Job Board - POC

Specification

Build a microservices-based backend for a job posting application. The backend should have the following features:

1. **Job posting service**: responsible for managing job postings, it should be able to create, read, update, delete job postings. The job postings should have the following fields: title, description, salary, and employment type. Should have a mechanism to relate job posts with the corresponding user.
2. **User service**: responsible for managing user registration, authentication, and authorization. It should handle all the functionality related to user management like creating new users, verifying their credentials, and managing their access tokens.
3. **Search service**: responsible for providing search functionality for job postings. It should handle the indexing and querying of the job postings based on different criteria, like title, description, etc. Title and description should be searchable and salary and employment type should be filterable.
4. **Gateway service**: responsible for routing the incoming requests to the appropriate microservice, handling any other cross-cutting concerns like security, rate limiting, and logging. It should be responsible for handling the API versioning, and to handle the request validation and sanitization.
5. Microservices should communicate via REST only, each microservice should have access to its own storage (database or other). Get resource list endpoints should work with pagination.
6. Introduce a Tasks workflow that periodically checks the system's consistency, so that it can gracefully recover from cross service communication failure.

You can use your desired technologies (programming language, database, web framework, search engine). We prefer technologies from the Javascript ecosystem (e.g. NestJS, elastic search).

Deliverables:

- A design (drawing with whatever technology you prefer) that showcases how the communication between services occurs.
- A repository that contains the code.
- All apps as docker containers.
- A docker-compose (or similar technology, e.g. K8S) definition that enables us to "run" the system alongside deployment instructions.
- Security report with the system's security details (e.g. auth based on JWT tokens served with cookies).
- Provide unit, functional and e2e tests.