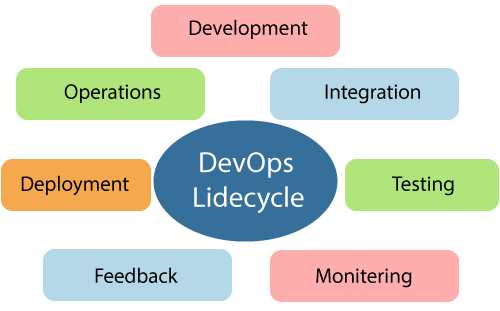


The **DevOps** is the combination of two words, one is Development and other is Operations. This allows a **single** **team** to handle the entire application lifecycle, from **development** to **testing, deployment**, and **operations**. DevOps helps you to reduce the disconnection between software developers, quality assurance (QA) engineers, and system administrators.

DevOps tools such as **Git, Ansible, Docker, Puppet, Jenkins, Chef, Nagios**, and **Kubernetes**.

**DevOps use Agile methodology to plan the development.**

**Dev ops life Cycles**

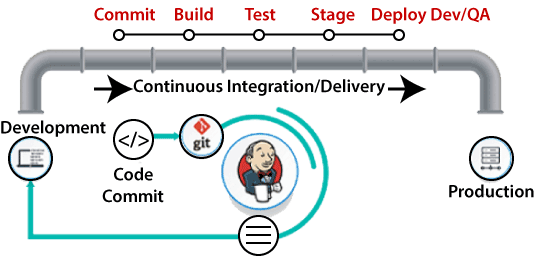


### **1) Continuous Development**

Planning and coding of the software. There are no DevOps tools that are required for planning, but there are several tools for maintaining the code.

### **2) Continuous Integration**

. The updated code needs to be integrated with the existing code.

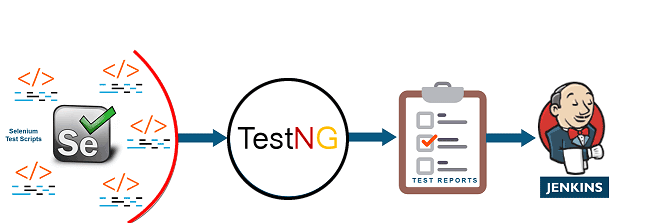


**Jenkins** is a popular tool used in this phase. Whenever there is a change in the Git repository, then Jenkins fetches the updated code and prepares a build of that code, which is an executable file in the form of **war or jar**.

Then this build is forwarded to the test server or the production server.

### **3) Continuous Testing**

 Developed software is continuously tested.  For constant testing, automation testing tools such as **TestNG, JUnit, Selenium**, etc are used. **Docker** Containers can be used for simulating the test environment.



**Selenium** does the automation testing, and TestNG generates the reports. This entire testing phase can automate with the help of a Continuous Integration tool called **Jenkins**.

### **4) Continuous Monitoring**

It maintains the security and availability of the service.

### **5) Continuous Feedback**

### **6) Continuous Deployment**

### **7) Continuous Operations**

**Dev ops life Cycles**: <https://www.javatpoint.com/devops-lifecycle>