



Hackathon Challenge

Your challenge is to take this simple e-commerce backend and turn it into a fully containerized microservices setup using Docker and solid DevOps practices.

Problem Statement

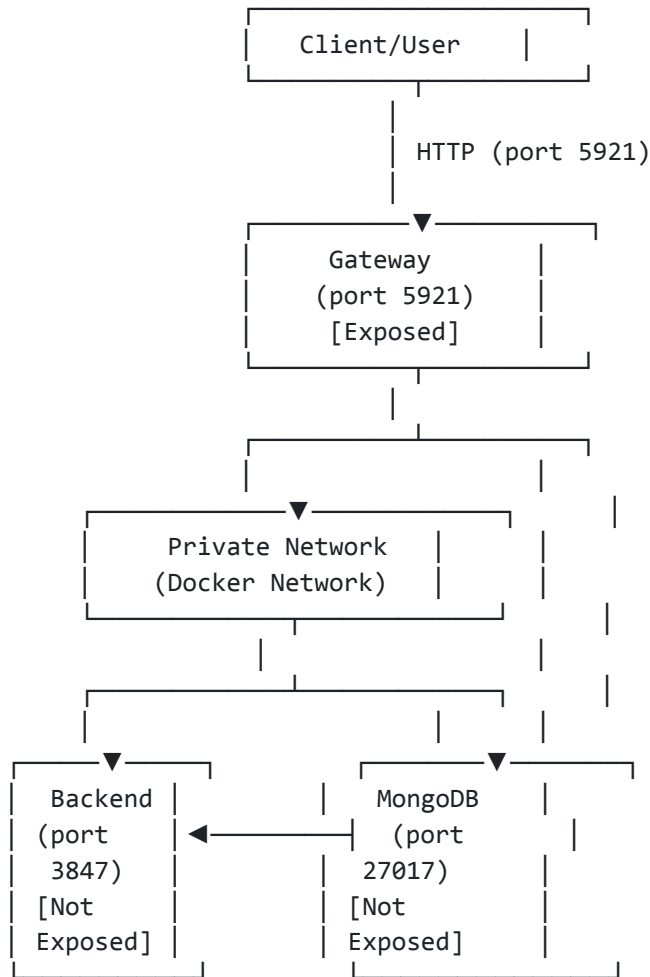
The backend setup consists of:

- A service for managing products
- A gateway that forwards API requests

The system must be containerized, secure, optimized, and maintain data persistence across container restarts.

GIT REPO: <https://github.com/bongodev/cuet-cse-fest-devops-hackathon-preli>

Architecture



Key Points:

- Gateway is the only service exposed to external clients (port 5921)
- All external requests must go through the Gateway
- Backend and MongoDB should not be exposed to the public network

Project Structure

DO NOT CHANGE THE PROJECT STRUCTURE. The following structure must be maintained:

```
.
├── backend/
│   ├── Dockerfile
│   ├── Dockerfile.dev
│   └── src/
├── gateway/
│   ├── Dockerfile
│   ├── Dockerfile.dev
│   └── src/
├── docker/
│   ├── compose.development.yaml
│   └── compose.production.yaml
├── Makefile
└── README.md
```

Environment Variables

Create a `.env` file in the root directory with the following variables (do not commit actual values):

```
MONGO_INITDB_ROOT_USERNAME=
MONGO_INITDB_ROOT_PASSWORD=
MONGO_URI=
MONGO_DATABASE=
BACKEND_PORT=3847 # DO NOT CHANGE
GATEWAY_PORT=5921 # DO NOT CHANGE
NODE_ENV=
```

Expectations (Open-ended, DO YOUR BEST!!!)

- Separate Dev and Prod configs
- Data Persistence
- Security
- Docker Image Optimization

- Makefile CLI Commands for smooth dev and prod deploy experience (TRY TO COMPLETE THE COMMANDS COMMENTED IN THE Makefile) ADD WHAT EVER BEST PRACTICES YOU KNOW

Testing

Use the following curl commands to test your implementation.

Health Checks

Check gateway health:

```
curl http://localhost:5921/health
```

Check backend health via gateway:

```
curl http://localhost:5921/api/health
```

Product Management

Create a product:

```
curl -X POST http://localhost:5921/api/products \
-H 'Content-Type: application/json' \
-d '{"name":"Test Product","price":99.99}'
```

Get all products:

```
curl http://localhost:5921/api/products
```

Security Test





Verify backend is not directly accessible (should fail or be blocked):

```
curl http://localhost:3847/api/products
```

Submission Process

1. Fork the Repository
 - Fork this repository to your GitHub account
 - The repository must remain private during the contest
2. Make Repository Public
 - In the last 5 minutes of the contest, make your repository public
 - Repositories that remain private after the contest ends will not be evaluated
3. Submit Repository URL
 - Submit your repository URL at arena.bongodev.com
 - Ensure the URL is correct and accessible
4. Code Evaluation
 - All submissions will be both automated and manually evaluated
 - Plagiarism and code copying will result in disqualification

Rules

-  **NO COPYING:** All code must be your original work. Copying code from other participants or external sources will result in immediate disqualification.
-  **NO POST-CONTEST COMMITS:** Pushing any commits to the git repository after the contest ends will result in disqualification. All work must be completed and committed before the contest deadline.
-  **Repository Visibility:** Keep your repository private during the contest, then make it public in the last 5 minutes.
-  **Submission Deadline:** Ensure your repository is public and submitted before the contest ends.

Good luck!