1. **CI**: Continuous Integration.
2. **CD**: Continuous Delivery/Deployment.  
   Sometimes called Delivery and sometimes Deployment but Delivery is better. We will discuss about it.
3. **What is Pipeline?**
   1. In a manufacturing industry, we’ve to create a sequence of operations and that sequence is called pipeline.
   2. So, pipeline is a process but automated process.
4. **What is CI/CD pipeline**?
   1. Unlike Git, Chef, Maven which are tools, it is a **methodology** for SDLC which **makes the process automated**.
   2. It is an automated process (a sequence of operations).
   3. The advantage of it is that if any mistake, it can be traced at earlier point.

Diagram

Description automatically generated

1. **Let’s see before CI, how we’re facing problems**:
   1. Each developer writes the complete code (50,000 LOC) then push the code onto VCS (Git).
   2. Then, tester integrates the code from each developer and test it.  
      It takes many days and meanwhile the developers start working on some other tasks.
   3. If any error comes, tester inform the corresponding developer which may take many days.  
      Now, after those many days, it is difficult for the developer to remember what he has done.  
      So, again developer has to sit to understand the code and fix it.
   4. **Conclusion**: More time consumption to produce a working product.
2. **Let’s see after CI, how the above problem was solved**:

**50 LOC by Developer 1**

**50 LOC by Developer 2**

**Build**

**Test**

**Deploy**

**Integration**

**Source Code Repo**

**CI Server**

**Feedback**

* 1. See, now developer is pushing each sprint (code for a small task).
  2. Immediately, code is pushed onto CI Server where it is integrated, built and tested.  
     If any bugs on any stage, the developers will be informed immediately after stopping the further process (build, test, deploy).
  3. So, now developer has to check a little piece of codes and immediately after he/she pushed as notification will come within few minutes.
  4. **Advantage**: Immediate feedback about the issue in the code and developer doesn’t need to spend much time as code is very small and immediately, he/she gets the feedback.
  5. **Continuous Integration = Continuous Push (Jatin) + Continuous Built + Continuous Test**
  6. A picture containing text, whiteboard

     Description automatically generated

1. Let’s understand Jenkins.

**Deploy**

**Plan**

**Jenkins**

**Monitor**

**Operate**

**Test**

**Build**

**Code**

* 1. Developers just push the code into Source Code Repo like **Github**.
  2. Then Jenkins is responsible for taking the code from Github to some **build tool** such as **Maven** to build the code.   
     Then Jenkins takes the code and do **testing** using **Junit or Selenium framework**.  
     Then Jenkins **deploy** the code and **operate** it using **Chef, Puppet**.   
     For **monitoring**, we have **Negios**.
  3. **Other popular CI Integration tools** alternative to Jenkins but they are paid
     1. **Bamboo**
     2. **Travis CI**
     3. **BuildBot**

1. D
   1. **Is Jenkins a tool:** No. It is a framework.
   2. **Written:** In Java.
   3. **Required Software**: JDK
   4. **Default Port**: 8080
   5. **Platform**: Windows, MacOS, Unix-like OS.
   6. **Why So famous**:
      1. Free
      2. Community Support
      3. Lots of plugins
   7. **Usage**: For CI
   8. **History**:Sun Microsystems created Hudson. Later on, when Oracle took over the Sun Microsystems, Oracle created two versions one Jenkins from Hudson and second Hudson itself where they made Jenkins open source whereas Hudson was paid version with support from Oracle.   
      But Hudson didn’t receive the popularity whereas Jenkins got famous as it was open source free and a lot of developers gave their contributions and so there as a lot of plug-ins, good documentations, strong community.

Text, whiteboard

Description automatically generated Graphical user interface, application

Description automatically generated

1. Text, letter

   Description automatically generated Diagram

   Description automatically generated  
   **Checkstyle** is a **development tool** to help programmers write Java code that adheres to a **coding standard.**
2. **Advantages of Jenkins**:  
   Unlike Chef, Puppet, Ansible which are just tools which Jenkins is a framework.  
   Like Chef, you have everything fixed, you can’t change anything but in Jenkins, you can make changes as per your requirements.  
   Text, letter

   Description automatically generated A picture containing text, whiteboard

   Description automatically generated