What is CI/CD? Continuous Integration & Continuous Delivery

What is Continuous Integration?

Continuous Integration is a software development method where team members integrate their work at least once a day. In this method, every integration is checked by an automated build to detect errors. This concept was first introduced over two decades ago to avoid "integration hell," which happens when integration is put off till the end of a project.

In Continuous Integration after a code commit, the software is built and tested immediately. In a large project with many developers, commits are made many times during a day. With each commit code is built and tested. If the test is passed, build is tested for deployment. If the deployment is a success, the code is pushed to Production. This commit, build, test, and deploy is a continuous process, and hence the name continuous integration/deployment.

In this CI tutorial, you will learn:

- What is Continuous Integration?
- Development without CI vs. Development with CI
- Difference between Compilation and Continuous Integration
- What do you need to conduct CI process?
- How Continuous integration work?
- Features of CI
- Why Use CI?
- Best practices of using CI
- Disadvantages of CI
- Tools for CI process

Development without CI vs. Development with CI

Here are key differences between development using CI or without CI.

Development without CI	Development with CI
Lots of Bugs	Fewer bugs

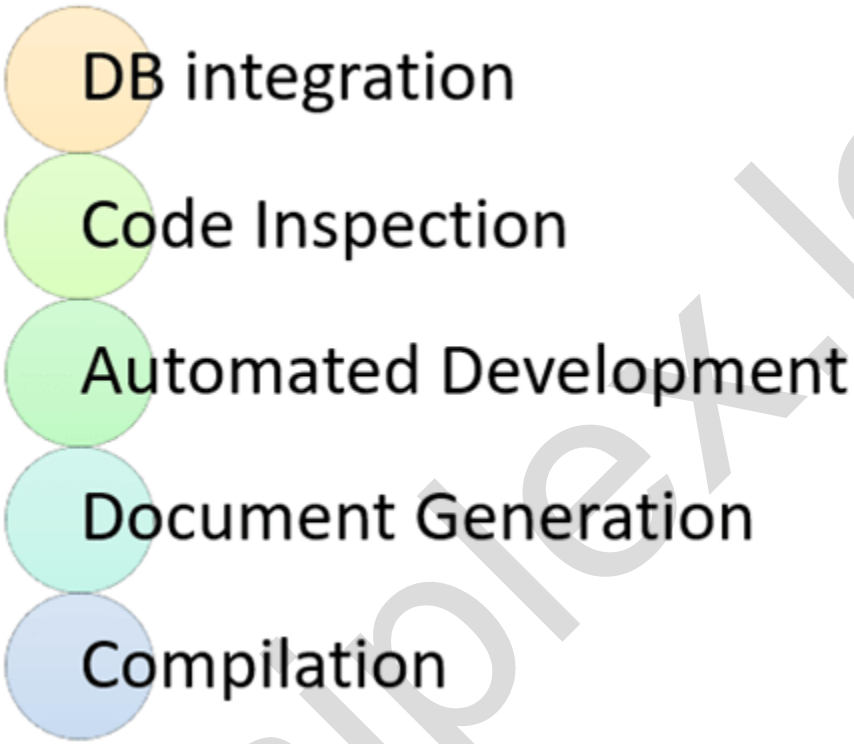
www.aiplex.lol

Infrequent commits	Regular commits
Infrequent and slow releases	Regular working releases
Difficult integration	Easy and Effective Integration
Testing happens late	Continuous Integration testing happens early and often.
Issue raised are harder to fix	Find and fix problems faster and more efficiently.
Poor project visibility	Better project visibility

www.aiplex.lol

Get more free courses at www.aiplex.lol

Difference between Compilation and Continuous Integration



Activities in Continuous Integration

While compilation only compiles a code, CI does the following activities

DB integration:

- Ensure DB and code in sync
- Automated creation of DB and test data.

Code Inspection:

- Ensures a healthy codebase
- Identifies problems early and applies best practices

Automated Deployment:

- Allows you to release product anytime
- Continually demo-able state and it is works on any machine

www.aiplex.lol

Get more free courses at www.aiplex.lol

Document generation:

- Ensure documentation is current
- Removes burned from the developer
- Produces build reports and metrics

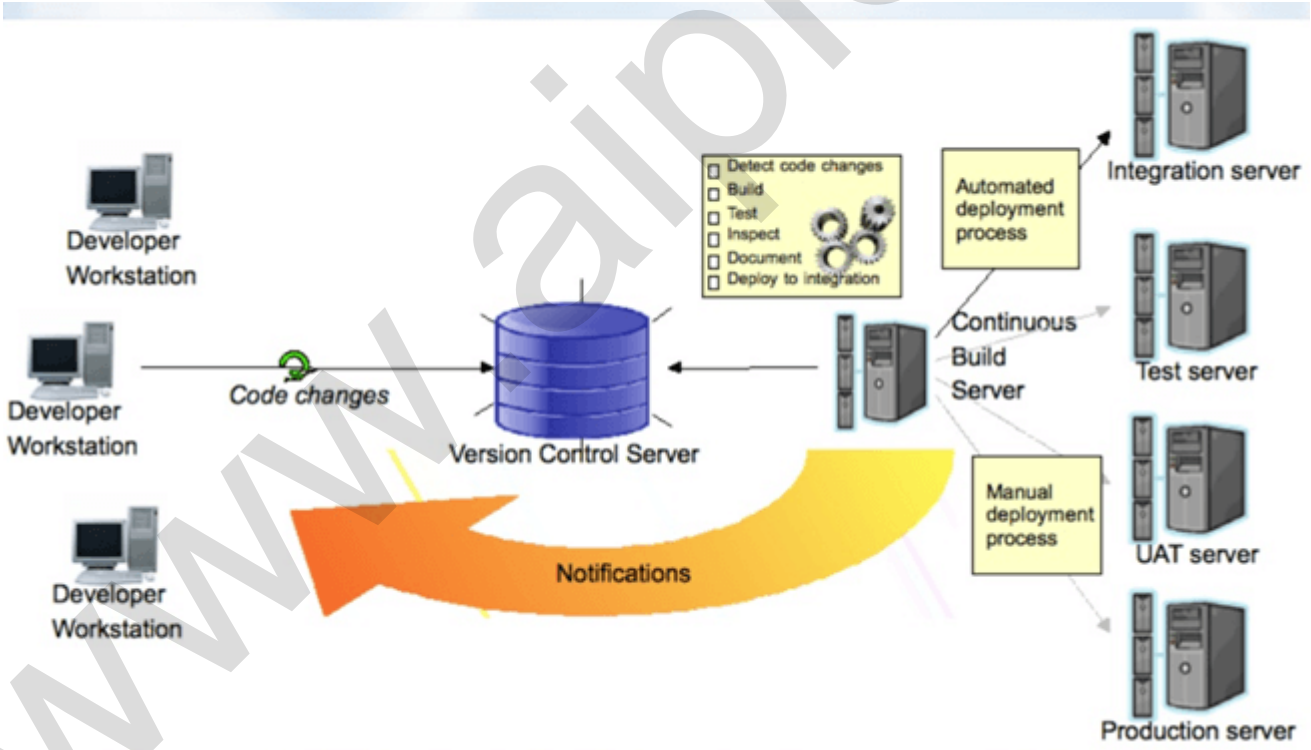
Compilation:

Compilation is the process the computer takes to convert a high-level programming language code into a machine language that the computer able to understand. It ensures a code compiler on every target platform.

When do I build?

- At every check-in
- Every time a dependency changes

How do I build?



CI process

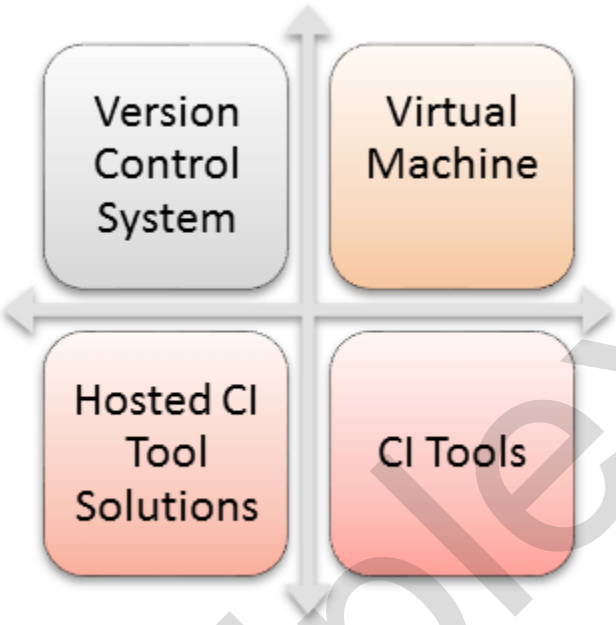
- Ideally, the build should come from the command line and should not depend on IDE.
- The build should happen continuously using a dedicated CI server, not a cron job.
- CI built should be triggered on every check-in and not just at midnight
- The build should provide immediate feedback and Require no developer effort

www.aiplex.lol

Get more free courses at www.aiplex.lol

- Identify key metrics and track them visually. More importantly, act on them immediately

What do you need to conduct CI process?



Here, are the key elements which you need to perfom the entire CI process:

- **Version Control System (VCS):** It offers a reliable method to centralize and preserve changes made to your project over time.
- **Virtual Machine:** You should have a spare server or at least one virtual machine to build your system.
- **Hosted CI Tool Solutions:** To avoid servers or virtual machines, you should go for hosted CI tool solutions. This tool helps in the maintenance of the whole process and offers easier scalability.
- **Tools:** If you select a self-hosted variant, you will need to install one of the many CI tools like Jenkins, TeamCity, Bamboo, GitLab, etc.

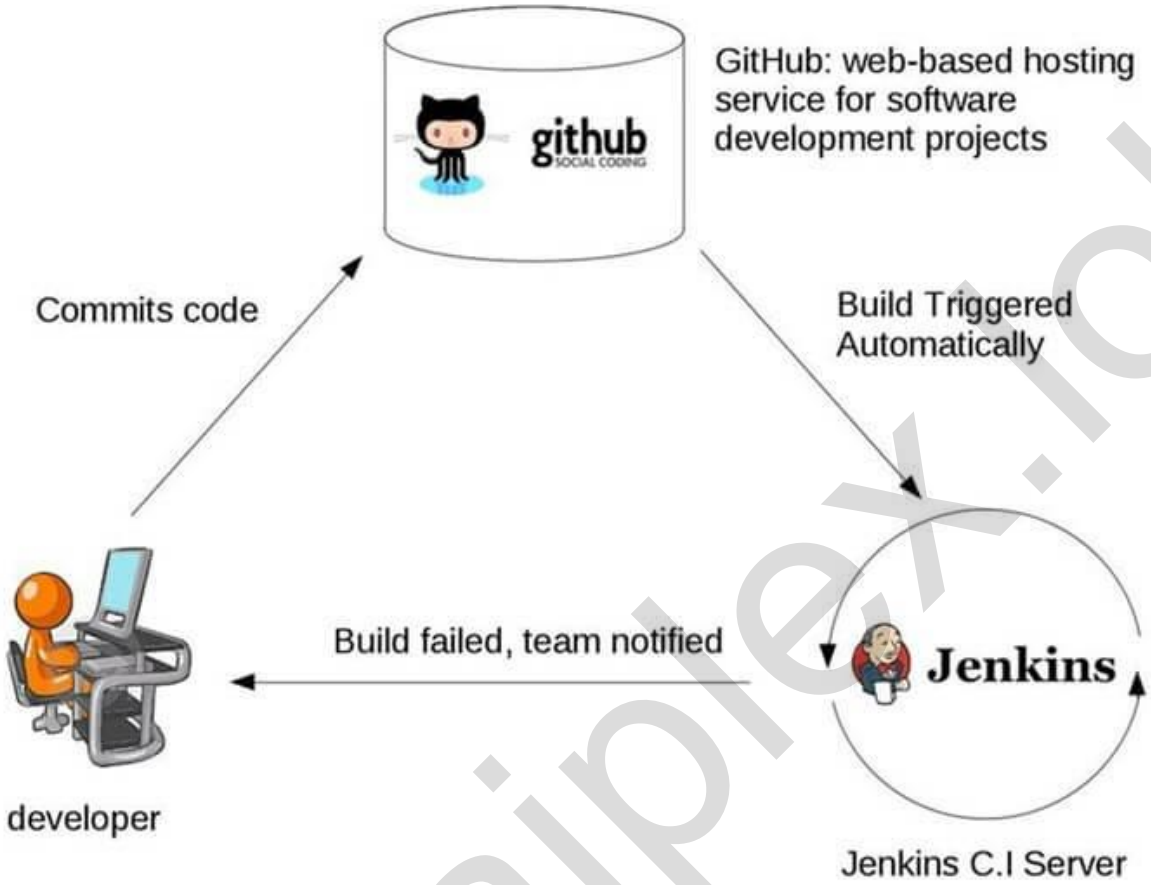
How Continuous integration work?

You are surely aware of the old phone Nokia. Nokia used to implement a procedure called nightly build. After multiple commits from diverse developers during the day, the software built every night. Since the software was built only once in a day, it's a huge pain to isolate, identify, and fix the errors in a large codebase.

Later, they adopted the Continuous Integration approach. The software was built and tested as soon as a developer committed code. If any error is detected, the respective developer can quickly fix the defect.

www.aiplex.lol

Get more free courses at www.aiplex.lol



Example of Continuous Integration

Features of CI

Here, are important features and benefits of Continuous Integration:

- Allows you to maintain just a single source repository
- You can test the clone of the production CI environment
- The built environment should be close to the production environment.
- One of the advantages of continuous integration is Constant availability of a current build
- The complete process of build and testing and deployment should be visible to all the stack holders.

Why Use CI?

