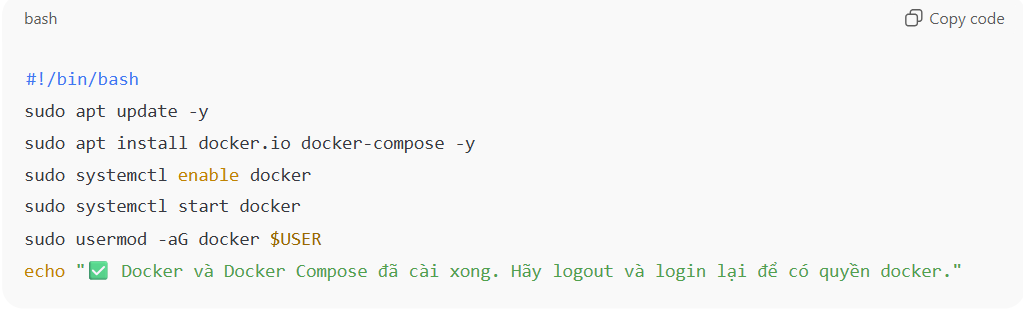
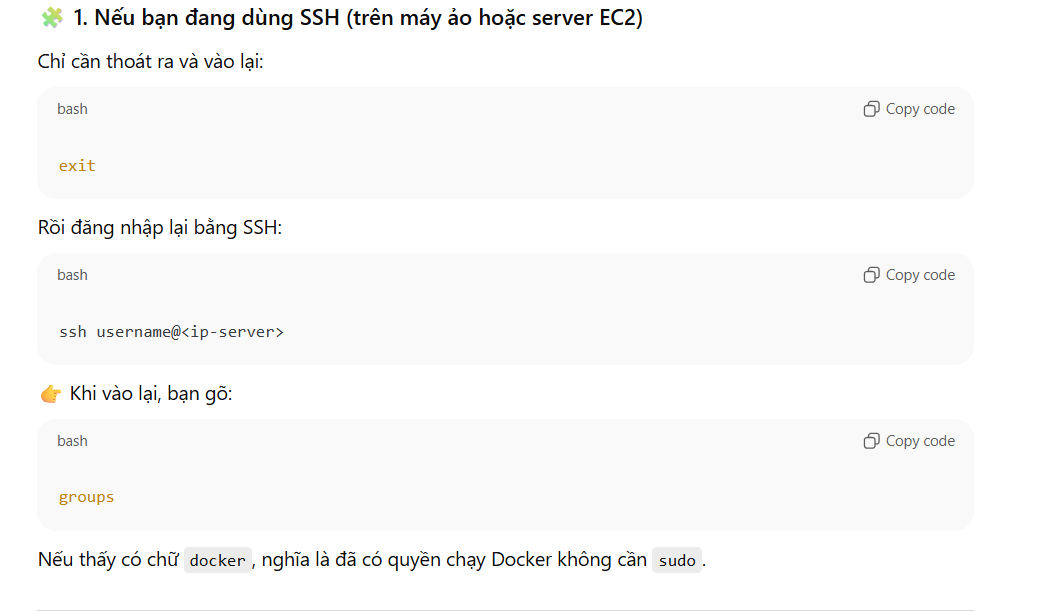
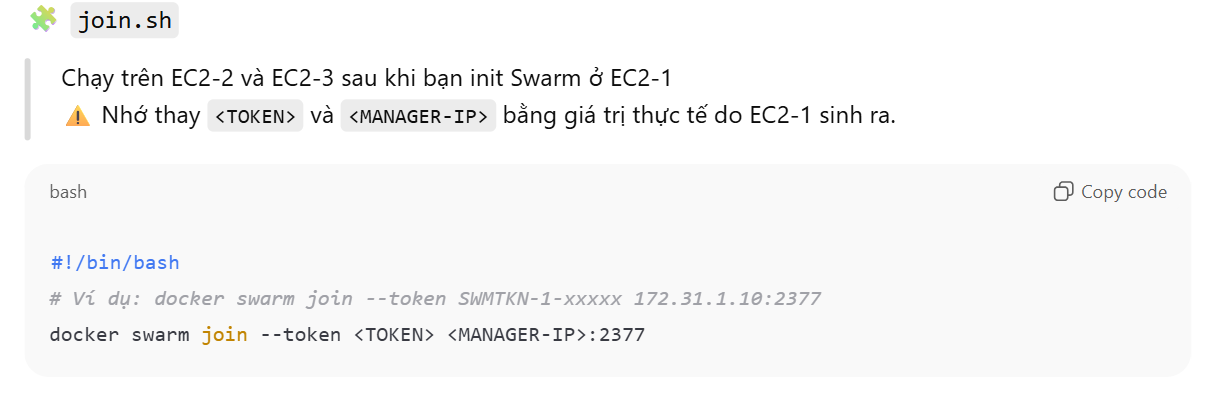


