



# Reducing the Application Deployment Costs via Caching

In this lesson, we will discuss a real-world example of how to reduce the deployment cost of an application by using a cache.

## We'll cover the following ^

- Real-life use case
- Conclusion

## Real-life use case#

In this lesson, I am going to share an insight from a stock market-based gaming app that I developed and deployed on the cloud.

The game had several stocks of companies listed on the stock market and the algorithm would trigger the stocks' price movement every second, if not before.

Initially, I persisted in the updated stock price in the database as soon as the prices changed to create a stock price movement timeline at the end of the day. However, so many databases writes would cost a fortune. The number of writes every hour was just crazy.

Eventually, I decided to not persist the updated price every second in the database and use *Memcache* to persist the stock prices. Then, I ran a batch operation at regular intervals to update the database.

*Memcache* was comparatively a lot cheaper than the disk-based database

access. The cache served all the stock price requests, and the database did not have the updated values until the batch operation ran.

## Conclusion#

This tweak may not be ideal for a real-life Fintech app. However, it helped me save a truckload of money, and I was able to run the game for a longer period of time.

So, this is one instance where you can leverage the caching mechanism to cut down costs. You might not want to persist each and every information in the database and instead, use the cache to store not so mission-critical information.

Now, let's look into some of the caching strategies we can leverage to further enhance the app's performance.

[← Back](#)[Do I Need A Cache?](#)[Next →](#)[Caching Strategies](#)[Mark as Completed](#)[Report an Issue](#)