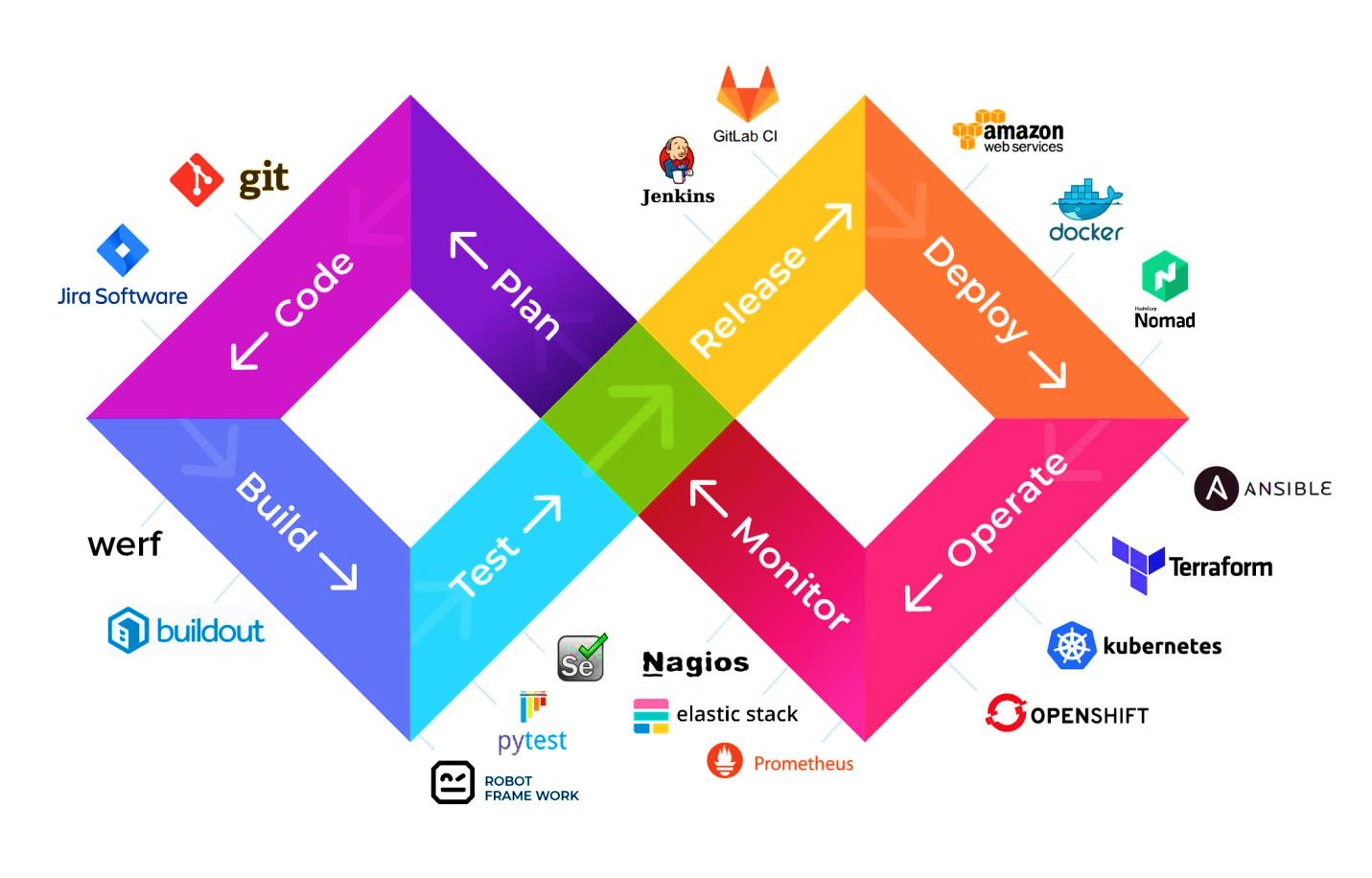
# This course focuses on the modern world agile practices for software development and delivery, we will cover implementation practices of tools in the following sequence:-

* Plan
* Code
* Build
* Test
* Release
* Deploy
* Operate



1. **Git & GitHub (VCS v DVCS):**  To be able to practice DevOps, need have all the communication for maintaining the code under VCS or DVCS for collaboration, so that teams could work on their specific codebase in isolation, It is used to track changes in the source code and allows multiple developers to work together. We cover the following :
   1. Initialize
   2. Add
   3. Commit
   4. Pull
   5. Push
   6. Branching
   7. Merging
   8. Rebasing.
2. **Maven/Graddle:** Developers used to spend huge time in setting up dependencies searching libraries over google and at the end of the day they have to manage build path using maven we can bypass the procedure. We specify every dependency in POM. XML i.e. Project Object Model and maven as a build tool resolves get dependency from its central repository and copy it into a local repo and add stated dependency to the build path, In Yiddish language the meaning of Maven is **"accumulator of knowledge"**. It is written in Java Language and used to build and manage projects written in C#, Ruby, Scala, and other languages
3. **JUNIT/PyTest :** In today’s modern software development practices it is important to test your code module in isolation, using Junit it provides the capability of testing the code written by developer without even merging the code.
4. **Selenium :**
5. **Jenkins:** is used for Continuous Integration purpose actually is an open source automation tool written in Java with plugins built for Continuous Integration. Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build.
6. **Docker**:
7. **Ansible** :
8. **Kubernetes:**  An open source system for deploying, scaling, and managing containerized applications, Kubernetes handles the work of scheduling containers onto a compute cluster and manages the workloads to ensure they run as the user intended. Instead of bolting on operations as an afterthought, Kubernetes brings software development and operations together by design. By using declarative, infrastructure-agnostic constructs to describe how applications ar e composed, how they interact, and how they are managed, Kubernetes enables an order-of-magnitude increase in operability of modern software systems.
9. **Nagios:**