Objectives

With this challenge, we want to evaluate your ability to solve real-world problems by applying system design principles and Cloud architecture best practices. Therefore, we recommend being as detailed and explanatory as possible in the solution you build, since this will help us understand your way of thinking and solving problems, as well as your ability to apply theoretical knowledge in practice.

If the concepts and technologies mentioned in the challenge are completely new to you, don't worry — the goal is also to evaluate your ability to learn new technologies and tools. You may ask any questions you consider necessary; we will try to guide you within the limits of the challenge.

If you have any doubts regarding the statement or objectives of the challenge, you can contact us by email, and we will try to respond as soon as possible.

For the development of the solution, we recommend focusing on the following aspects:

- Fault tolerance
- Scalability
- Disaster recovery

Statement

Our client, Project X, wants to implement a new 3-tier infrastructure in AWS in their production environment. This consists of a Frontend application layer, a Backend application layer, and a Database layer. Before implementing the solution, the client asked us to design the architecture, including network infrastructure components, applications, etc. The client emphasized business continuity and disaster recovery, so the solution must include a disaster recovery strategy aligned with the client's required timelines (detailed below).

To simplify the challenge, here is how the different components in the 3-tier infrastructure are defined:

- **Frontend:** There are two frontend applications, *home* and *checkout*, which must run on ECS and must be in High Availability (HA).
- **Backend:** There are three applications: *payments*, *products*, and *shipping*. All of them must run on Lambda. Only the *products* app needs access to the database; the others only connect to external APIs.
- **Database:** PostgreSQL running on RDS. All applications use the same RDS cluster.
- **S3 buckets:** There are two S3 buckets used by the backend application to store metadata about payments and shipping orders.

The new architecture must meet the following requirements requested by the client:

- **High Availability:** The infrastructure must be configured so that the failure of an AWS Availability Zone does not affect the functioning of the application. This must apply to each component mentioned above (Frontend, Backend, and DB).
- **The infrastructure must withstand the failure of an entire AWS region.** The client is willing to tolerate a few minutes of downtime and data loss, not exceeding 10 minutes.
- **The region failover process must be monitored and semi-automated**, where the client only needs to "press a button" to switch to the recovery region.
- **Networks must be segmented** based on their functions. For example, the Backend applications must be separated at the network level from the Database applications.
- **Communication between applications must be protected using firewalls.**
- **From internal networks, only outbound connections must be allowed**—no inbound connections should be permitted.

The ports used by each application are:

- **Frontend:** TCP/80 and TCP/443
- **Backend:** Not exposed at the network level; each Lambda function is invoked through its corresponding API.
- **Database:** TCP/5432

Deliverables

The following points are the required deliverables for the challenge. We recommend including as much detail as possible so that the reviewer can fully understand how you developed the solution.

- **Architecture diagram** showing the different components and how they are connected. You must also include detailed configurations of key components, such as route tables, Security Group rules, IAM roles, inter-region connections, etc.
- **Documentation and explanation of the architecture.** Here it is important to explain and justify the design decisions. The more information you provide, the faster we will understand your reasoning and problem-solving approach.
- **Recommendations for future improvements** to the architecture.
- **Add missing infrastructure components:** if you detect that the challenge does not specify certain infrastructure components that are needed, you may add them and justify the design decision. We highly value initiative and proactivity. You have the freedom to add any components you consider necessary, as long as you explain why you chose to include them.