Cloud System Architecture

Prepared by: Jose E. Padilla

Date: July 2, 2014

# Description

A Cloud System Architecture is basically the same as a System Architecture, only that its for the Cloud. Their are various approaches in Cloud Computing, SaaS, PaaS, and IaaS.

SaaS - Software as a Service

PaaS - Platform as a Service

IaaS - Infrastructure as a Service

The approach taken in the diagram is SaaS, since it is the most common and direct way to Cloud Compute. On the left side of the diagram, you will see subsystems needed for the cloud to work as a cloud. Some of them are, Database, Content, Queue, Logging, and Monitoring. In the following paragraphs, I will explain every component of the cloud and the reasons of why we need it.

# Layers

The Database layer of the cloud is obviously needed to store data. This part of the cloud is essential and must be well organized and structured.

The Content layer is needed for serving the data that we have in our Database. All of the data, files, and others, will be served using a Restful api(Application Programming Interface). By serving all of our content this way, will easily provide access to any application built on any platform.

The Queue is pretty much like the Queue we learned in our Data Structures course. This is used for making heavy tasks asynchronous. When we talk about asynchronous tasks, what we are actually saying is that, we could have thousands of request falling to our application and the problem we face is that we can not handle all those requests at the same time. Therefore, we could have users stuck on our application, because thousands or millions of people want to run a heavy processes that is slowing the response time. That is when asynchronous task help us out, by adding processes to a queue and in sometime in the future, the process will execute and we can handle thousands of requests at the same time. This should only be used for long processing tasks, for example, sending emails.

The Logging system will help us store necessary details that we could use when are application crashes unexpectedly. A good example of a logging system is Sentry. Sentry will send specific users and email for every error that the application may have with details about the time and date, url and more.

Finally, the Monitoring layer is used to see how many people are making requests to our site, how long is the server taking to respond, location of where the requests are coming from, and more. We could use Google Analytics, since its the most easy way to integrate to our site. Also, we could integrate our own custom monitoring techniques and get more stats about our application.

# Notes

Please be aware that all of the information and the diagram, if any, related to this document is subject to change and revisions by the members of the Bookshub project. Any comments, recommendations and doubts, please open an issue on the Github repository so we can all be on the same track and discuss the issue as a group.