# *Site Reliability Engineering Challenge*

---

At adidas we are quite concerned about reliability and quality of our developments. We would like to propose you a challenge to solve, similar to the type of challenges you would be facing in our daily work at adidas.

In this challenge, you'll be interacting with **Product Service**. This service is in charge of providing product information such as **product ID, product name, model number, description, number of reviews, average review score, etc**… to ecom sites, mobile apps and other internal applications. To provide this information, Product Service relies on an external **Product Review** service that provides the number of reviews and average review score for each product. Product Review is **OUT** of scope for this challenge and a mock implementation is provided to better understand the interaction between both services.

**The challenge is comprised of two exercises**:

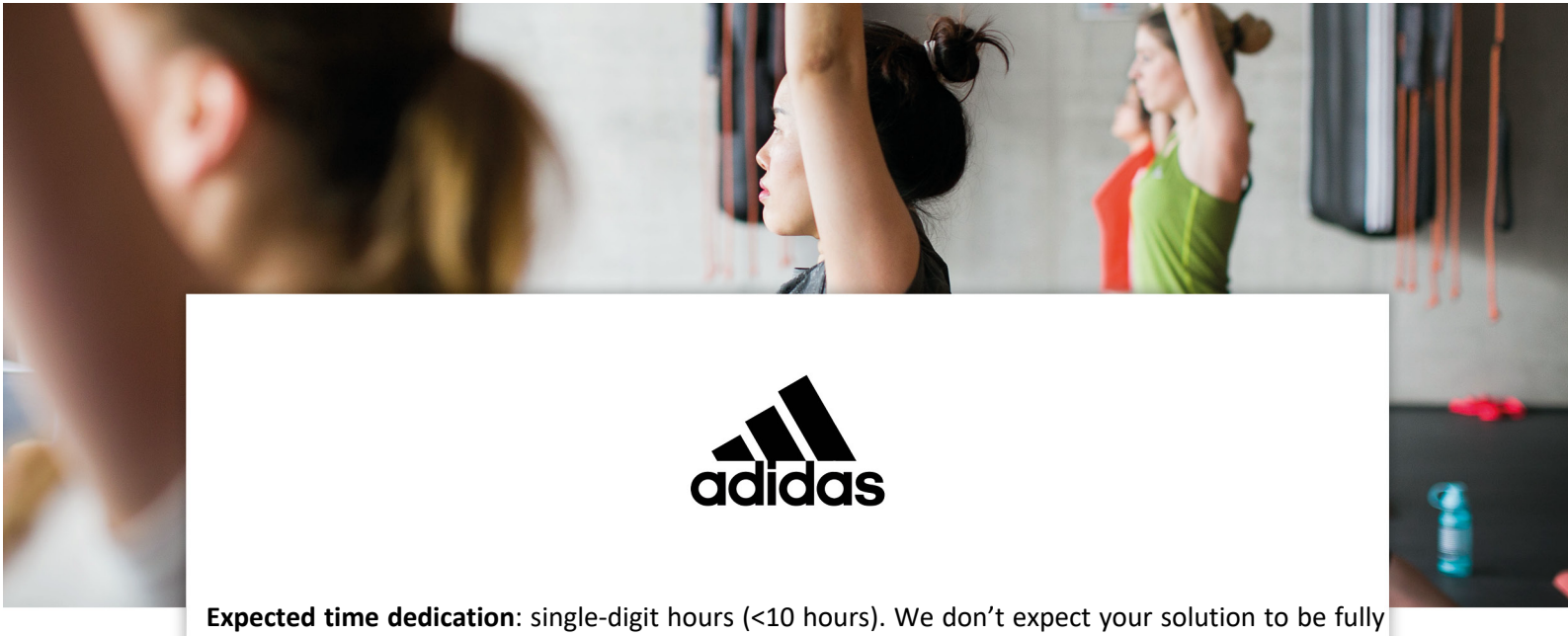
- Solution design document
- Solution implementation.