Here are important reasons for using Continuous Integration:

- Helps you to build better quality software
- CI process helps to scale up headcount and delivery output of engineering teams.
- CI allows software developers to work independently on features in parallel.

www.aiplex.lol

Get more free courses at www.aiplex.lol

- Helps you to conduct repeatable testing
- Increase visibility enabling greater communication
- Helps develop a potentially shippable product for fully automated build
- Helps you to reduced risks by making deployment faster and more predictable
- immediate feedback when issue arrives
- Avoid last-minute confusion at release date and timing

Best practices of using CI Systems

Here, are some important best practices while implementing

- Commit Early and Commit Often never Commit Broken Code
- Fix build failures immediately
- Act on metrics
- Build-in every target environment Create artifacts from every build
- The build of the software need to be carried out in a manner so that it can be automated
- Do not depend on an IDE
- Build and test everything when it changes
- The database schema counts as everything
- Helps you to find out key metrics and track them visually
- Check-in often and early
- Stronger source code control
- Continuous integration is running unit tests whenever you commit code
- Automate the build and test everyone
- Keep the build fast with automated deployment

Disadvantages of CI

Here, are cons/drawbacks of Continuous Integration process:

- Initial setup time and training is required to get acquainted with Cl server
- Development of suitable test procedures is essential
- Well-developed test-suite required many resources for Cl server
- Conversion of familiar processes
- Requires additional servers and environments
- Waiting times may occur when multiple developers want to integrate their code around the same time

Tools for CI process

Here, are some most essential CI/CD tools:

www.aiplex.lol

Get more free courses at www.aiplex.lol

Jenkins:

Jenkins is an open-source continuous integration software. It is written using the Java programming language. It facilitates real-time testing and reporting on isolated changes in a more massive codebase. This software helps developers to quickly find and solve defects in their codebase & automate testing of their builds.

Bamboo:

Bamboo is a continuous integration build server that performs - automatic build, test, and releases in a single place. It works seamlessly with JIRA software and Bitbucket. Bamboo supports many languages and technologies such as CodeDeply, Ducker, Git, SVN, Mercurial, AWS, and Amazon S3 buckets.

TeamCity:

TeamCity is a Continuous Integration server that supports many powerful features. It maintains a CI server healthy and stable even when no builds are running. It provides better code quality for any project

Summary:

- Continuous Integration definition: Continuous integration is a software development method where members of the team can integrate their work at least once a day
- CI/CD meaning combination of Continuous Integration and Continuous Delivery or Continuous Deployment.
- Development without CI creates lots of bugs whereas Development with CI offers Fewer bugs

www.aiplex.lol

Get more free courses at www.aiplex.lol

- Important activities of Continous Integration are 1) DB integration, 2) Code Inspection, 3) Automated Deployment, Document generation, and Compilation.
- The build should happen continuously using a dedicated Cl server, not a cron job.
- Important elements of CI are 1) Version Control System 2) Virtual Machine 3) Host CI Tool solutions 4) Tools
- Continuous Integration system allows you to maintain just a single source repository
- CI/CD process helps you to build better quality software
- The most important best practices of Azure Continuous Integration process is to Commit Early and Commit Often never Commit Broken Code
- The major drawback of the CICD pipeline process is that well-developed test-suite required many resources for Cl server
- Jenkins, Bambook, and Team City are some useful AWS Continuous Integration tools.

www.aiplex.lol

Get more free courses at www.aiplex.lol

What is Jenkins? Why Use Continuous Integration (CI) Tool?

What is Jenkins?

Jenkins is an open-source Continuous Integration server written in Java for orchestrating a chain of actions to achieve the Continuous Integration process in an automated fashion. Jenkins supports the complete development life cycle of software from building, testing, documenting the software, deploying, and other stages of the software development life cycle.

Jenkins is a widely used application around the world that has around 300k installations and growing day by day. By using Jenkins, software companies can accelerate their software development process, as Jenkins can automate build and test at a rapid rate.

It is a server-based application and requires a web server like Apache Tomcat. The reason Jenkins software became so popular is that of its monitoring of repeated tasks which arise during the development of a project. For example, if your team is developing a project, Jenkins will continuously test your project builds and show you the errors in early stages of your development.

In this tutorial, you will learn

- What is Jenkins?
- What is Continuous Integration?
- Jenkin History
- Why use Continuous Integration with Jenkins?
- Real-world case study of Continuous Integration
- Advantages of using Jenkins
- Disadvantages of using Jenkins

What is Continuous Integration?

Continuous Integration is a process of integrating code changes from multiple developers in a single project many times. The software is tested immediately after a code commit. With each code commit, code is built and tested. If the test is passed, the build is tested for deployment. If the deployment is successful, the code is pushed to production.

This commit, build, test, and deploy is a continuous process and hence the name continuous integration/deployment.

www.aiplex.lol

Get more free courses at www.aiplex.lol

How does Jenkins work?

Jenkins is a server-based application and requires a web server like Apache Tomcat to run on various platforms like Windows, Linux, macOS, Unix, etc. To use Jenkins, you need to create pipelines which are a series of steps that a Jenkins server will take. Jenkins Continuous Integration Pipeline is a powerful instrument that consists of a set of tools designed to **host**, **monitor**, **compile** and **test** code, or code changes, like:

- **Continuous Integration Server** (Jenkins, Bamboo, CruiseControl, TeamCity, and others)
- **Source Control Tool** (e.g., CVS, SVN, GIT, Mercurial, Perforce, ClearCase and others)
- **Build tool** (Make, ANT, Maven, Ivy, Gradle, and others)
- **Automation testing framework** (Selenium, Appium, TestComplete, UFT, and others)

Jenkin History

- Kohsuke Kawaguchi, a Java developer, working at SUN Microsystems, was tired of building the code and fixing errors repetitively. In 2004, created an automation server called Hudson that automates build and test task.
- In 2011, Oracle who owned Sun Microsystems had a dispute with Hudson open source community, so they forked Hudson and renamed it as Jenkins.
- Both Hudson and Jenkins continued to operate independently. But in short span of time, Jenkins acquired a lot of projects and contributors while Hudson remained with only 32 projects. With time, Jenkins became more popular, and Hudson is not maintained anymore.

Why use Continuous Integration with Jenkins?

Some people might think that the old-fashioned way of developing the software is the better way. Let's understand the advantages of CI with Jenkins with the following example

Let us imagine, that there are around 10 developers who are working on a shared repository. Some developer completes their task in 25 days while others take 30 days to complete.

Before Jenkins	After Jenkins
Once all Developers had completed their assigned coding tasks, they used to commit their code all at same time. Later, Build is tested and deployed.	The code is built and test as soon as Developer commits code. Jenkin will build and test code many times during the day If the build is successful,

www.aiplex.lol

Get more free courses at www.aiplex.lol

Code commit built, and test cycle was very infrequent, and a single build was done after many days.	then Jenkins will deploy the source into the test server and notifies the deployment team. If the build fails, then Jenkins will notify the errors to the developer team.
Since the code was built all at once, some developers would need to wait until other developers finish coding to check their build	The code is built immediately after any of the Developer commits.
It is not an easy task to isolate, detect, and fix errors for multiple commits.	Since the code is built after each commit of a single developer, it's easy to detect whose code caused the built to fail
Code build and test process are entirely manual, so there are a lot of chances for failure.	Automated build and test process saving timing and reducing defects.
The code is deployed once all the errors are fixed and tested.	The code is deployed after every successful build and test.
Development Cycle is slow	The development cycle is fast. New features are more readily available

www.aiplex.lol

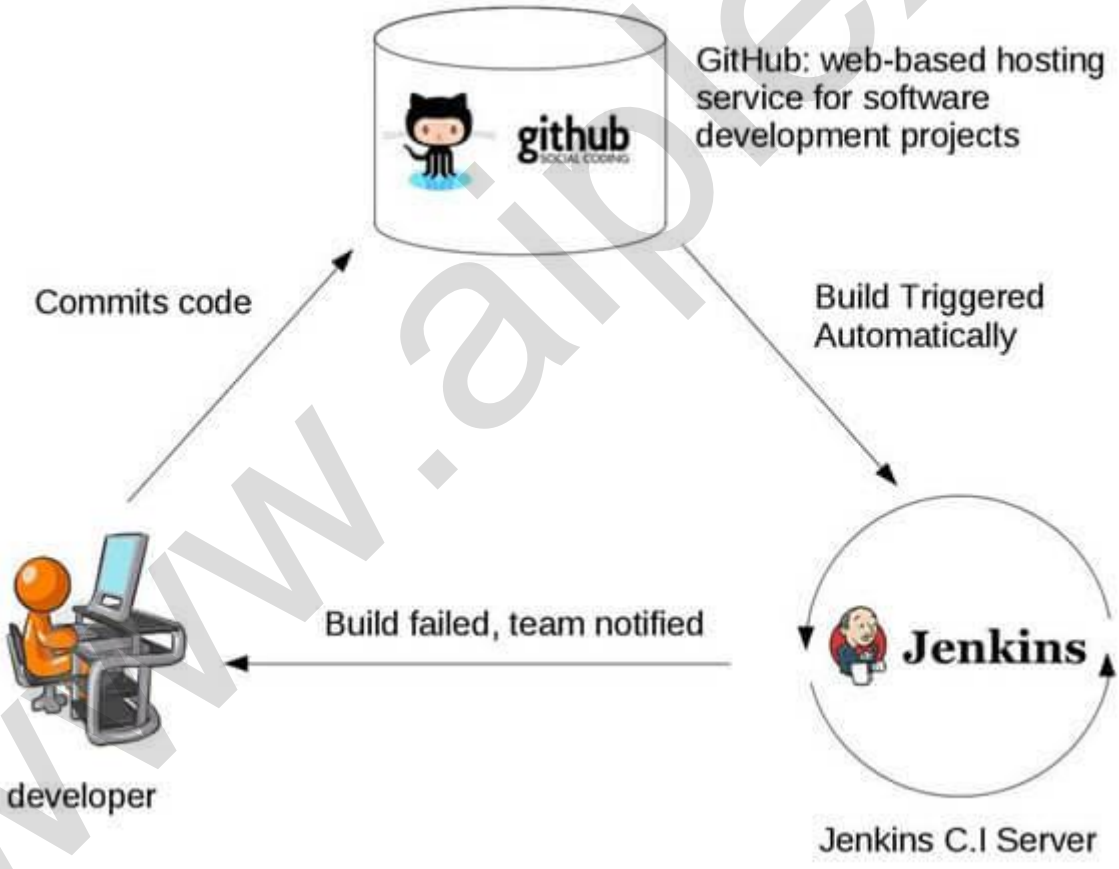
Get more free courses at www.aiplex.lol

to users. Increases profits.

Real-world case study of Continuous Integration

I am sure all of you aware of old phone Nokia. Nokia used to implement a procedure called nightly build. After multiple commits from diverse developers during the day, the software built every night. Since the software was built only once in a day, it's a huge pain to isolate, identify, and fix the errors in a large code base.

Later, they adopted Continuous Integration approach. The software was built and tested as soon as a developer committed code. If any error is detected, the respective developer can quickly fix the defect.



Jenkins Plugins

By default, Jenkins comes with a limited set of features. If you want to integrate your Jenkins installation with version control tools like Git, then you need to install plugins related to Git. In

www.aiplex.lol

Get more free courses at www.aiplex.lol

fact, for integration with tools like Maven, Amazon EC2, you need to install respective plugins in your Jenkins.

Plugins integration in Jenkins

Advantages of using Jenkins

- Jenkins is being managed by the community which is very open. Every month, they hold public meetings and take inputs from the public for the development of Jenkins project.
- So far around 280 tickets are closed, and the project publishes stable release every three months.
- As technology grows, so does Jenkins. So far Jenkins has around 320 plugins published in its plugins database. With plugins, Jenkins becomes even more powerful and feature rich.
- Jenkins tool also supports cloud-based architecture so that you can deploy Jenkins in cloud-based platforms.
- The reason why Jenkins became popular is that it was created by a developer for developers.

Disadvantages of using Jenkins

Though Jenkins is a very powerful tool, it has its flaws.

- Its interface is out dated and not user friendly compared to current UI trends.
- Though Jenkins is loved by many developers, it's not that easy to maintain it because Jenkins runs on a server and requires some skills as server administrator to monitor its activity.

www.aiplex.lol

Get more free courses at www.aiplex.lol

- One of the reasons why many people don't implement Jenkins is due to its difficulty in installing and configuring Jenkins.
- Continuous integrations regularly break due to some small setting changes. Continuous integration will be paused and therefore requires some developer attention.

Conclusion:

- In Continuous Integration, after a code commit, the software is built and tested immediately
- Jenkins used for orchestrating a chain of actions for Continuous Integration in a software project
- Before Jenkins when all Developers had completed their assigned coding tasks, they used to commit their code all at same time. Later, Build is tested and deployed.
- After Jenkins the code is built and test as soon as Developer commits code. Jenkin will build and test code many times during the day
- By default, Jenkins comes with a limited set of features. If you want to integrate your Jenkins installation with version control tools like Git, then you need to install plugins related to Git
- The biggest pros of Jenkins is that it is managed by the community which holds public meetings and take inputs from the public for the development of Jenkins projects
- The biggest con of Jenkin is that Its interface is out dated and not user friendly compared to current UI trends.

www.aiplex.lol

Get more free courses at www.aiplex.lol

How to Download & Install Jenkins on Windows

Jenkins may be installed on either Windows or Unix platforms, but we will focus on Windows installation only.

Prerequisites:

Before you proceed to install Jenkins in your windows system, there are some prerequisites for Jenkins to install Jenkins in your computer.

Hardware requirements:

- You need minimum 256 MB of RAM in your computer or laptop to install Jenkins
- You need at least 1 GB of space in your hard drive for Jenkins.

Software Requirements:

- Since Jenkins runs on Java, you need either latest version of Java Development Kit (JDK) or Java Runtime Environment (JRE).

Release Types

Jenkins releases two types of versions based on the organization needs.

- Long-term support release
- Weekly release

Long term support release (LTS) :

Long-term support releases are available every 12 weeks. They are stable and are widely tested. This release is intended for end users.

Weekly release:

Weekly releases are made available every week by fixing bugs in its earlier version. These releases are intended towards plugin developers.

We will use the LTS release though the process remains the same for Weekly release.

