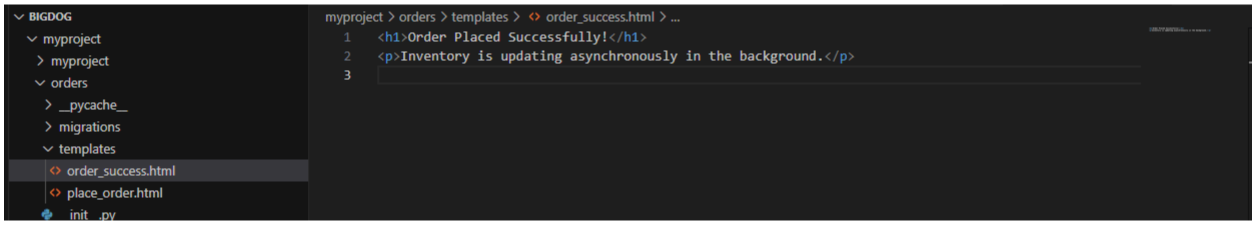
```python
from celery import shared_task

@shared_task
def update_inventory(order_id, items):
    for item_id, quantity in items.items():
        print(f"Reducing stock of item {item_id} by {quantity}")
        # Here you would normally update your database
    print(f"Inventory updated for order {order_id}")
    return True

```



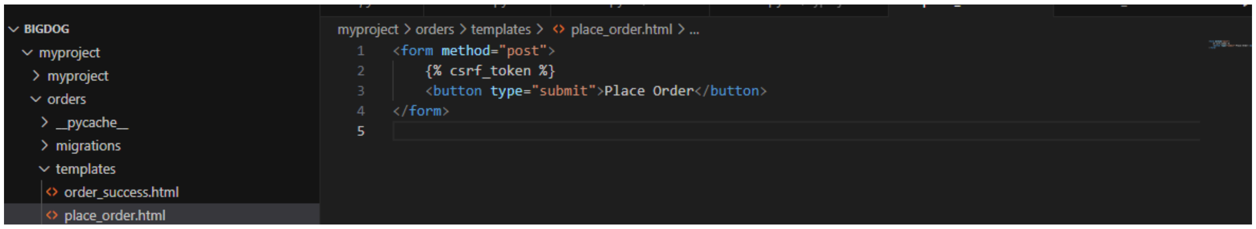
```python
from django.urls import path
from .views import place_order

urlpatterns = [
    path('place_order/', place_order, name='place_order'),
]

```



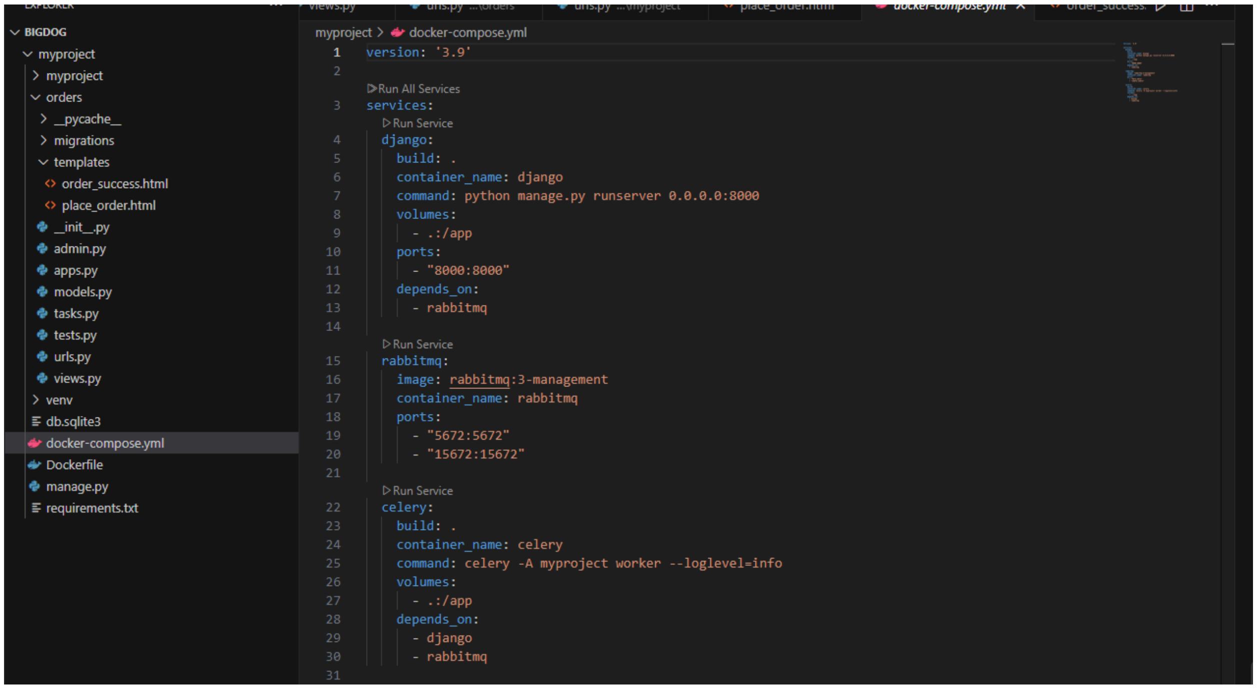
```python

from django.shortcuts import render
from .tasks import update_inventory

def place_order(request):
    if request.method == 'POST':
        order_id = 123  # Normally fetched from DB
        items = {'item1': 2, 'item2': 1}

        # Trigger async task
        update_inventory.delay(order_id, items)

        return render(request, 'order_success.html')
    return render(request, 'place_order.html')

