Contents

[DevOps vs Agile 1](#_Toc522103972)

[Continuous Integration And Continuous Delivery: 2](#_Toc522103973)

[What is Continuous Integration (CI)? 2](#_Toc522103974)

[What is a Continuous delivery pipeline? 2](#_Toc522103975)

[What are the benefits of using Jenkins? 2](#_Toc522103976)

[What are the pre-requisites for using Jenkins? 3](#_Toc522103977)

[Explain how you can set up Jenkins job? 3](#_Toc522103978)

[Explain how to create a backup and copy files in Jenkins? 3](#_Toc522103979)

[What is the relation between Hudson and Jenkins? 3](#_Toc522103980)

[What are the two components Jenkins is mainly integrated with? 3](#_Toc522103981)

# DevOps vs Agile

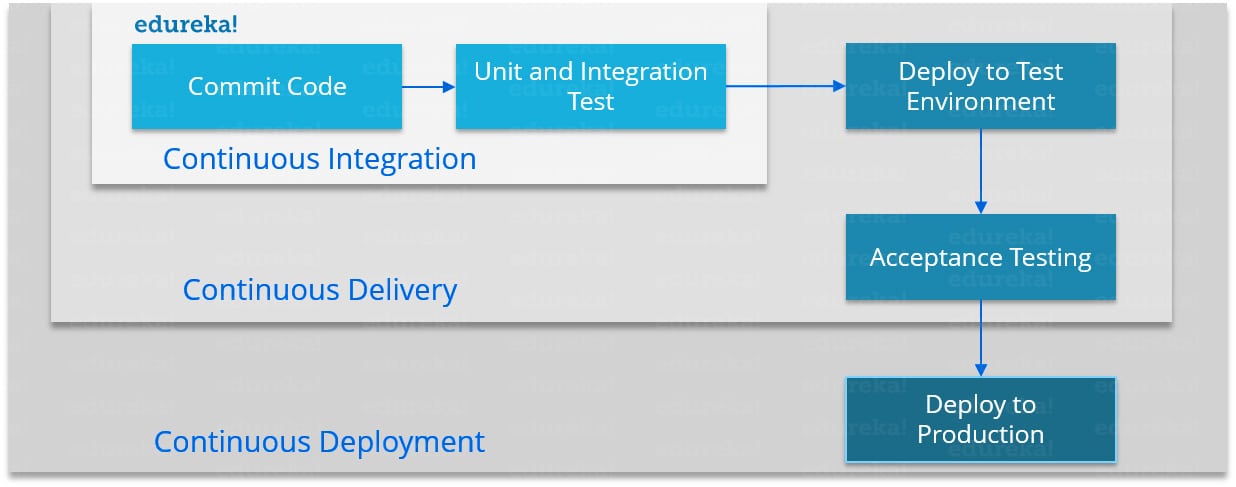
Agile’s founding principle is bringing Agility to Development. But, DevOps’ founding principle is bringing Agility to both Development and Operations.

|  |  |  |
| --- | --- | --- |
| **Features** | **DevOps** | **Agile** |
| **Agility** | Agility in both Development & Operations | Agility in only Development |
| **Processes/ Practices** | Involves processes such as CI, CD, CT, etc. | Involves practices such as Agile Scrum, Agile Kanban, etc. |
| **Key Focus Area** | Timeliness & quality have equal priority | Timeliness is the main priority |
| **Release Cycles/ Development Sprints** | Smaller release cycles with immediate feedback | Smaller release cycles |
| **Source of Feedback** | Feedback is from self (Monitoring tools) | Feedback is from customers |
| **Scope of Work** | Agility & need for Automation | Agility only |

# Continuous Integration and Continuous Delivery:

Knowledge on various tools is not enough, you should know where to use these tools.

These tools should be used in order to facilitate Continuous Integration and Continuous Delivery. Even Continuous Deployment in few cases, but Continuous Deployment is not considered as a good practice. Consider the below diagram to understand the difference.

****

# What is Continuous Integration (CI)?

A) Continuous integration (CI) is a software engineering practice in which isolated changes are immediately tested and reported on when they are added to a larger code base.

Continuous Integration (CI) is a development practice where developers integrate code into a shared repository frequently, preferably several times a day. Each integration can then be verified by an automated build and automated tests.

# What is a Continuous delivery pipeline?

A continuous delivery pipeline is an automated expression of your process for getting software from version control right through to your users and customers.

# What are the benefits of using Jenkins?

I will suggest you to include the following benefits of Jenkins, if you can recall any other benefit apart from the below mentioned points you can include that as well.

* At integration stage, build failures are cached.
* For each change in the source code an automatic build report notification is generated.
* To notify developers about build report success or failure, it is integrated with LDAP mail server.
* Achieves continuous integration agile development and test-driven development.
* With simple steps, maven release project is automated.
* Easy tracking of bugs at early stage in development environment than production.

# What are the pre-requisites for using Jenkins?

Answer to this is pretty straightforward to use Jenkins you require:

* A source code repository which is accessible, for instance, a Git repository.
* A working build script, e.g., a Maven script, checked into the repository.

# **Where did Jenkins store its data?**

Jenkins uses data directory to store all configuration and job information. **It doesn’t use any database**.

# **Explain how you can set up Jenkins job?**

My approach to this answer will be to first mention how to create Jenkins job.

Go to Jenkins top page, select “New Job”, then choose “Build a free-style software project”.

Now you can tell the elements of this freestyle job:

* Optional SCM, such as CVS or Subversion where your source code resides.
* Optional triggers to control when Jenkins will perform builds.
* Some sort of build script that performs the build (ant, maven, shell script, batch file, etc.) where the real work happens.
* Optional steps to collect information out of the build, such as archiving the artifacts and/or recording java doc and test results.
* Optional steps to notify other people/systems with the build result, such as sending e-mails, IMs, updating issue tracker, etc...

# Explain difference between Job & Build in Jenkins.

Job is the process that we define, it could be as simple as compiling, testing and packaging an application. Job can have many builds and a Build represents a result of executing a job.

# Where the details of Job stored in Jenkins?

Jenkins job details is stored in job configuration file which is basically an **xml file**.

# Explain how to create a backup and copy files in Jenkins?

Answer to this question is really direct.

To create a backup all you need to do is to periodically back up your JENKINS\_HOME directory. This contains all of your build jobs configurations, your slave node configurations, and your build history. To create a back-up of your Jenkins setup, just copy this directory. You can also copy a job directory to clone or replicate a job or rename the directory.

# What is the relation between Hudson and Jenkins?

You can just say Hudson was the earlier name and version of current Jenkins. After some issue, the project name was changed from Hudson to Jenkins.

# What are the two components Jenkins is mainly integrated with?

According to me Jenkins is mainly integrated with the following:

* Version Control system like GIT,SVN.
* Build tools like Apache Maven.