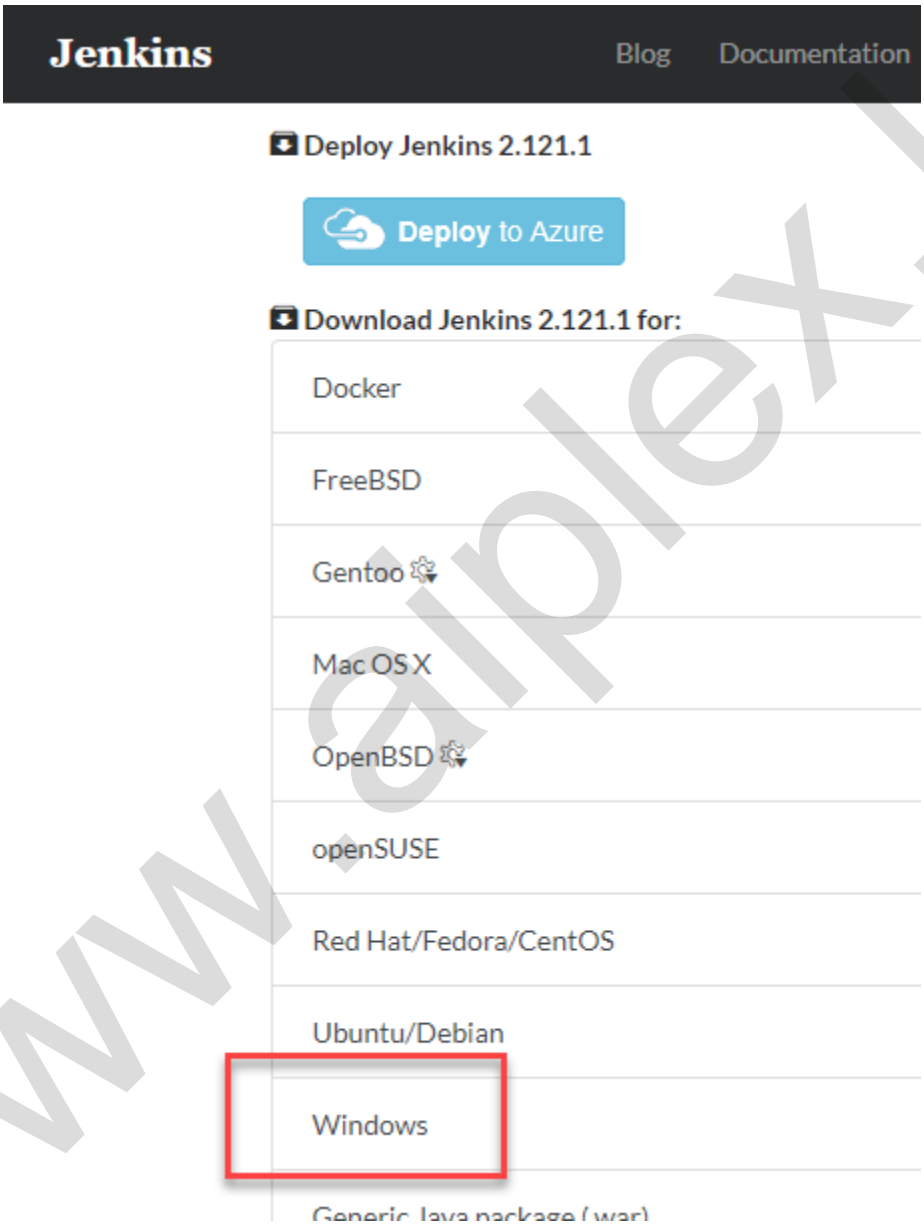
www.aiplex.lol

Get more free courses at www.aiplex.lol

How to Download Jenkins?

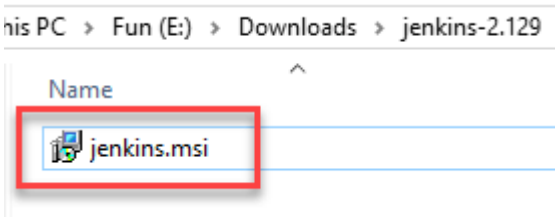
Following steps should be followed so that to install Jenkins successfully:

Step 1) Got to https://www.jenkins.io/download/ and select the platform. In our case Windows

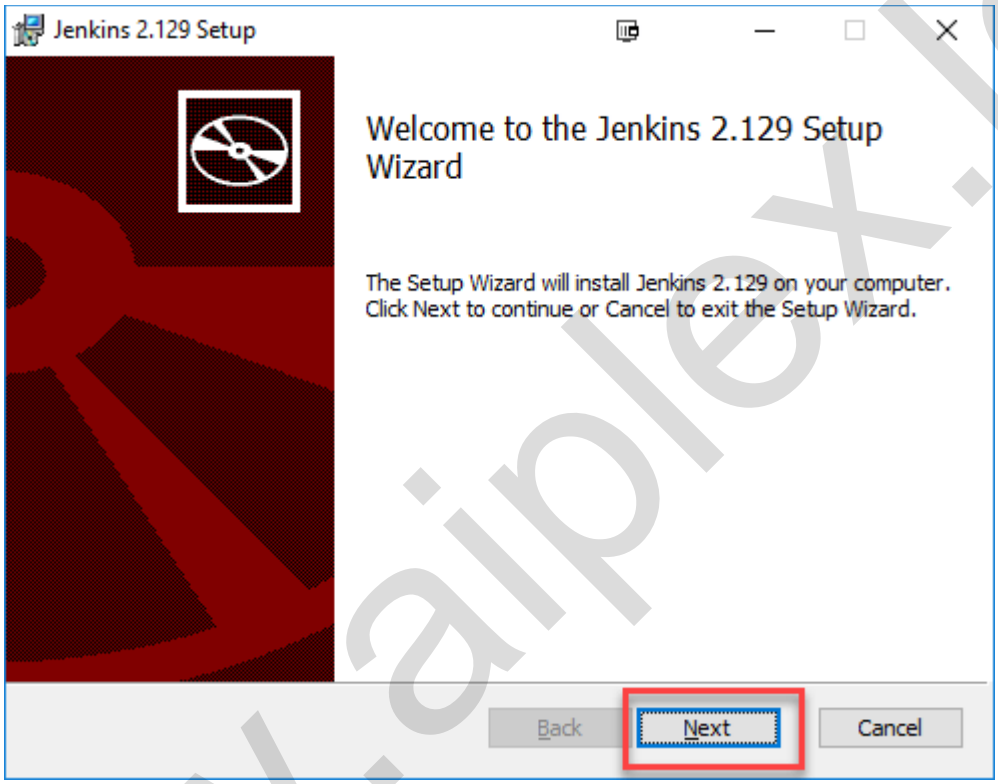


Step 2) Go to download location from local computer and unzip the downloaded package. Double-click on unzipped **jenkins.msi**. You can also Jenkin using a WAR (Web application ARchive) but that is not recommended.

www.aiplex.lol

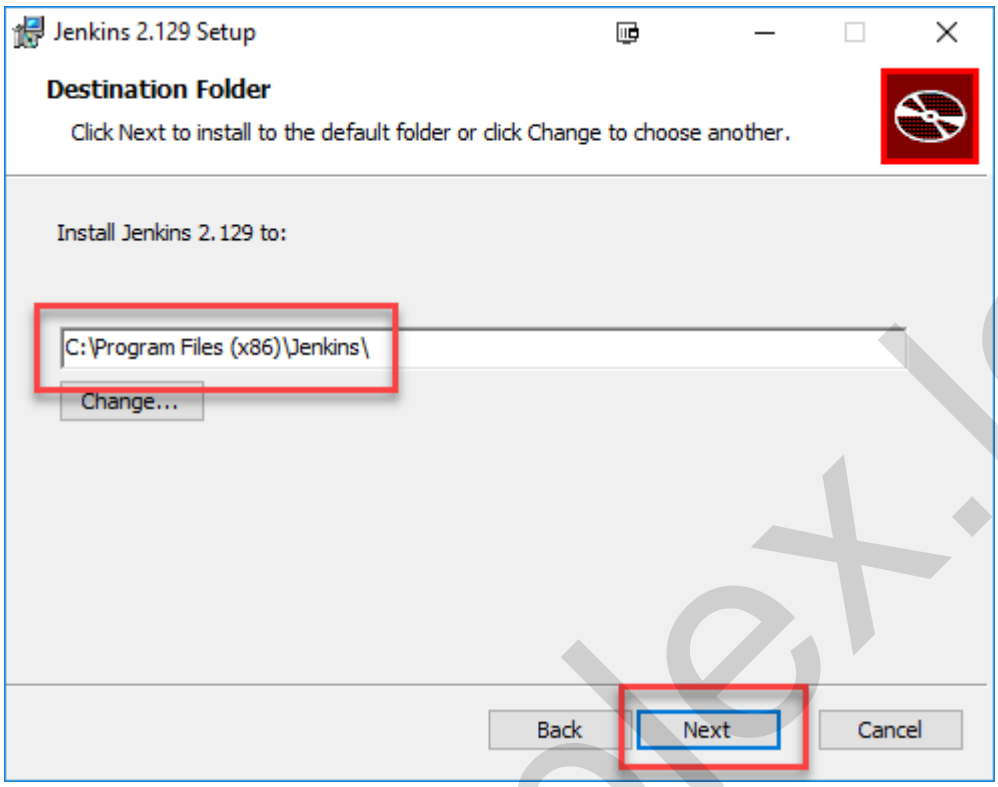


Step 3) In the Jenkin Setup screen, click Next.

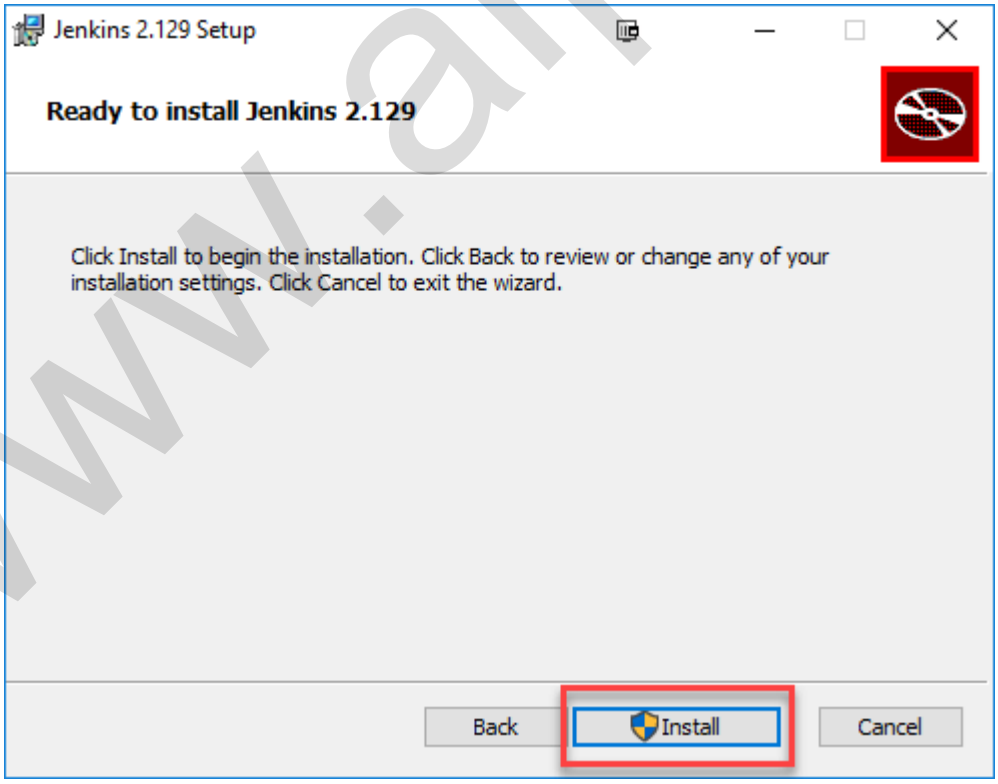


Step 4) Choose the location where you want to have the Jenkins instance installed (default location is C:\Program Files (x86)\Jenkins), then click on **Next** button.

www.aiplex.lol

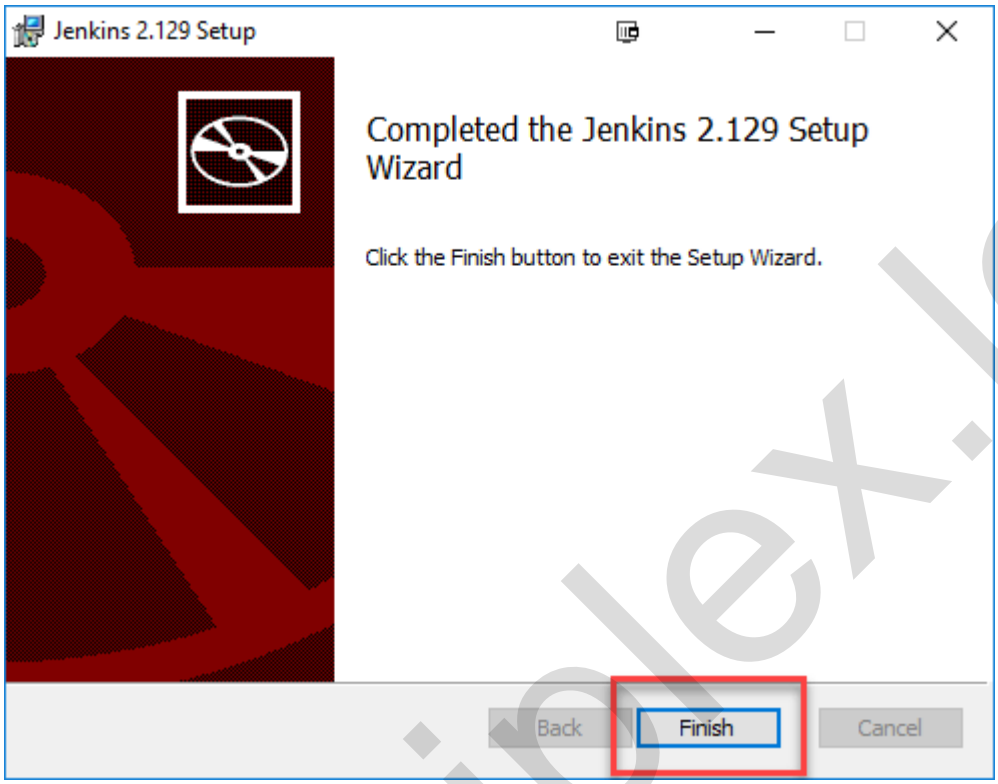


Step 5) Click on the Install button.

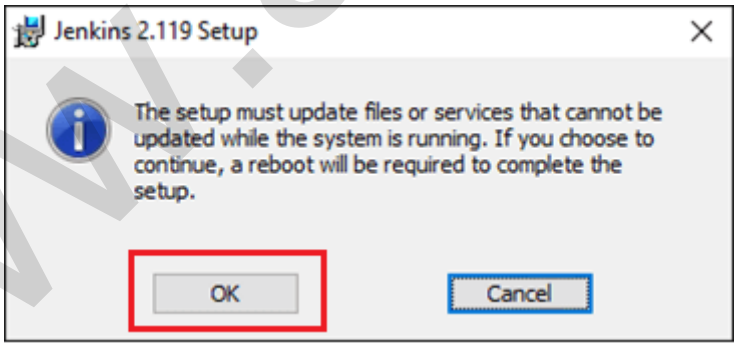


www.aiplex.lol

Step 6) Once install is complete, click Finish.