```



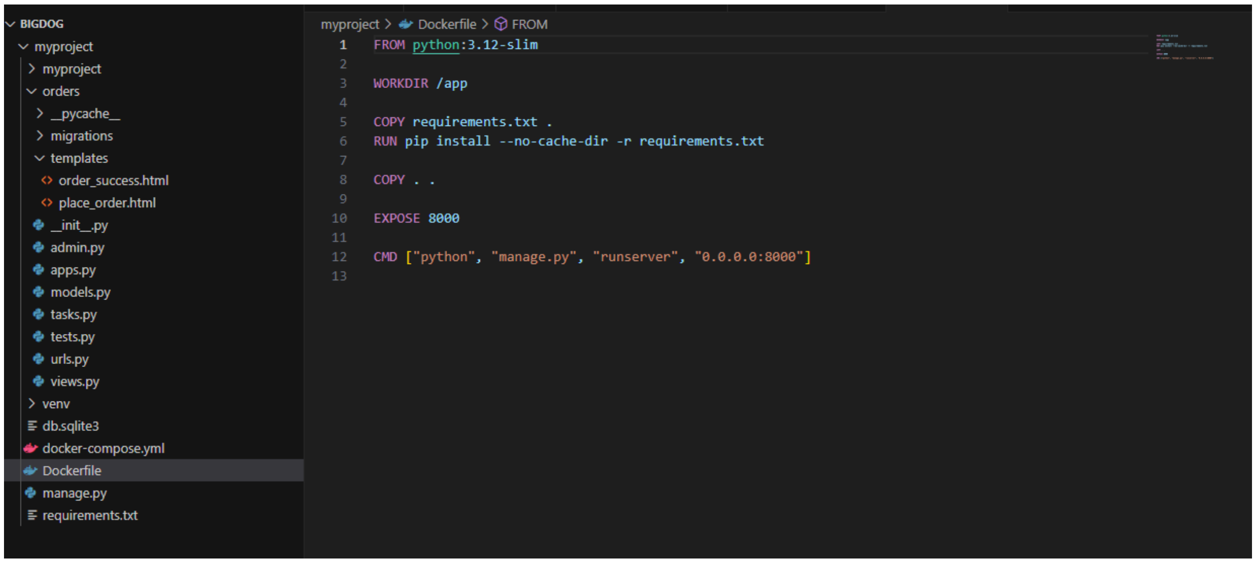
```html
<h1>Order Placed Successfully!</h1>
<p>Inventory is updating asynchronously in the background.</p>

