Summary of Design Decisions:

Mohammed Saber BELLAAZIRI - Bouthayna ATIK

1. **Infrastructure**: We decided to create a custom network stack with public and private subnets. The EC2 instance will be placed in the public subnet, and the RDS database will be placed in the private subnet. This setup ensures that only the EC2 instance can access the RDS database, providing an additional layer of security.

2. **EC2 Instance**: We chose to use the t2.micro instance type to run the PHP application. To provide secure access to the EC2 instance, we used EC2 Instance Connect, which allows administrators to connect without the need for a key pair.

3. **RDS Database**: For the database, we selected the MySQL engine with the db.t2.micro instance type. We made sure to enable the Free Tier options to keep costs low. The RDS database will be hosted in the private subnet, ensuring that it is not publicly accessible.

4. **Security Groups**: We created separate security groups for the EC2 instance and the RDS database. The security group associated with the EC2 instance allows inbound traffic on port 22 (SSH) and port 80 (HTTP) for anonymous access. The security group associated with the RDS database allows inbound traffic from the EC2 instance's security group on the MySQL port (usually 3306).

5. **IAM Roles:** We created an IAM role that allows the EC2 instance to access the Systems Manager Parameter Store to retrieve database connection information securely. The role is granted the necessary permissions with the principle of least privilege.

6. **Systems Manager Parameter Store**: The PHP application retrieves the database connection information (endpoint, username, password, and database name) from the Systems Manager Parameter Store. We stored the parameters under the path “/myapp/database/".

7. **PHP Application Deployment**: We downloaded the PHP application code and SQL dump files. After launching the EC2 instance, we used user data to install Apache, PHP, and MariaDB on the instance. We then uploaded the PHP application code to the web server's document root (/var/www/html) and imported the SQL dump into the RDS database.

8. **Cloud9**: To avoid issues, we used the Cloud9 service for development and deployment. This provided an integrated development environment and allowed us to execute shell scripts inside the created EC2 instance directly from the Cloud9 interface.

9. **AWS QuickSight**: If applicable, we used AWS QuickSight to create data visualization dashboards using the sample dataset provided. We explored different visualization options and created a dashboard to showcase insights and data trends.

10. **Security and Cost Considerations**: Throughout the design process, we were mindful of security best practices and adopted the principle of least privilege for IAM permissions. We also considered cost optimization by utilizing Free Tier options, stopping or terminating resources when not in use, and avoiding unnecessary resource allocation.

By following these design decisions, we were able to deploy a secure, scalable, and cost-effective solution for the Example Social Research Organization. The PHP application runs on an EC2 instance, securely accessing the RDS database with its connection information stored in the Systems Manager Parameter Store. The infrastructure is set up in a custom network stack, ensuring proper isolation and security. AWS QuickSight is used to create insightful data visualizations.

(All screenshots are included in the file aws\_project\_BELLAAZIRI\_ATIK.pdf)