Step 7) During the installation process an info panel may pop-up to inform the user that for a complete setup, the system should be rebooted at the end of the current installation. Click on OK button when the Info panel is popping-up:



How to Unblock Jenkins?

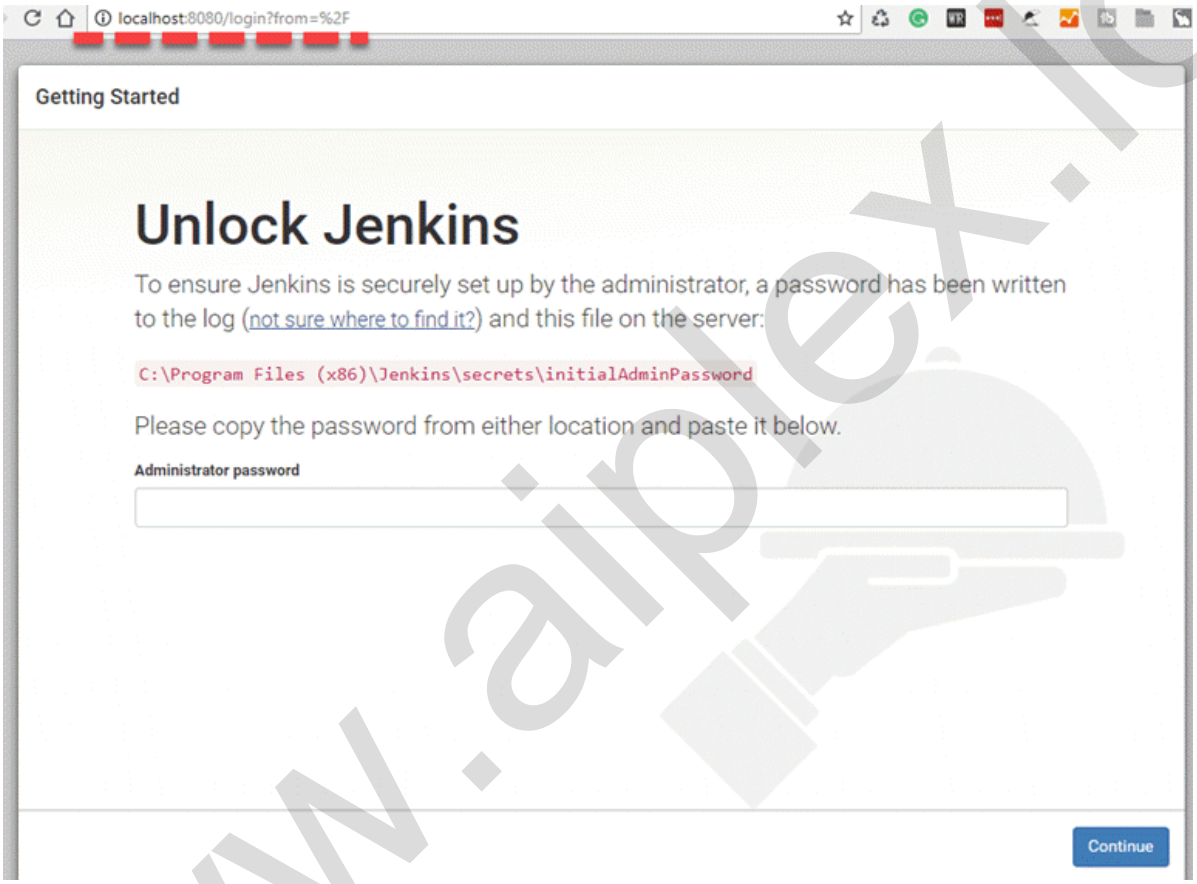
After completing the Jenkins installation phase, you should proceed further and start its configuration. Next steps will guide you how you can unblock Jenkins application:

www.aiplex.lol

Step 1) After completing the Jenkins installation process, a browser tab will pop-up asking for the initial Administrator password. To access Jenkins, you need to go to browse the following path in your web browser.

http://localhost:8080

If you can access the above URL, then it confirms that Jenkins is successfully installed in your system.



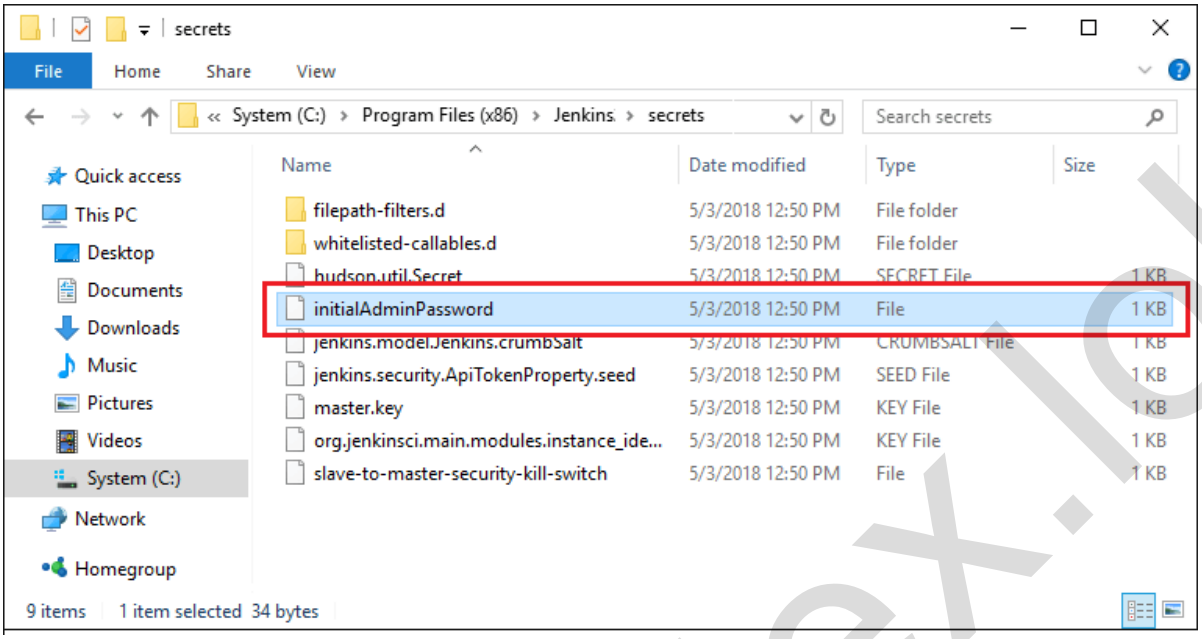
Step 2) The initial Administrator password should be found under the Jenkins installation path (set at Step 4 in Jenkins Installation).

For default installation location to C:\Program Files (x86)\Jenkins, a file called **initialAdminPassword** can be found under C:\Program Files (x86)\Jenkins\secrets.

However, If a custom path for Jenkins installation was selected, then you should check that location for **initialAdminPassword** file.

www.aiplex.lol

Get more free courses at www.aiplex.lol



Step 3) Open the highlighted file and copy the content of the **initialAdminPassword** file.

Step 4) Paste the password it into browser's pop-up tab (http://localhost:8080/login?form=%2F) and click on Continue button.

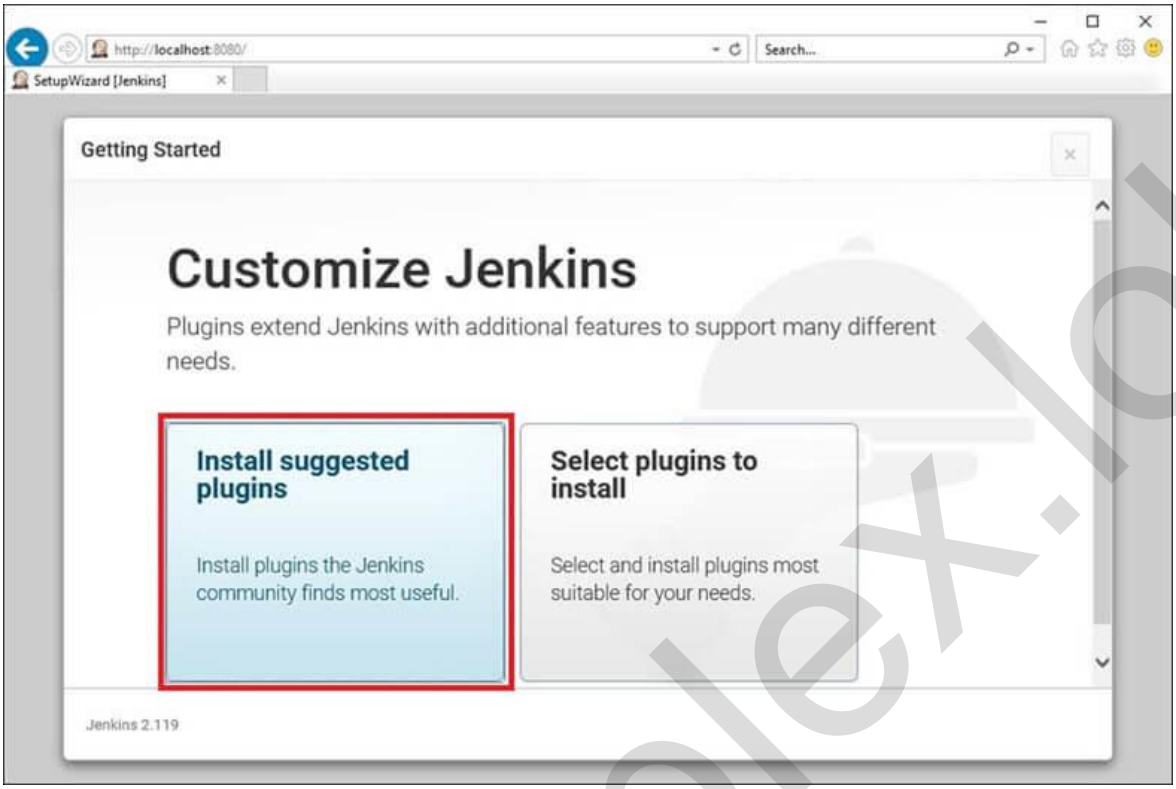
www.aiplex.lol

Customize Jenkins

You can also customize your Jenkins environment by below-given steps:

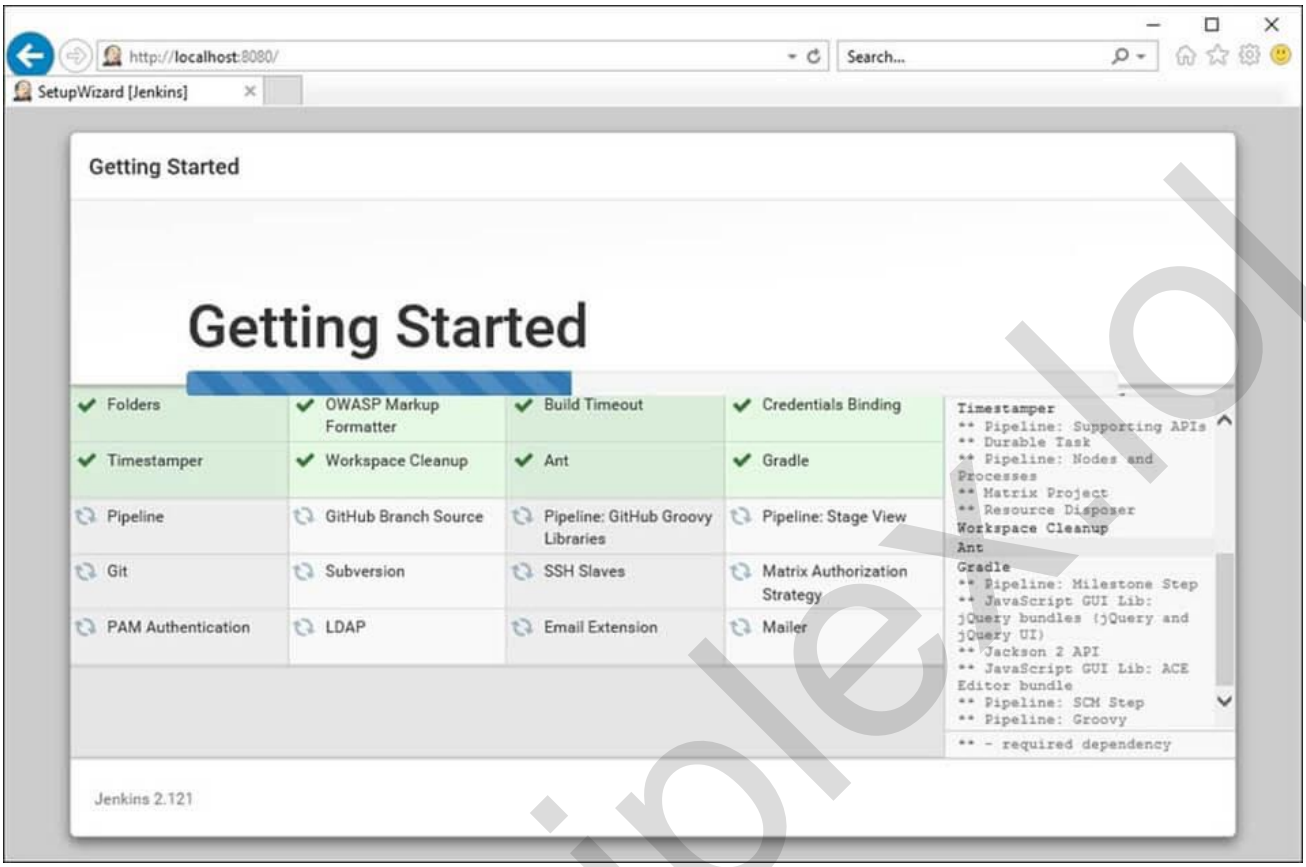
Step 1) Click on the "Install suggested plugins button" so Jenkins will retrieve and install the essential plugins