```



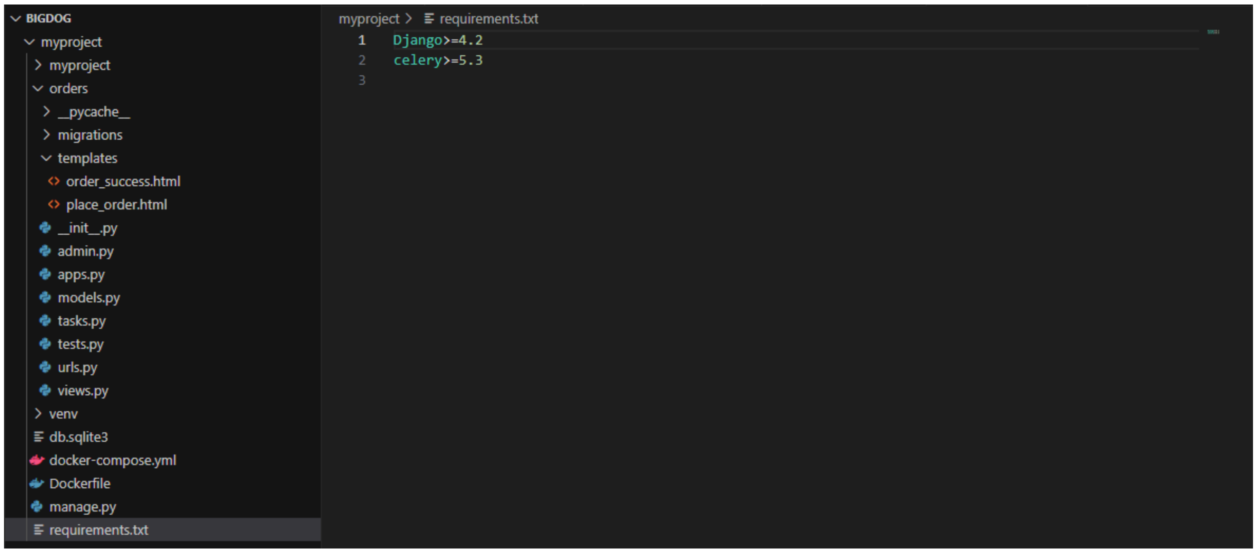
```html
<form method="post">
    {% csrf_token %}
    <button type="submit">Place Order</button>
</form>

```



```yaml
version: '3.9'

services:
  django:
    build: .
    container_name: django
    command: python manage.py runserver 0.0.0.0:8000
    volumes:
      - .:/app
    ports:
      - "8000:8000"
    depends_on:
      - rabbitmq

  rabbitmq:
    image: rabbitmq:3-management
    container_name: rabbitmq
    ports:
      - "5672:5672"
      - "15672:15672"

  celery:
    build: .
    container_name: celery
    command: celery -A myproject worker --loglevel=info
    volumes:
      - .:/app
    depends_on:
      - django
      - rabbitmq