🧩 setup.sh





🧩 join.sh





**🧩 docker-stack.yml**

Dưới đây là bản cập nhật hoàn chỉnh, **không bỏ bất kỳ service nào** và thêm phần deploy cho 2 service cần scale:

|  |
| --- |
| version: "3.8"  services:  kafka:  image: bitnami/kafka:3.8.0  hostname: kafka  environment:  - KAFKA\_CFG\_NODE\_ID=0  - KAFKA\_CFG\_PROCESS\_ROLES=controller,broker  - KAFKA\_CFG\_CONTROLLER\_QUORUM\_VOTERS=0@kafka:9093  - KAFKA\_CFG\_LISTENERS=PLAINTEXT://:9092,CONTROLLER://:9093,EXTERNAL://:9094  - KAFKA\_CFG\_ADVERTISED\_LISTENERS=PLAINTEXT://kafka:9092,EXTERNAL://localhost:9094  - KAFKA\_CFG\_LISTENER\_SECURITY\_PROTOCOL\_MAP=CONTROLLER:PLAINTEXT,EXTERNAL:PLAINTEXT,PLAINTEXT:PLAINTEXT  - KAFKA\_CFG\_CONTROLLER\_LISTENER\_NAMES=CONTROLLER  - ALLOW\_PLAINTEXT\_LISTENER=yes  ports:  - "9094:9094"  networks:  - micro-net  mysql:  image: mysql:8.0.36-debian  restart: always  environment:  MYSQL\_ROOT\_PASSWORD: root  ports:  - "3306:3306"  volumes:  - mysql-data:/var/lib/mysql  - ./init:/docker-entrypoint-initdb.d  networks:  - micro-net  mongo:  image: mongo:6.0  ports:  - "27017:27017"  volumes:  - mongo-data:/data/db  networks:  - micro-net  api-gateway:  build:  context: ./api-gateway-service  dockerfile: Dockerfile  environment:  - SPRING\_PROFILES\_ACTIVE=docker  ports:  - "8888:8888"  depends\_on:  - identity-service  - profile-service  - product-service  - order-service  networks:  - micro-net  identity-service:  build:  context: ./identity-service  dockerfile: Dockerfile  depends\_on:  - mysql  environment:  - SPRING\_PROFILES\_ACTIVE=docker  - JAVA\_OPTS=-Dserver.port=8000  ports:  - "8000:8000"  networks:  - micro-net  profile-service:  build:  context: ./profile-service  dockerfile: Dockerfile  depends\_on:  - mysql  environment:  - SPRING\_PROFILES\_ACTIVE=docker  - JAVA\_OPTS=-Dserver.port=8100  ports:  - "8100:8100"  networks:  - micro-net  product-service:  build:  context: ./product-service  dockerfile: Dockerfile  depends\_on:  - mysql  volumes:  - product-uploads:/app/uploads  environment:  - SPRING\_PROFILES\_ACTIVE=docker  - JAVA\_OPTS=-Dserver.port=8200  networks:  - micro-net  order-service:  build:  context: ./order-service  dockerfile: Dockerfile  depends\_on:  - mysql  environment:  - SPRING\_PROFILES\_ACTIVE=docker  - JAVA\_OPTS=-Dserver.port=8300  networks:  - micro-net  deploy:  replicas: 3  placement:  constraints: [node.role == worker]  payment-service:  build:  context: ./payment-service  dockerfile: Dockerfile  depends\_on:  - mysql  environment:  - SPRING\_PROFILES\_ACTIVE=docker  - JAVA\_OPTS=-Dserver.port=8500  ports:  - "8500:8500"  networks:  - micro-net  notification-service:  build:  context: ./notification-service  dockerfile: Dockerfile  environment:  - SPRING\_PROFILES\_ACTIVE=docker  - JAVA\_OPTS=-Dserver.port=8700  ports:  - "8700:8700"  networks:  - micro-net  customer-summary-service:  build:  context: ./customer-summary-service  dockerfile: Dockerfile  environment:  - SPRING\_PROFILES\_ACTIVE=docker  - JAVA\_OPTS=-Dserver.port=8800  networks:  - micro-net  deploy:  replicas: 3  placement:  constraints: [node.role == worker]  volumes:  mysql-data:  mongo-data:  product-uploads:  networks:  micro-net:  driver: overlay |