www.aiplex.lol

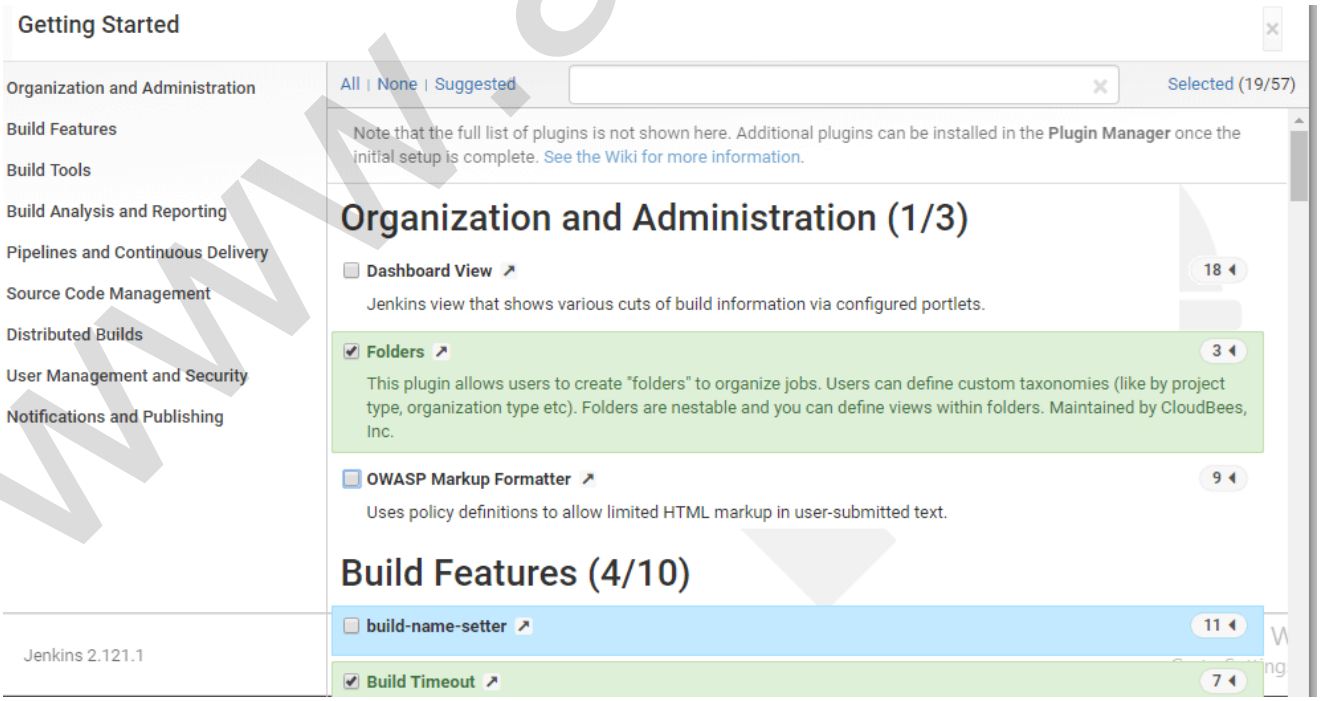


Jenkins will start to download and install all the necessary plugins needed to create new Jenkins Jobs.

www.aiplex.lol



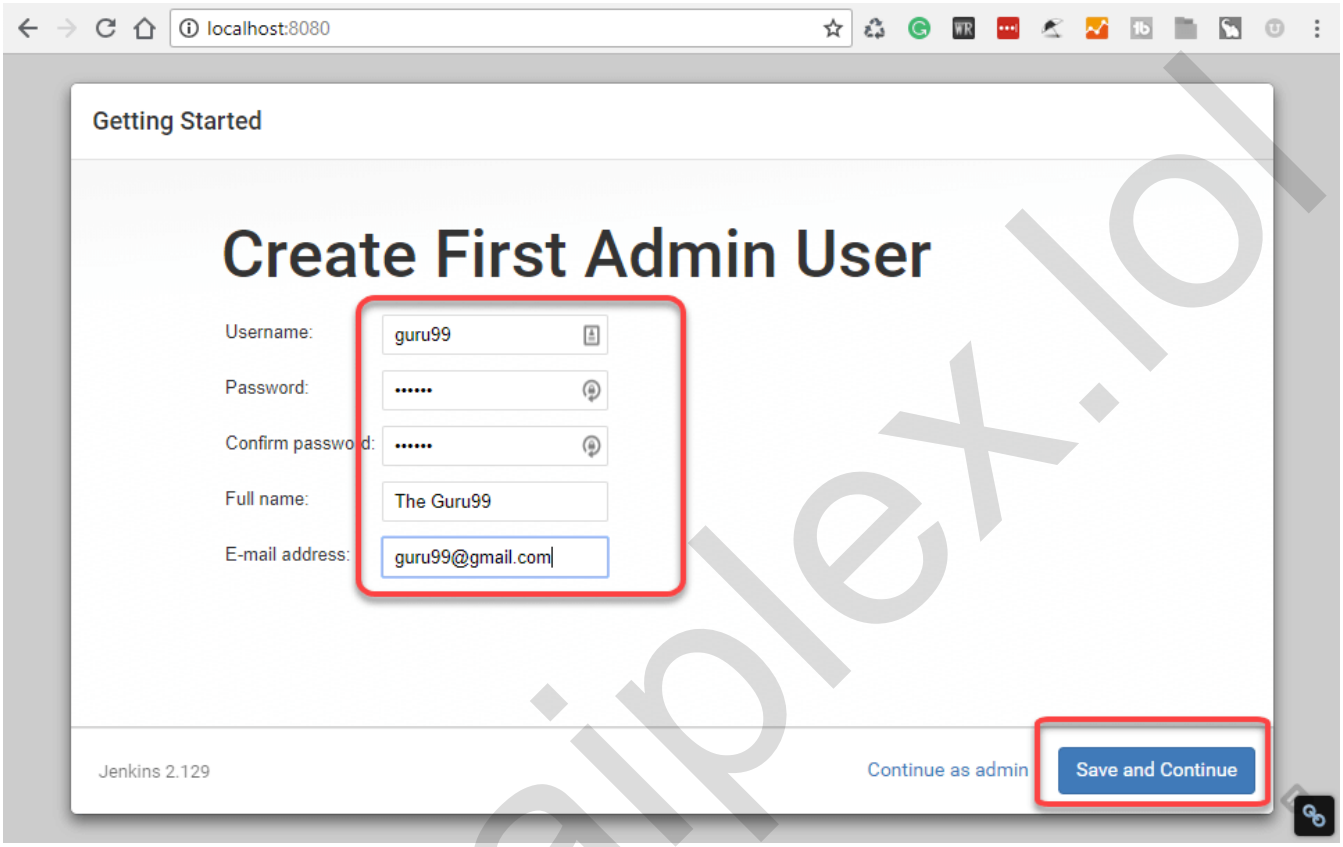
Note: You can choose the Option "Select Plugins to Install" and select the plugins you want to install



www.aiplex.lol

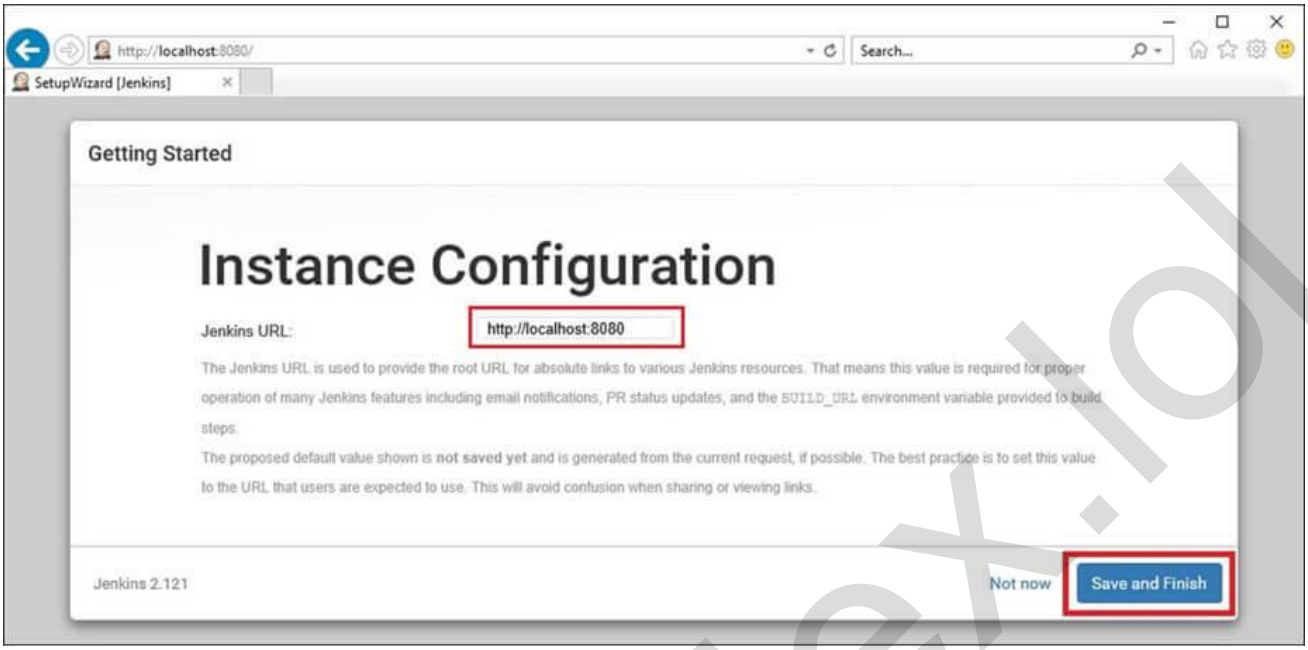
Get more free courses at www.aiplex.lol

Step 2) After all suggested plugins were installed, the "Create First Admin User" panel will show up. Fill all the fields with desired account details and hit the "**Save and Finish**" button.



Step 3) Once you have filled the above data, finally it will ask for URL information where you can configure the default instance path for Jenkins. Leave it as it is to avoid any confusions later. However, if another application is already using 8080 port, you can use another port for Jenkins and finally save the settings, and you are done with installation of Jenkins. Hit the "**Save and Continue**" button:

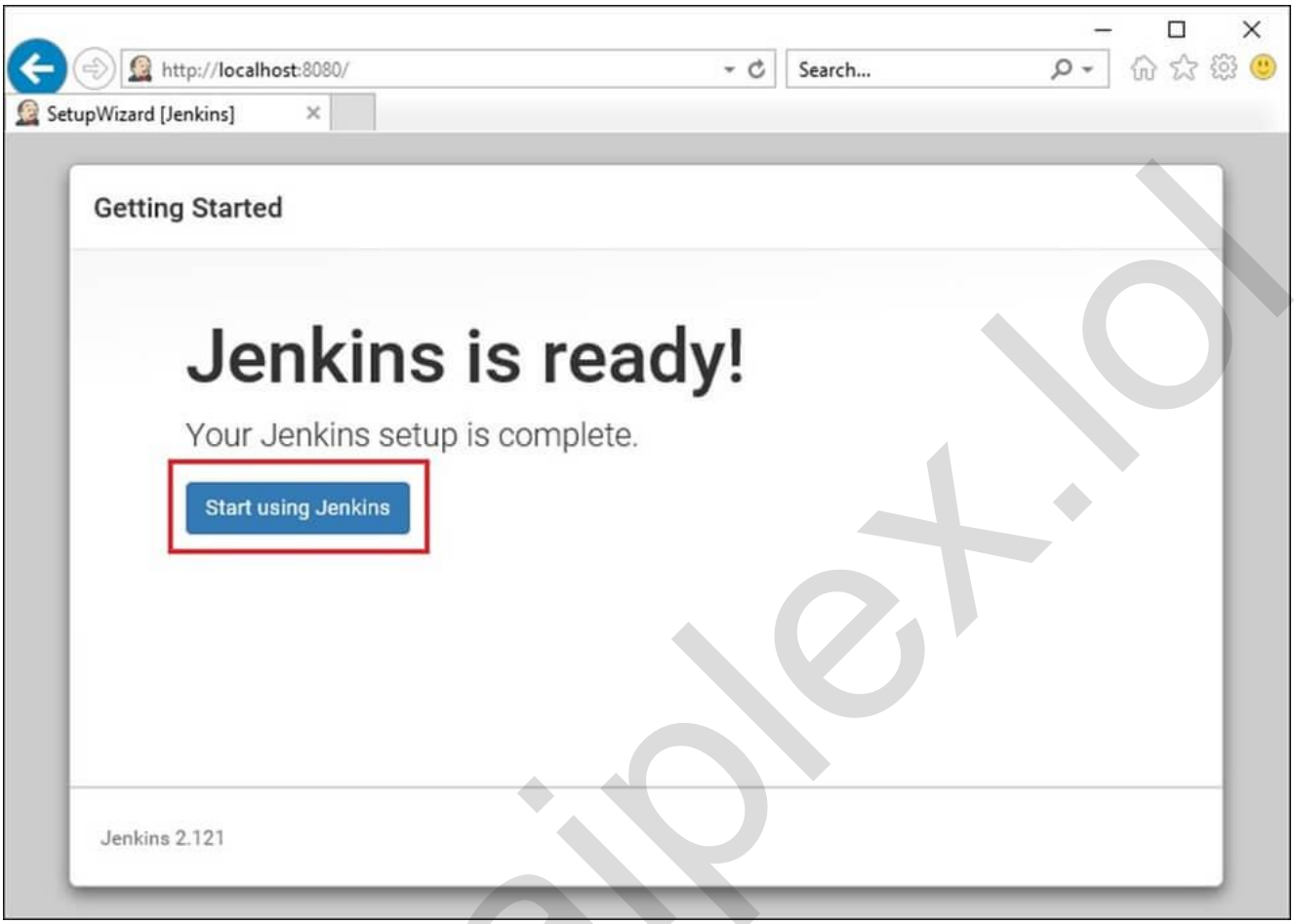
www.aiplex.lol



Congratulations! We have successfully installed a new Jenkins Server. Hit the "Start using Jenkins" button.