```



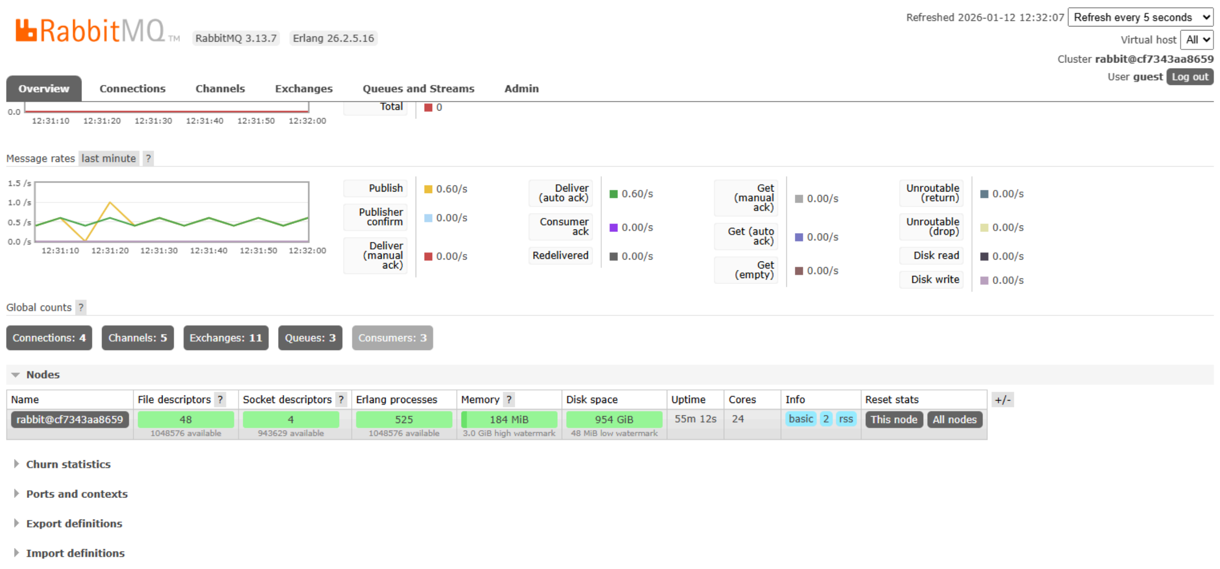
```dockerfile
FROM python:3.12-slim

WORKDIR /app

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

COPY . .

EXPOSE 8000

CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]
```



```
Django>=4.2
celery>=5.3
```

```
docker-compose up
docker ps -a
docker rm rabbitmq
docker stop rabbitmq
docker rm rabbitmq
docker ps -a
docker-compose up
cd "C:\Users\KJ Library 04\Documents\bigdog\myproject"
docker-compose build
dir
cd "C:\Users\KJ Library 04\Documents\bigdog\myproject"
```

```
notepad requirements.txt
notepad Dockerfile
notepad docker-compose.yml
dir
docker-compose build
cd "C:\Users\KJ Library 04\Documents\bigdog\myproject"
dir
ren Dockerfile.txt Dockerfile
dir
docker-compose build
```



RabbitMQ 3.13.7 Erlang 26.2.5.16

Refreshed 2026-01-12 12:32:07 Refresh every 5 seconds
Virtual host All
Cluster rabbit@cf7343aa8659
User guest Log out

Overview | Connections | Channels | Exchanges | Queues and Streams | Admin

Total	🟥 0

Message rates last minute ?

Publish	🟨 0.60/s	Deliver (auto ack)	🟩 0.60/s	Get (manual ack)	■ 0.00/s	Unroutable (return)	■ 0.00/s
Publisher confirm	■ 0.00/s	Consumer ack	🟪 0.00/s	Get (auto ack)	🟦 0.00/s	Unroutable (drop)	■ 0.00/s
Deliver (manual ack)	🟥 0.00/s	Redelivered	■ 0.00/s	Get (empty)	■ 0.00/s	Disk read	■ 0.00/s
						Disk write	■ 0.00/s

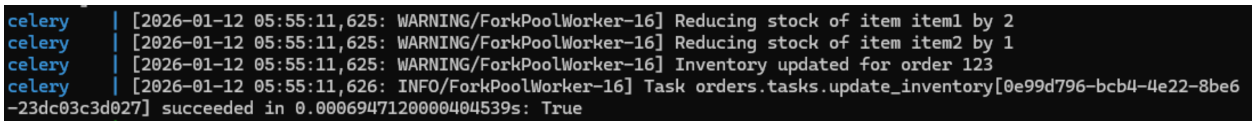
Global counts ?

Connections: 4 | Channels: 5 | Exchanges: 11 | Queues: 3 | Consumers: 3

Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Cores	Info	Reset stats
rabbit@cf7343aa8659	48 1048576 available	4 943629 available	525 1048576 available	184 MiB 3.0 GiB high watermark	954 GiB 48 MiB low watermark	55m 12s	24	basic 2 rss	This node All nodes

Churn statistics
Ports and contexts
Export definitions
Import definitions



```
celery     | [2026-01-12 05:55:11,625: WARNING/ForkPoolWorker-16] Reducing stock of item item1 by 2
celery     | [2026-01-12 05:55:11,625: WARNING/ForkPoolWorker-16] Reducing stock of item item2 by 1
celery     | [2026-01-12 05:55:11,625: WARNING/ForkPoolWorker-16] Inventory updated for order 123
celery     | [2026-01-12 05:55:11,626: INFO/ForkPoolWorker-16] Task orders.tasks.update_inventory[0e99d796-bcb4-4e22-8be6
-23dc03c3d027] succeeded in 0.0006947120000404539s: True
```