**3 Tier Architecture:**

The three-tier architecture is an architecture that organizes an application into three layers such as client, application and data end. The main advantage of this scenario is scalability, reliability and security.

Here, I’m considering the application as TeamCity (A DevOps, CI tool) and describing the entire structure. TeamCity now supports High Availability concept in later versions, however when we are setting up (earlier in 2019’s) TeamCity doesn’t support High Availability, so we transformed into Active/Passive mode with a dedicated database.

**Architecture:**

Database

Client (Master Server) (Secondary Server) (Build Farms)

(User Requests) Active Server Passive Server Agents

Common shared directory

**Prerequisites to start with this setup:**

1. A dedicated app server 1 (Windows server) where we would be installing the TeamCity
2. A dedicated app server 2 (Windows server) where we would be installing the TeamCity
3. A dedicated DB server (Windows server) where we would be installing MS SQL
4. Now, TeamCity has been installed with the default port, as a major security best practice change the port number from default value to other value
5. Since the connection is not secure enable the TeamCity to be accessible only with https
6. Connect the app server 1 to DB server, so that the data will be stored in a database
7. Whenever the load is high and the app server 1 crashes, immediately start the TeamCity service in app server 2 by connecting it to DB server as a database
8. Since we have enabled the common shared directory between the two TeamCity instances, the data will be available though the app server1 crashes
9. Since the TeamCity may be consumed by many project teams, it is advisable to own the build machines dedicatedly by app teams
10. So, once the app team shares us the machines, we will install the build agent and registers with the Master Server