www.aiplex.lol

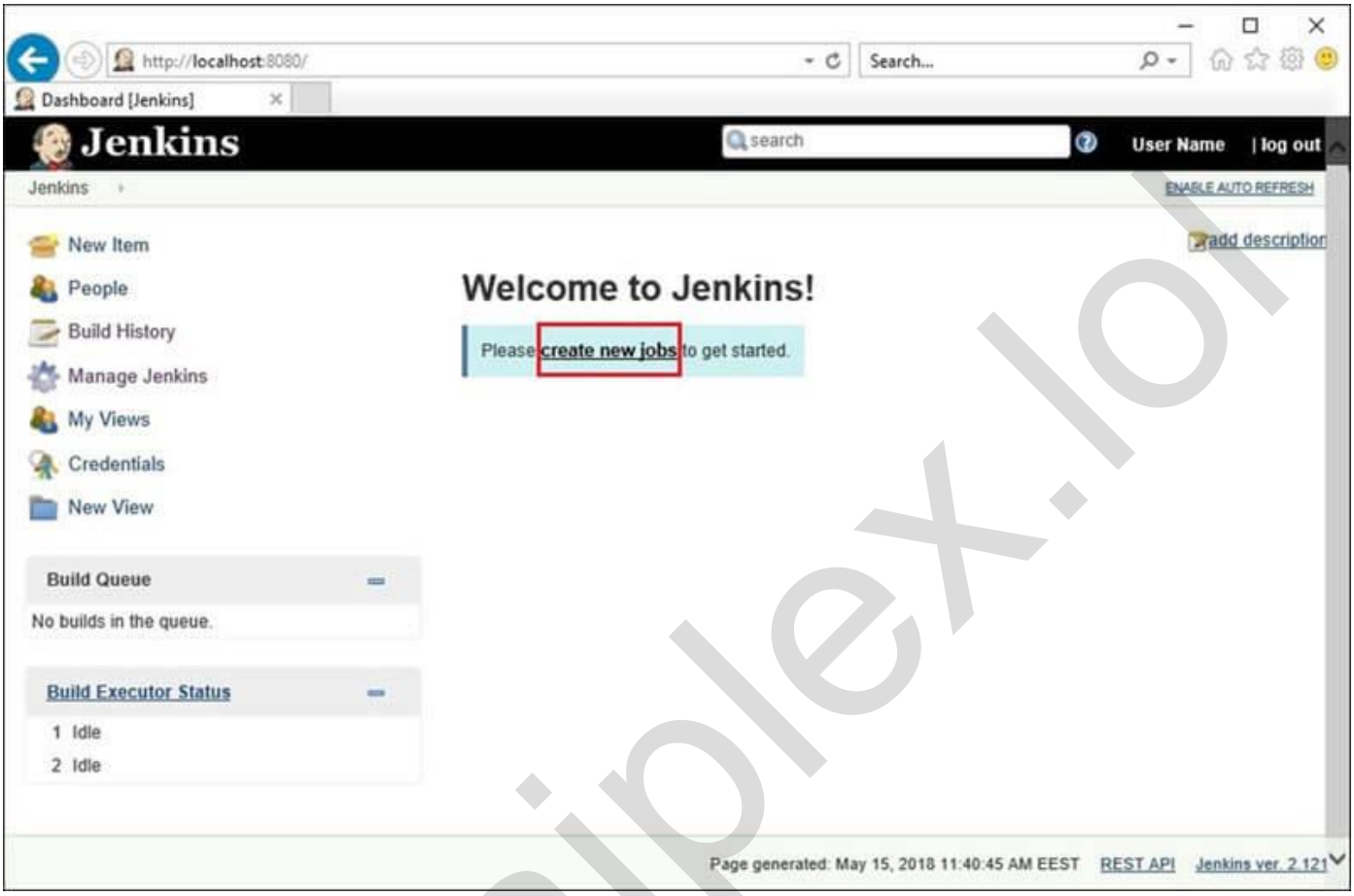
Get more free courses at www.aiplex.lol



Below you can find the Jenkins instance up and run, ready to create first Jenkins jobs:

www.aiplex.lol

Get more free courses at www.aiplex.lol



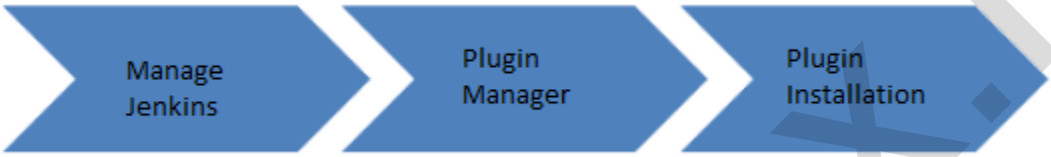
www.aiplex.lol

Get more free courses at www.aiplex.lol

Jenkins GitHub Integration: How to Install Git Plugin

You have learned, from previous Guru99 tutorials, what Jenkins is and how to install it onto a Windows system. Assuming you have completed those basic steps, we shall now move on to Plugin management.

Jenkins has outstanding plugin support. There are thousands of third-party application plugins available on their website. To know if Jenkins supports the third-party applications you have in mind, check their plugins directory at https://plugins.jenkins.io/.



In this Jenkins tutorial, you will learn:

- Installation of Plugins in Jenkins
- How to Install Git Plugin in Jenkins
- How to Integrate Jenkins With GitHub

Installation of Plugins in Jenkins

Jenkins comes with a pretty basic setup, so you will need to install the required plugins to enable respective third-party application support.

GitHub is a web-based repository of code which plays a major role in DevOps. It provides a common platform for multiple developers working on the same code/project to upload and retrieve updated code, thereby facilitating continuous integration.

Jenkins needs to have GitHub plugin installed to be able to pull code from the GitHub repository.

You need not install a GitHub plugin if you have already installed the Git plugin in response to the prompt during the Jenkins' installation setup. But if not, here is how you install GitHub plugins in Jenkins and pull code from a GitHub repository.

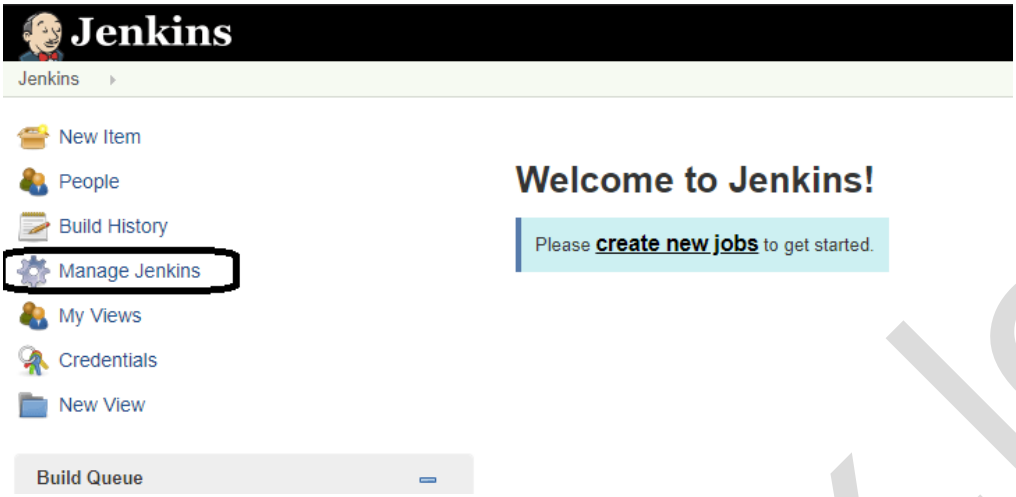
How to Install Git Plugin in Jenkins

Following is a step by step process on how to Install Git plugin in Jenkins:

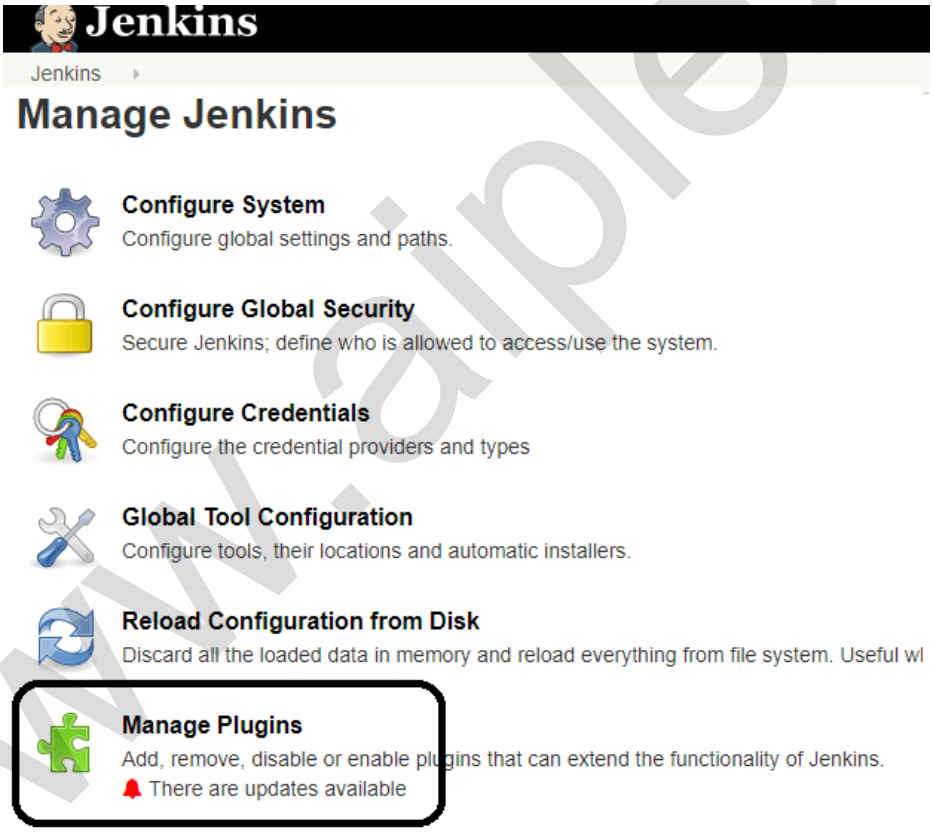
Step 1: Click on the **Manage Jenkins** button on your Jenkins dashboard:

www.aiplex.lol

Get more free courses at www.aiplex.lol



Step 2: Click on **Manage Plugins**:



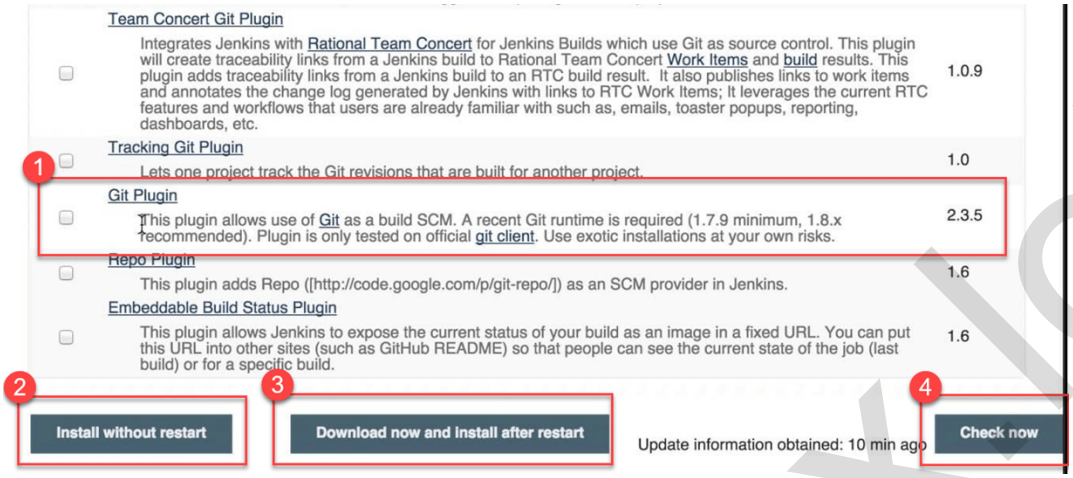
Step 3: In the Plugins Page

1. Select the GIT Plugin
2. Click on **Install without restart.** The plugin will take a few moments to finish downloading depending on your internet connection, and will be installed automatically.
3. You can also select the option **Download now and Install after restart** button. In which plugin is installed after restart

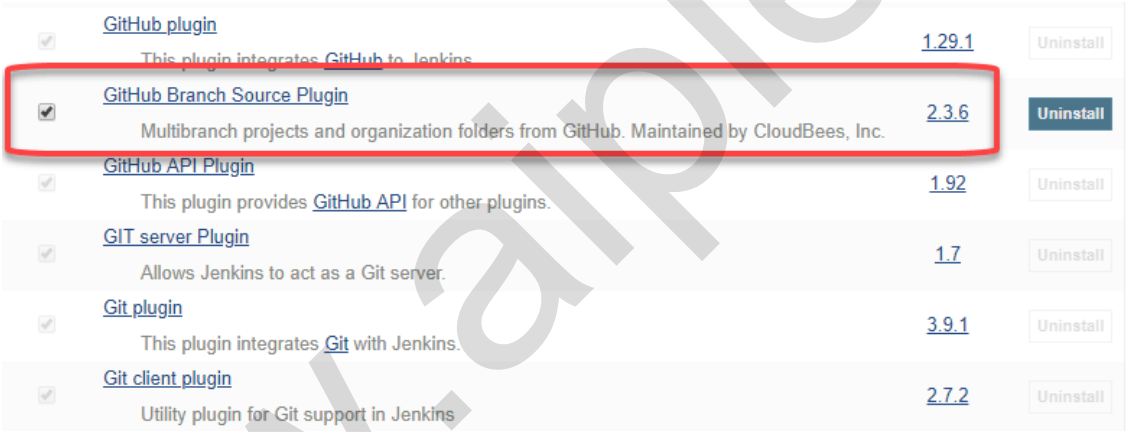
www.aiplex.lol

Get more free courses at www.aiplex.lol

4. You will be shown a "No updates available" message if you already have the Git plugin installed.



Step 4: Once the plugins have been installed, go to **Manage Jenkins** on your Jenkins dashboard. You will see your plugins listed among the rest.



