



Case Study

Leverage the advantage of containers

Lloyd Armstrong

Jenkins Pipeline

A Jenkins pipeline for this Case Study would be ideal for encouraging an agile environment. Developers could make changes locally, push their changes to Git and leave the rest to Jenkins.

Jenkins will pull the latest changes from Git, destroy previously built containers of older versions and then build new containers with the updated code by prompting Docker Compose to orchestrate the new release.

Log Handling

There are numerous options for log handling, however, for this case I'll suggest two:

I. Syslog or JSON Docker Logging Driver

Docker has multiple logging drivers that can be easily configured. The syslog driver will help send all logs to a centralized syslog server. The JSON logging driver will capture all ``stdout`` and ``stderr`` logs and outputs information for each container in a separate file.

II. Logstash

As a personal choice, I prefer logstash. Logstash gives you the ability to ingest and manipulate logs into any format you chose, and output the logs into a high-performance search-engine (like Elasticsearch). Logstash will also help with the next point: Monitoring.

Monitoring

Monitoring can be implemented in a variety of ways. At the most basic level, a CRON job can be configured to call an API that sends daily health updates, failures, and warnings to a Telegram bot. Using Logstash for our log handling will also allow for us to send logs into Elasticsearch and use Kibana to visualize logs in an organized layout.

High Availability

Starting with the MySQL instance, in order to create a high level of redundancy, creating a MySQL slave instance would be the logical approach. If the database container goes down, the slave will be easily available and up to date to switch over to.

The web server can become redundant by creating a duplicate Nginx instance and load balancing between the 2 containers. In the case of one going down, all traffic will be routed through the one surviving instance, with zero downtime.

We can use HAProxy for creating a high-availability environment across all the containers. It especially works well with PHP-FPM instances.

Possible Improvements

One clear issue with this project in its most basic level is the security of credentials. For this project, the credentials to make a connection to the MySQL are in plain text. This is far from ideal and insecure. In a production environment, a way to handle this is to place the credentials in a local `.env` file where the PHP of the project can reach it.

Once the infrastructure was stable and redundancy was correctly implemented, we could look at making the application more seamless. A possible improvement could be adding Redis to the environment. Implementing a Redis server as a cache can help improve database performance and deliver quicker results to the end-user.

Beyond these back-end changes, a streamlined and useful front-end could be added to make the project more aesthetically pleasing.