In the **solution design document**, we ask you to propose improvements on Product Service focusing on a set of topics. We expect you to provide a document (you can include diagrams, drawings or anything that you see fit) with your proposals for the different topics. We expect a high-level proposal for most of the topics, and a more detailed proposal for one or two of the topics of your choosing.

In the **solution implementation**, we ask you to implement one of your proposals. Please, don't spend effort or money setting up infrastructure for this challenge; for example, if your proposal for CI/CD was a set of Jenkins pipelines, we expect to see in the repo the Jenkinsfile (or files) but we don't expect you to deploy a Jenkins instance with its worker nodes and everything needed to run Jenkins. You can assume the required infrastructure will be there, and you can explicitly set those assumptions on both solution design document and solution implementation.

**Expected deliverables:**

- ✓ Solution design document
- ✓ Git repo with solution implementation



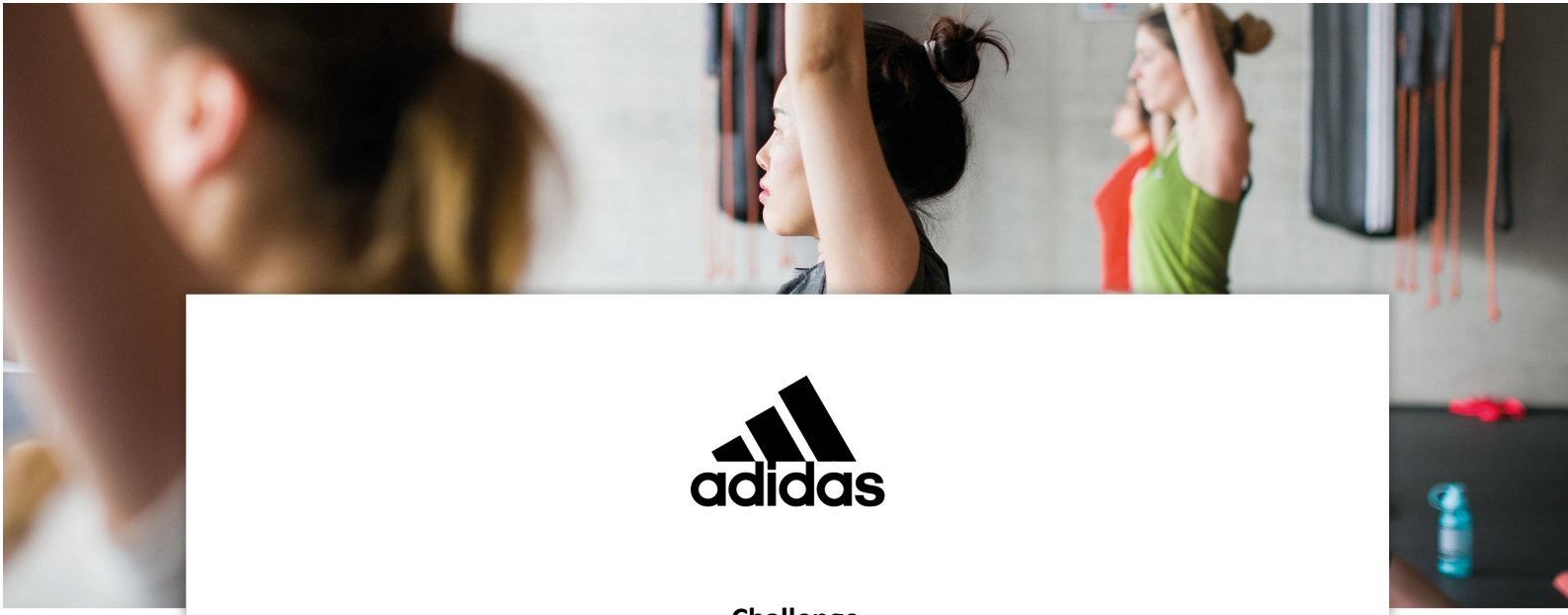
**Expected time dedication**: single-digit hours (<10 hours). We don't expect your solution to be fully detailed and complete; a **posterior challenge review session will be scheduled where you'll have the chance to explain your ideas and implementation**.