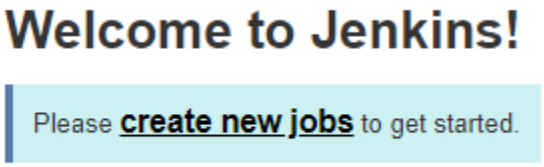
How to Integrate Jenkins With GitHub

We shall now discuss the process of integrating Jenkins and GitHub a Windows system:

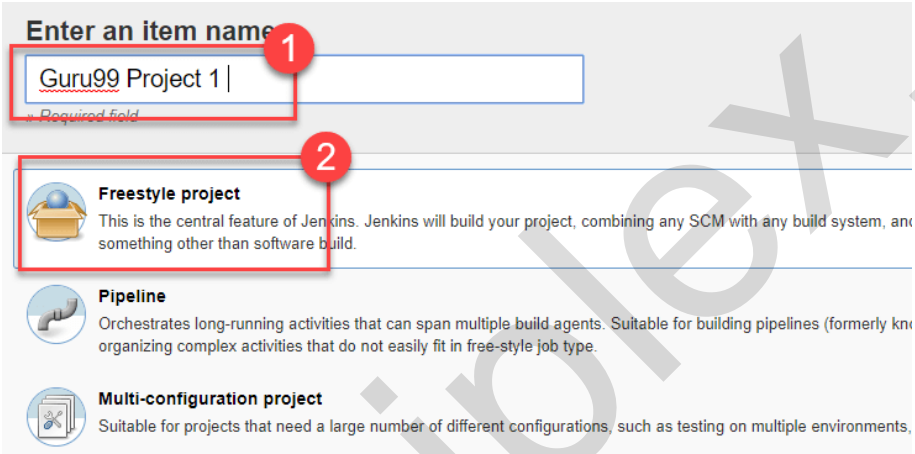
Step 1) Create a new job in Jenkins, open the Jenkins dashboard with your Jenkins URL. For example, http://localhost:8080/

Click on **create new jobs**:

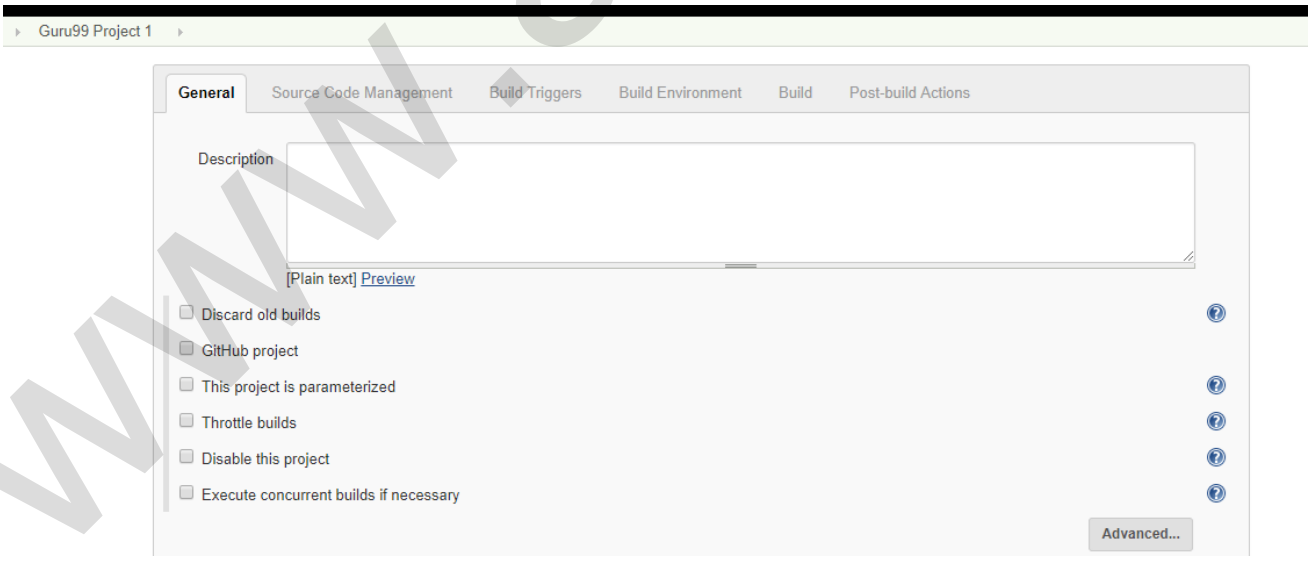
www.aiplex.lol



Step 2) Enter the item name, select job type and click **OK**. We shall create a Freestyle project as an example.

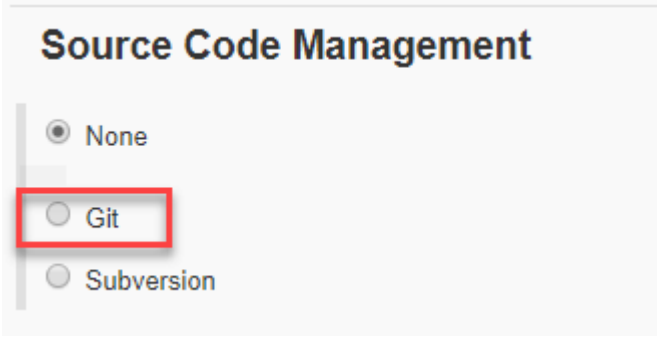


Step 3) Once you click **OK,** the page will be redirected to its project form. Here you will need to enter the project information:



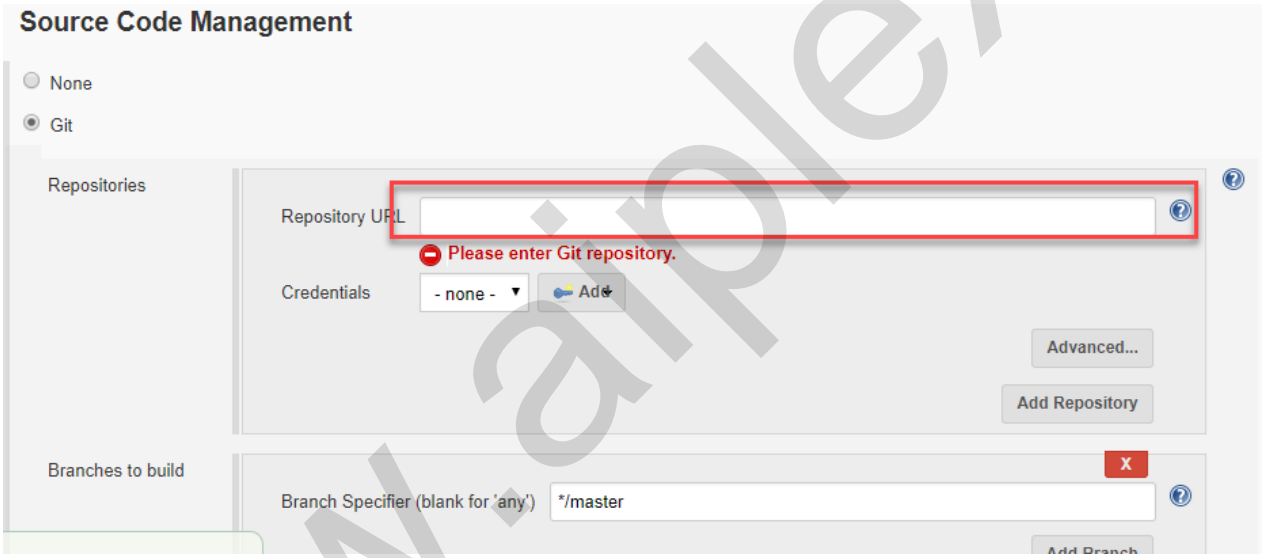
Step 4) You will see a **Git** option under **Source Code Management** if your Git plugin has been installed in Jenkins:

www.aiplex.lol

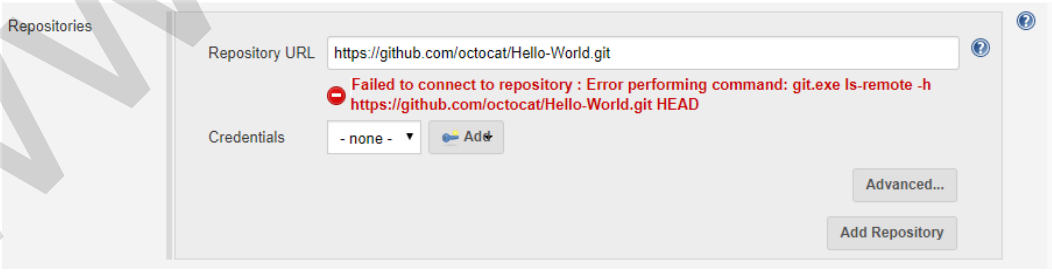


NOTE: If the **Git** option does not appear, try re-installing the plugins, followed by a restart and a re-login into your Jenkins dashboard. You will now be able to see the **Git** option as mentioned above.

Step 5) Enter the Git repository URL to pull the code from GitHub.



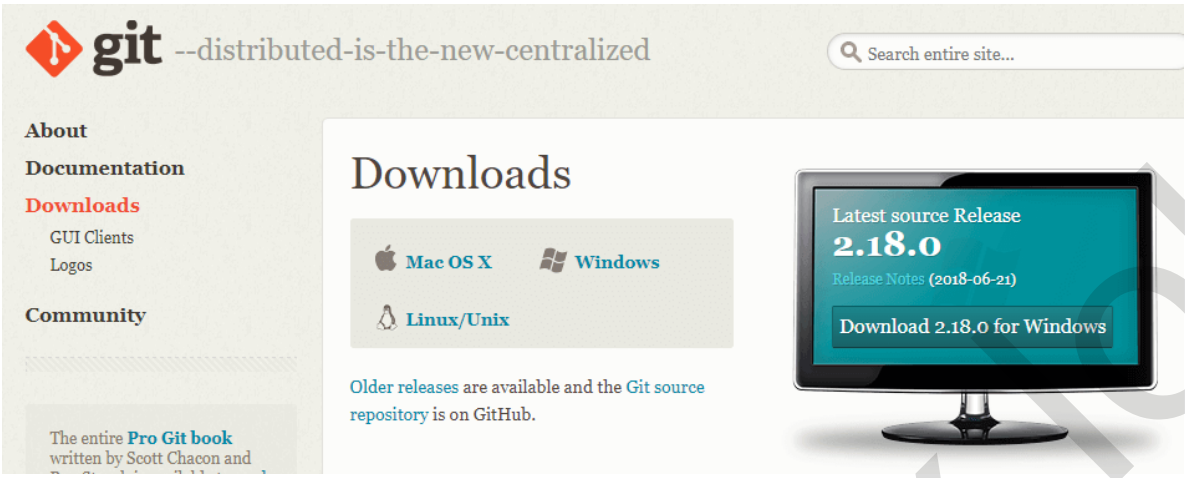
Step 6) You might get an error message the first time you enter the repository URL. For example:



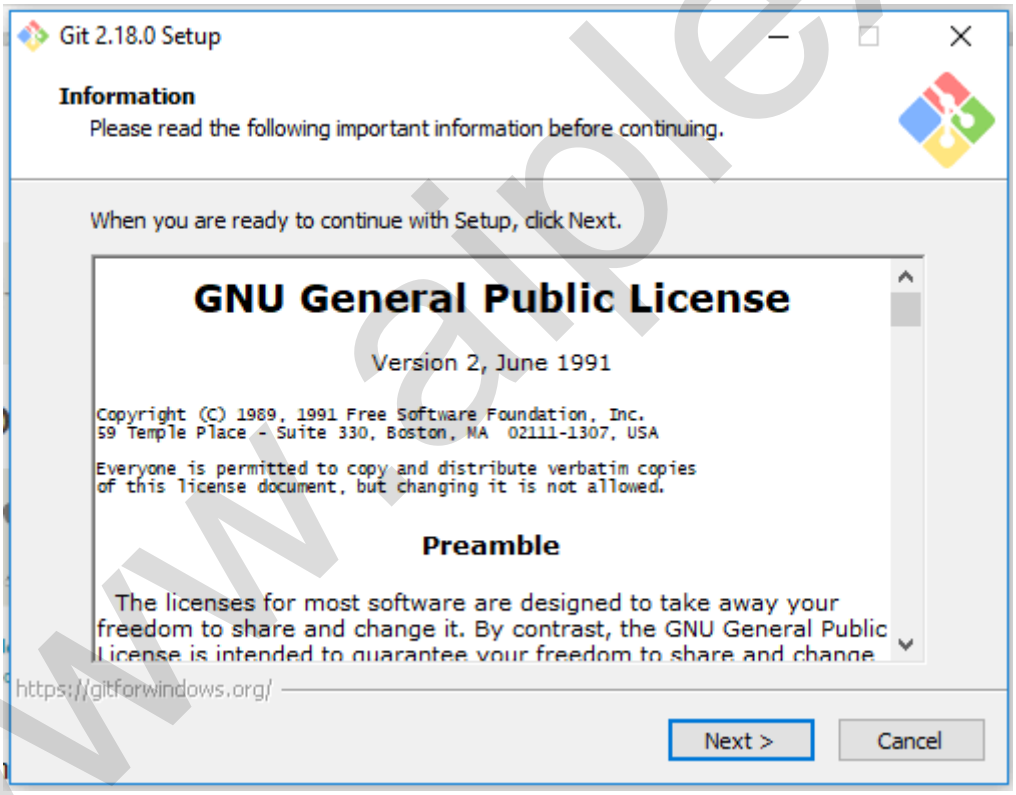
This happens if you do not have Git installed in your local machine. To install Git in your local machine, go to https://git-scm.com/downloads

www.aiplex.lol

Get more free courses at www.aiplex.lol



Download the appropriate Git file for your Operating System, in this case, Windows, and install it onto your local machine running Jenkins. Complete the onscreen instructions to install GIT.



Step 7) You can execute Git repositories in your Jenkins once Git has been installed on your machine. To check if it has been successfully installed onto your system, open your **command prompt,** type "Git" and press enter. You should see different options come up for Git:

www.aiplex.lol

```
Microsoft Windows [Version 10.0.16299.309]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\alex>git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

These are common Git commands used in various situations:
```

This means that Git has been installed in your system.

Note: If you have GIT already installed in your system, just add git.exe path in Global Tool Configuration.

Step 8) Once you have everything in place, try adding the Git URL into Jenkins. You will not see any error messages for Jenkins Git integration:

Git is now properly configured on your system.

www.aiplex.lol

How to Create/Add Users in Jenkins & Manage Permissions

Generally, in a large organization, there are multiple, separate teams to manage and run jobs in Jenkins. But managing this crowd of users and assigning roles to them can prove troublesome.

By default, Jenkins comes with very basic user creation options. You can create multiple users but can only assign the same global roles and privileges to them. This not ideal, especially for a large organization.

The **Role Strategy Plugin** enable you to assign different roles and privileges to different users**.** You will first need to install the plugin in your Jenkins mange environment.

