1. **Introduction**
   1. Background
   2. Problem statement
   3. Project Goals
   4. Motivation
   5. Limitations
   6. Report Structure

Name my project:

Background:

1. I am going to a talk about software and when they were made and how they were made and how they were deployed.
2. Then I will talk about how cloud made the deployment of monolithic application easier.
3. Then I will talk about how cloud computing gave raise to cloud native which primarily use microservice architecture.

**Background:**

[Some of the sentences in the following paragraph are one to one copy from the source. Remember to present it in a different way.]

The first web server in the world was deployed at CERN in Geneva, Switzerland in 1990. By late 1993, there were 500 known we servers and WWW accounted for 1% of inter traffic [source]. Suffice to say they were all hosted on premises. Today as I am writing this, according to [source], there are [number] of web servers.

In 2006, amazon launched AWS with Elastic Compute Cloud (EC2) service followed soon by google’s Google App engine (2008) and Microsoft’s Azure (2009). The could services transformed the landscape of web-server deployment. After this point, business, individual, or organizations no longer needed to host there software on premise. However, at this point the technology was still quite primitive, most applications were followed monolithic architecture (More about it later).

The first well known application to use cloud native approach was Netflix. It took them two years to transform their monolithic architecture to that of microservice.

**Problem statement:**

[Some of the sentences in the following paragraph are one to one copy from the source. Remember to present it in a different way.]

Most of the world’s web applications are monolithic in nature. They don’t take full advantage of the cloud computing. More companies would benefit from moving toward more cloud native applications.

**Project goals:**

[Some of the sentences in the following paragraph are one to one copy from the source. Remember to present it in a different way.]

Monolithic applications were industry standard for a web application from 1960s. They are perfect for small to medium size applications that don’t have large user base. However, internet is present everywhere these days. The number of users for an application can reach from thousands to millions to hundreds of millions. On top of that, technologies change and market keep on changing regularly. With these new factors, monolithic applications have started to hit bottle necks. Updating and adding new features to increasily changing market presents a problem especially if the monolithic application is already big. It might introduce a new bug, it make take a long time to introduce a new feature, it might take a long time to test give the size of the application, on top of that undected bug can take the entire application down. And how are you to know if you user size grows in the future. It took Netflix two years to change their monolithic applications to cloud native. Should they have stared with microservice application straight away. But what if the Netflix as an application failed if they had started as cloud native applications.

An alternative that appease these problem is of course cloud native applications with microservice architecture. You can introduce features quickly, you decrease cognitive overhead, and multiple teams can work on the same application simultaneously give the principle of loose coupling. They can scalled horizontally as well as vertically much easily monothic applications. This sounds like exactly what we needed to solve the problem of monolithic architecture. Yet, it has its flaws. Initial design stage takes longer, is just too complicated for small to medium applications with small user base , is costly on short term.

Given these complex properties of each type of application type, to a normal software developer, it isn’t exactly obvious what is the best choice. People and companies spend hours trying to understand what is best for there specific needs. They have specific people employed to make these decisions. But how do these specific people know what is good for a software. I am willing to be a scapegoat to provide resources to these people to make their life easier. [This paragraph needs more research]

**Project Goals:**

[The following paragraph is a brain shit]

With this project, I aim to clarify what each type of application provide along with their limitations along with their performance capabilities along with technologies that best suit each of the feature in the applications. [Here insert the method that you are going to use performance test it]. I plan to develop same application using monolithic architecture and another use microservice architecture. One will be deployed in traditional style while other will using cloud native approach. These two application will be built using exact same programming languages as to remove the idea that performance boost might have been due the underlying programming language.

**Motivation:**

[The following paragraph is brain shit]

* Motivation seems to require bit more research.