**We want this challenge to be a trigger for a meaningful conversation with you**. You'll be able to explain your decisions in a follow-up session with the adidas team.

Our favorite technologies are: Node.js, Java, Spring, Docker, Jenkins, AWS, Terraform, Kubernetes, Grafana, Prometheus, Kibana-Elasticsearch but we are open to hear about any other if you are more experienced, you think it would be more appropriate to use, or you feel more confident with other tools.

Thank you very much for your time and effort. If you have any doubts or are not clear on what is expected, please do not hesitate to contact us.

(**Kindly note:** It is not allowed to upload this PDF document of the coding challenge on a public repository (e.g. Github). Kindly only upload your solution. Thank you!)

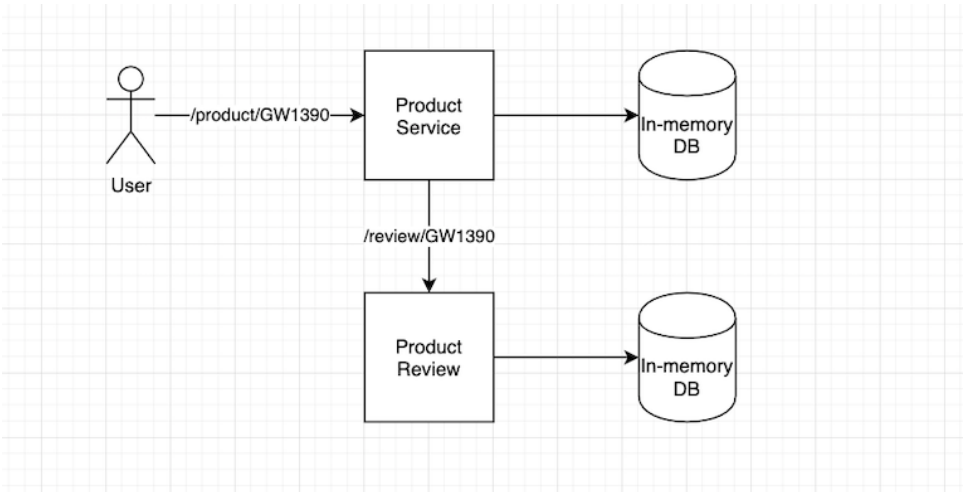


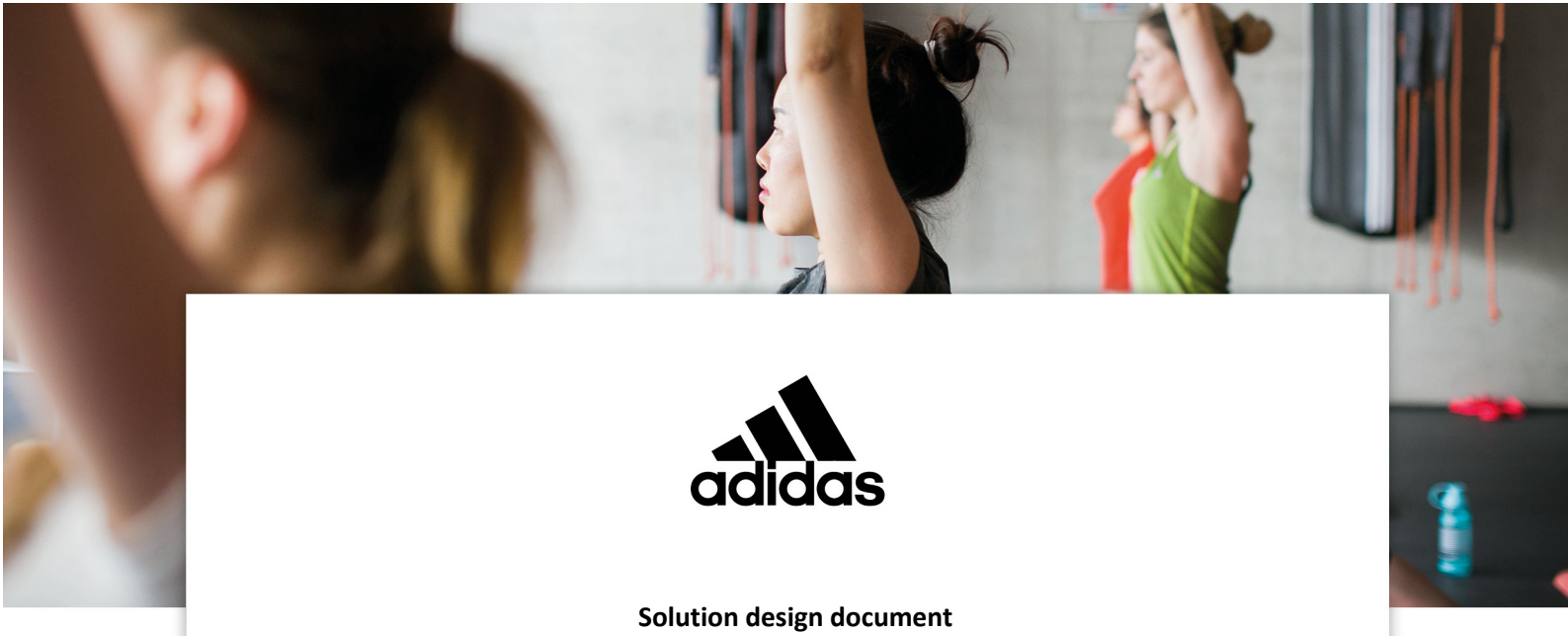
**Challenge**

https://github.com/adidas/product-manage-challenge

Let's imagine that a new Product Service is being developed and you are part of the SRE team. This **Product Service** is critical for our business as it's one of the sources for our ecom sites, mobile apps and several internal apps; so, it must be available and reliably be able to fulfill expectations. The product-service enriches product data with reviews provided by an external service: product-reviews.

You'll be interacting with **Product Service**. This service provides product information such as **product ID, product name, model number, description, number of reviews, average review score, etc**… to ecom sites, mobile apps and other internal applications. To provide this information, Product Service relies on an external **Product Review** service that provides the number of reviews and average review score for each product. Product Review is **OUT** of scope for this challenge and a mock implementation is provided to better understand the interaction between both services.





