**Monolithic Applications:**

* The underlying architecture of monolithic applications is monolithic, which is designed to deploy as a single, indivisible unit. Monolithic architecture says that all the component of the application tightly integrated and run in a single process space. This sort of applications have easier development, deployment, and management, especially for small to medium size applications. Traditionally, all monolithic applications were deployed in premise but nowadays they are hosted on cloud due to cloud computing.

**Advantages of Monolithic applications:**

* Give that the whole application is built to be run on the same process. The initial process of developing and deploying is lot easier.
* Simple to develop and deploy
  + Given that whole application is built to be run on the same process. The process of developing and deploying is lot easier. The developers have to constantly test the applications functionality and ensure that they are what is required from the application, this is lot easier to do in monolithic applications because everything to do with the application can be invoked on single run.
* Ease of debugging and Testing
  + Testing is easy in monolithic because all you have to do is think about the single process. (I ACTUALLY DON’T KNOW WHY TESTING IS EASIER)
* Consistency:
  + Less duplication of the code. Easier to enforce coding practices. [FIND AN EXAMPLE]
* Less Operational Overhead
  + Things such as single database and tight coupling means that there is no need to design how different components will interact. Of course, this might lead to applications with badly structured code base.
* Small Monolithic applications are easy to maintain.
  + Small applications will not have complex design of how the different components will interact mainly because there are very few components. Tracking how they interact is much easier this way.

**Disadvantages of monolithic applications:**

* Scalability challenges
  + One of the main reasons, monolithic applications are going out of fashion is because the can only be scaled vertically, more about it later.
* Difficulty in Adopting New Technologies
  + An application has many parts each part might function differently and there might be better technologies to each part more efficiently and effectively. However, with monolithic you can only use on technology to write it.
* Increased Risk of Deployment
  + During re-deployment, sometimes it is likely that you tampered with a function that messed up the functionality of the other.
* Longer Development Time as the application grow
  + Developer need to spend more time understanding the application as a whole as slight change in code can affect other functionality.
* Limited Flexibility
  + Change to one part of the application might make it so that another part of the application needs to be updated as well as the application is tightly coupled.
* Continuous deployment becomes more challenging
  + Once again we come to the part where a single bug or change affect the whole applications.
* More challenging to make applications reliable and fault proof
  + A singular bug can take down the application.
* Team collaboration becomes more difficult as the code base grows
  + As team grows which happens if the application grows, it is very difficult to segregate the duties as the application is tightly coupled give that they run on the same process.