# Code Challenge: Data ingestion pipeline with cloud integration

Below, you will find our Python skills assessment for the Software Engineer role that we use to see a working example of your coding technique and to see your personal style. I am very excited to see what you come up with and hope you find this skills challenge rewarding.

Objective  
 Build a data ingestion service that collects logs from a public API, processes them, and stores them in a cloud-native storage option.

## Project Requirements

### **Data collection:**

* Fetch data from the provided API endpoint:
  + API: <https://jsonplaceholder.typicode.com/posts>
  + This API provides placeholder data that should be ingested and transformed.

### **Data transformation:**

* Transform the collected data by adding the following fields:
  + ingested\_at: UTC timestamp of when the data was ingested.
  + source: Static string representing the data source (e.g., "placeholder\_api").

### **Data storage:**

* Choose a cloud-native storage option suitable for moderately structured log data. Justify your choice.

### **Containerization:**

* Dockerize the application to run it consistently across environments.
* Provide a Dockerfile and a docker-compose.yml (if needed) for local testing.

### **Testing:**

* Write unit tests for data transformation logic.
* Test edge cases such as API timeouts, invalid responses, and database errors.

### **Documentation:**

* Provide clear instructions to run the application locally and deploy to a cloud environment.kj
* Document API endpoints, transformation logic, and database schema.
* What trade-offs did you consider?
  + TODO
* What were the hardest parts to implement and why?
  + TODO
* What would you improve if you had more time?

Bonus Points (Optional):

* Implement CI/CD with GitHub Actions or Jenkins to automate testing and deployment.
* Implement a REST API endpoint to retrieve the ingested data.
* How would you track the latest successful data ingestion? Describe your approach, including any challenges you anticipate, and the trade-offs involved.
* Implement integration tests to validate end-to-end ingestion with cloud storage.
* Surprise us with some other feature

## Expectations

We understand that AI tools are part of modern workflows. For this challenge, you may use them for ideas or debugging help—but we expect the architecture, code, and decisions to reflect your own work. You should be able to explain and extend any part of it during the live follow-up session.

**What we are mainly looking for:**

* Programming style/best practices
* Git repo setup
* Testing methodology
* Strengths/Weaknesses
* Modularity and maintainability: Easy to extend and follow.
* Creativity

## Submission Guidelines

* Submit the project as a GitHub repository link.
* Ensure all necessary cloud credentials are configurable via environment variables.
* Provide a README.md with:
  + Setup instructions
  + Running the application
  + Running tests
  + Deploying to a cloud environment
  + Documentation
* Before submitting, your code must run locally on your machine. Please ensure that we can pull it down and run it without any setup issues. If we encounter problems starting the project and need to debug it ourselves, your submission will be disqualified.
* Please submit this challenge within one week (5-7 days). If you need more time, please feel free to reach out to request.

Feel free to reach out to me with any questions you may have. Good luck!