# Tiny URL System design

## Problem Statement

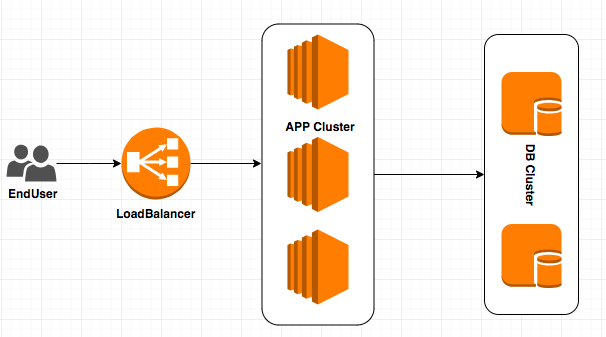
* URLs of the resource can be large, and not so easy to share with others.
* Managing long URLs is also bit difficult.

Solution:

Shortening the URLs.

* Easy to share.
* Can be manageable easily.
* You can simply put more links, it saves the space.
* Social media posts with short URLs looks good.

## High-Level Architecture:



## Tech Stack

1. JAVA
2. SpringBoot framework
3. Couchbase datastore
4. Docker
5. Gradle build tool

## Design Consideration.

1. Size of the Long URL.
   1. Upto 2048 bytes.
2. Size of the unique key.
   1. Upto 6 bytes.
   2. Will be considering [A-Z, a-z, 0-9] characters to generate unique key for each long URL.
   3. Possible number of unique keys can be generated is 64^6.
3. Request throughput.
   1. 5000 requests per day to generate shorten URL.
   2. 50000 requests per day to redirect to original URL.

Read Write ratio is 10:1. So, application should be READ intensive.

1. Unique Key Generation.
   1. Calculate CRC32 hash for the input long URL.
   2. Encode hash with [A-Z, a-z, 0-9] characters.
2. Expiry of URL.
   1. Couchbase supports TTL property for each document, after TTL record get purged automatically.
3. Duplicate URL.
   1. Unique key will be considered as primary key, and make sure our database rejects insert for duplications.
   2. If unique key conflicts, consider timestamp to generate hash and retry.
4. High Availability
   1. Consider cloud deployment, application and database nodes spawned across the multiple availability zones to achieve high availability.
   2. Couchbase is high available and portioned data store through its architecture.
5. Distributed database and replication
   1. Couchbase is a distributed document datastore.
   2. It supports replication of data, and it can be configurable.
   3. Couchbase uses hash based sharding to store vbuckets across the cluster.
   4. Couchbase java SDK maintains clustermap, which has mapping of keys and on which node document persisted. It improves the read performance.