**Solution design document**

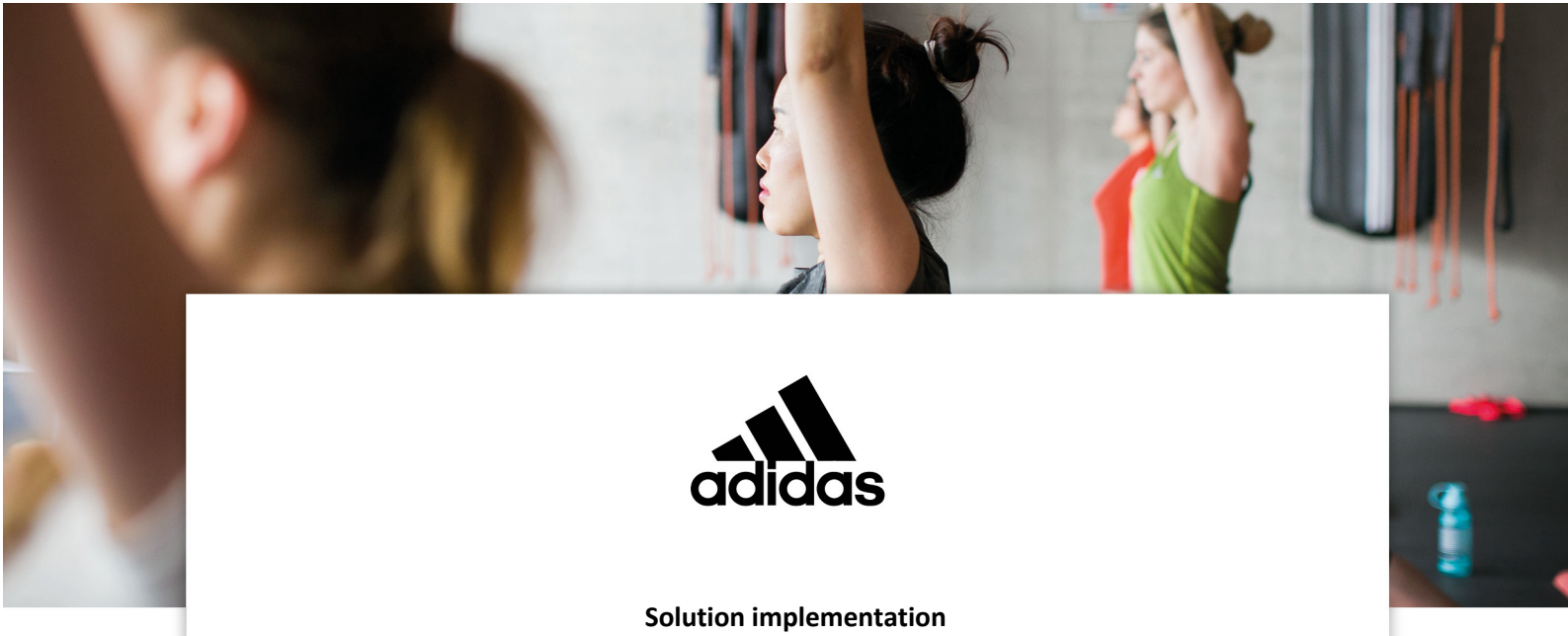
Therefore, we would like you to write a document which address the following topics, please note that we don't want to limit you in anyway or on anything, feel free to use diagrams, images, etc; show us your best ideas:

- ✓ CI/CD: Description of a pipeline, where every commit of each developer goes through phases (stages) finishing on a development environment and promoting to production environment.
- ✓ Observability: Description of what would you implement to improve application observability. What would you do to enable quick and easy debug of the application?
- ✓ SLOs: Our stakeholders expect to receive product data 99.5% of the requests that they made to product-service within 200ms. Describe how would you measure and alert to ensure SLOs are accomplished.
- ✓ Deployment: Description of a strategy for deploying the application containers in a high availability environment without downtime. How would you improve the resilience of the application?
- ✓ Resiliency: Currently, if product reviews service is down, we won't serve any product data at all. Describe how would you improve product-service in a way that can continue fulfilling (at least partly) the expectations in the event that product reviews service is unavailable in a region or globally.

These topics are wide in scope on purpose, to limit time and effort put into this challenge we suggest providing a high-level overview on most of the topics where you show key elements on how you would design them and choosing one or two where you dive deeper into details, tradeoffs and risks in your proposal, alternatives you've considered, … and highlight your strengths.

**Trigger questions to help you on your solution design:**

1. How can we reduce the time between a developer commits some changes and those changes are deployed in production? (CI/CD)
2. How can we limit impact if a commit causes an outage? (Resiliency - Deployment)
3. How would you propose the deployment of hotfixes or new features? (CI/CD)
4. How easy would it be to check what is going on in your environment? How are outages detected? (Observability)
5. What would you do to ensure availability and performance under high load peak events? What if they are unforeseeable? (Resiliency – Deployment)
6. What KPIs would you report to a business stakeholder? (SLOs)



**Solution implementation**

From your solution design topics (CI/CD, Observability, SLOs, Deployment, Resiliency) we'd like you to pick **just one** and implement the solution proposed. You can choose your favorite one, even if it's one of the high-level overview.

Please, create a private fork repo under your own username for your implementation. You'll be able to show and explain your implementation details in the posterior challenge review session.

**Source**

Product-service: https://github.com/adidas/product-manage-challenge

**Looking forward to seeing your solution!**