Explanatory Note:

The purpose of this tech assessment is to get a grasp of your level of knowledge in various concepts, and your level of experience in implementing these concepts.

There is no need to answer any of these questions in relation to the specific technologies listed in the job spec, unless that is what you're most comfortable in. For example, if your answer includes serverless functions, then feel free to discuss AWS Lambdas or Azure Functions or GCP Cloud Functions, etc; whatever you know best.

There is no right answer for any of these, the questions they are deliberately open and subject to opinion.

Your answers will form part of the final interview, so please remember why you chose them!

If you feel any question doesn't have enough information to give a full answer, feel free to make assumptions.

Scenario

The Company platform is a cloud-deployed solution which has a browser-based UI for individual customers. The backend has web API endpoints providing authenticated services for the web UI and also for 3rd parties to send bulk or direct requests to the platform.

A developer is building out a new application which will provide tools for the internal (non-technical) staff at the Company to perform administrative tasks on the platform such as updating customer data or modifying orders. The UI is a Node.js application, and the backend will expose 20 API endpoints. The backend will interact directly with its own MySQL database (storing some sensitive data), will interact with various API endpoints for third parties and API endpoints in the existing platform.

Questions

- It is important for this developer and the rest of the team, that this application can be built and deployed independently of any other releases, and that changes to the architecture of application do not accidentally change/compromise the infrastructure of the existing platform. The new application will only need to call the APIs provided by the existing platform and does not require access to its databases, queues, etc.
    - Where/how would you recommend we begin to deploy this application's infrastructure? (Just a few words, not a full description of the architecture, that's later!)
    - What CI/CD system(s) would you recommend in this case?

- The Node.js frontend is effectively a static website consisting of HTML and optimised javascript, making calls to the backend and the existing platform through the APIs. The volume of concurrent users is less than 100, but they are geographically diverse and responsiveness is important.
    - What approach would you recommend to host the front end and why?

- The developer is unsure whether they should build the backend as a single web application providing all 20 API endpoints, or as individual functions built and deployed independently for each endpoint. They have asked you to advise from an infrastructure perspective;
    - Provide a brief description of the main components/cloud services you would use in both approaches;
      Any format of answer here is fine - text description, drawings, IaC scripts, whatever you are most comfortable with.
    - Provide a list of the main pros and cons to each approach (don't go crazy, 3-5 for each)

- In regard to the database for the backend:
    - What are the main considerations for creating the database?
    - What would the architecture look like (if you haven't already included it above)?
    - How would you deploy schema changes across environments?

- The stakeholders expect 99% uptime for this application
    - Provide a high-level indication of what parts of the system you would monitor to achieve this.
      We don't need alert thresholds, just the things that need to be watched so we have a good idea of what is broken when things are broken