In this Jenkins tutorial, you will learn

- How to Create a User in Jenkins
- How to Install Role Strategy Plugin in Jenkins
- How to Manage Users and Roles in Jenkins
- How to Assign Roles in Jenkins
- How to Create Project Roles in Jenkins

How to Create/Add a User in Jenkins

Below are the steps to create new user in Jenkins:

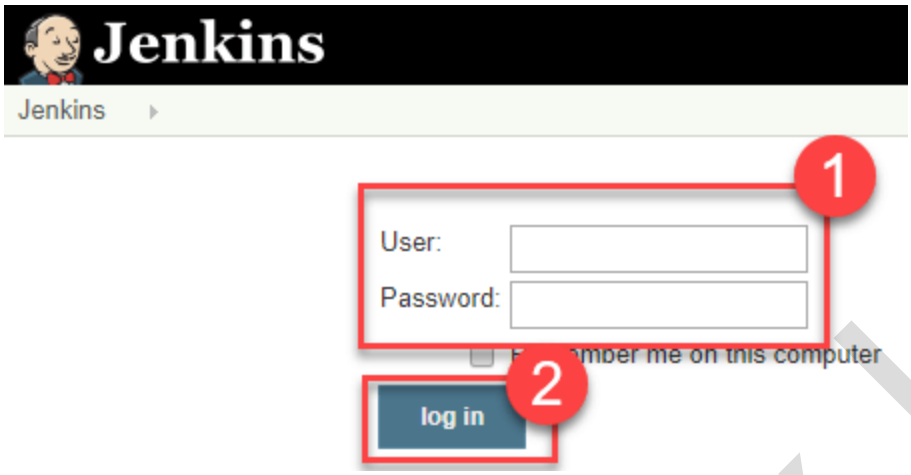
Step 1) Login to Jenkins Dashboard

Login to your Jenkins dashboard by visiting http://localhost:8080/

If you haven't installed Jenkins in your local server, go to the appropriate URL and access your dashboard by using your login credentials.

www.aiplex.lol

Get more free courses at www.aiplex.lol



Step 2) Choose the option

You will now see options to create and add user in Jenkins and manage current users.

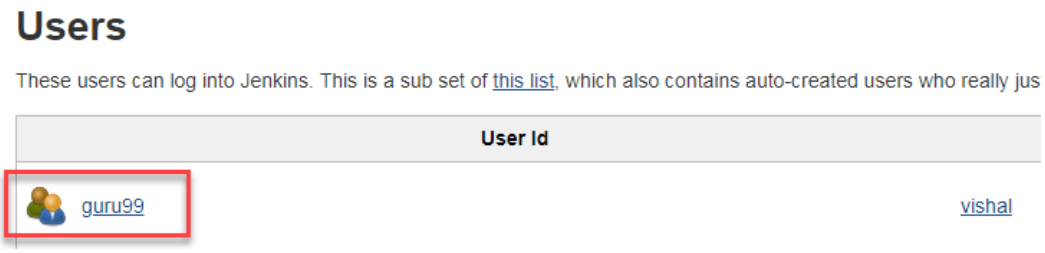
Step 3) Create a new User

- Under Manage Jenkins, Click Create User
- Enter Jenkins add user details like password, name, email etc.
- Click Create User

Step 4) User is created

You will see on the dashboard that a new Jenkins create user as per the details entered.

www.aiplex.lol



How to Install Role Strategy Plugin in Jenkins

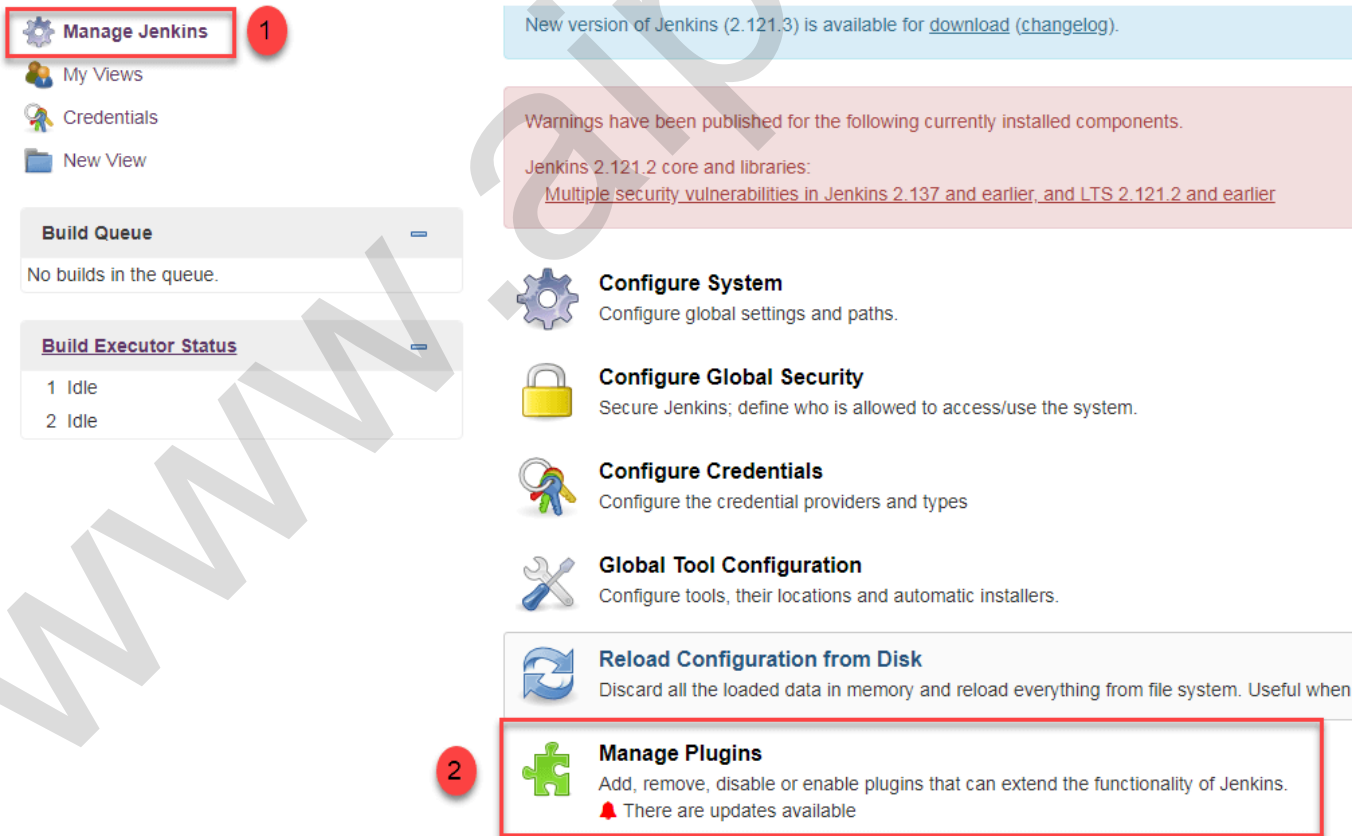
There are two methods for installing plugins in Jenkins:

1. Installing it through your Jenkins dashboard
2. Downloading the plugin from Jenkins website and installing it manually.

Step 1)

1. Go to **Manage Jenkins**

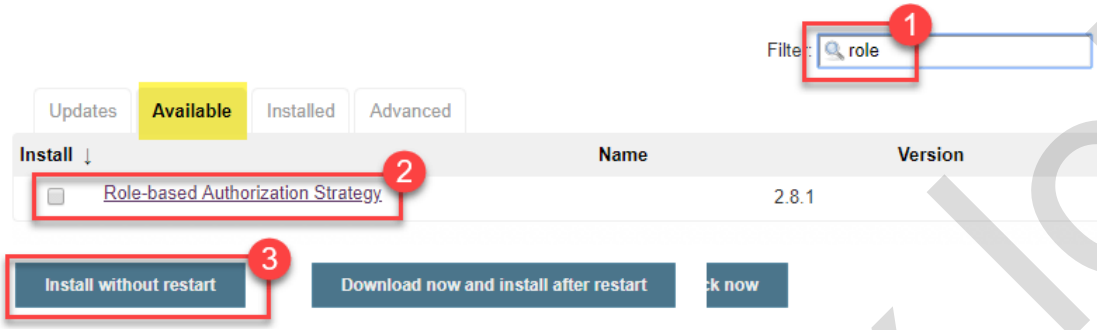
2. Click on the Manage Plugins option



Step 2)

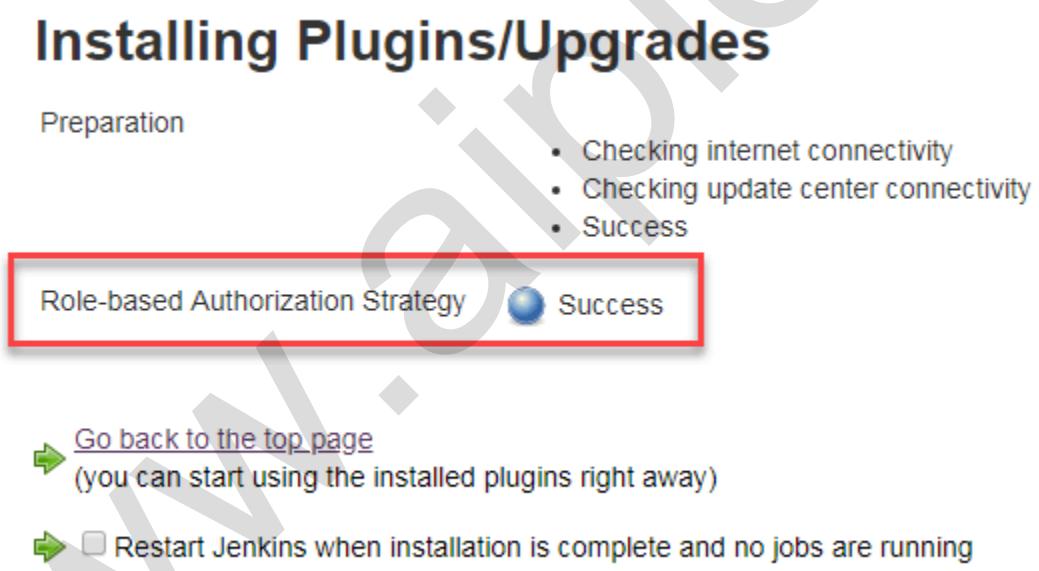
www.aiplex.lol

1. In available section, screen Search for "role".
2. Select Role-**based Authorization Strategy** plugin
3. Click on "**Install without restart**" (make sure you have an active internet connection)



Step 3)

Once the plugin is installed, a "success" status will be displayed.

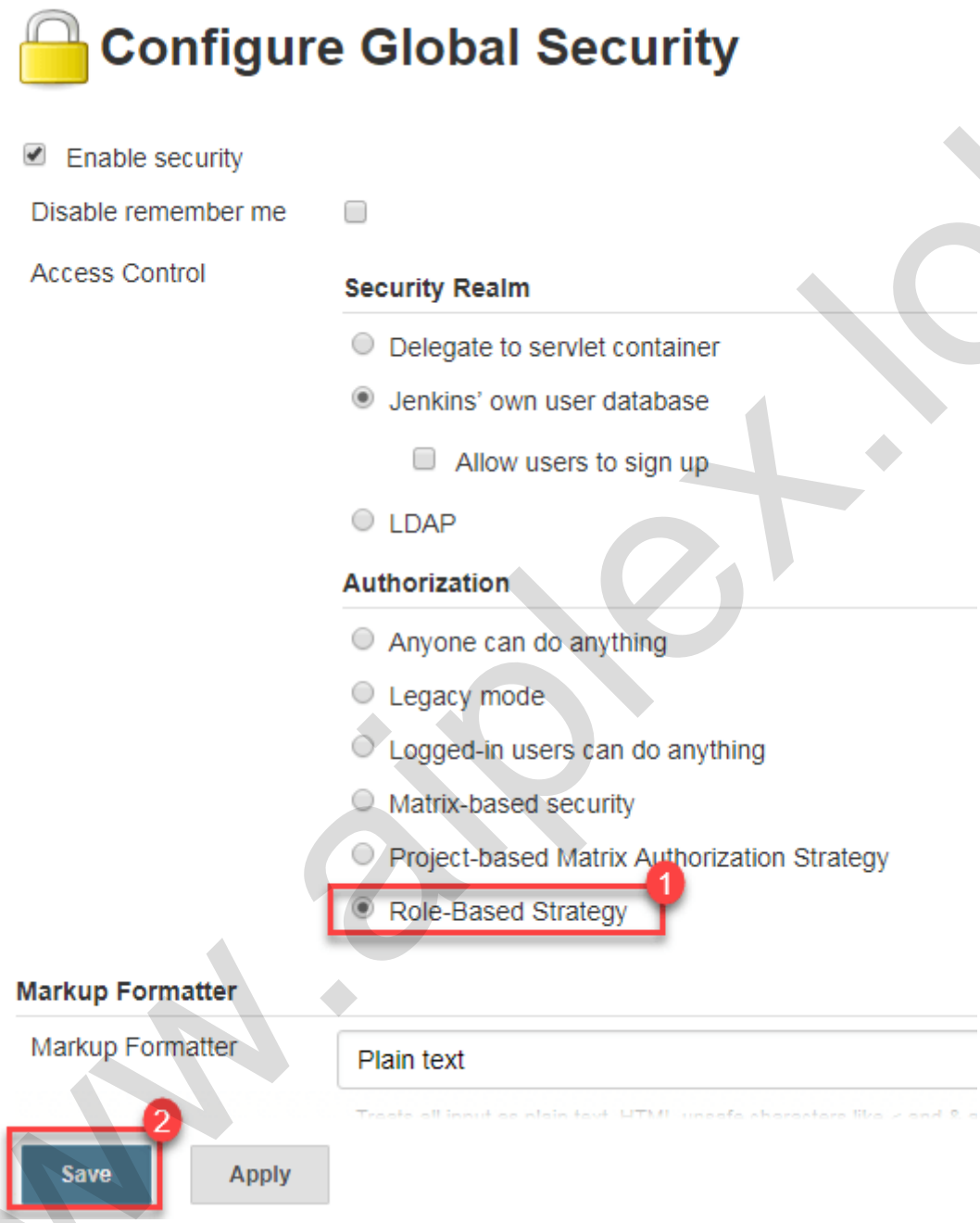


Click on **Go back to the top page.**

Step 4) Go to **Manage Jenkins** -> Configure Global Security ->
Under **Authorization,** select **Role Based Strategy**. Click on **Save**.

www.aiplex.lol

Get more free courses at www.aiplex.lol



How to Manage Users and Roles in Jenkins

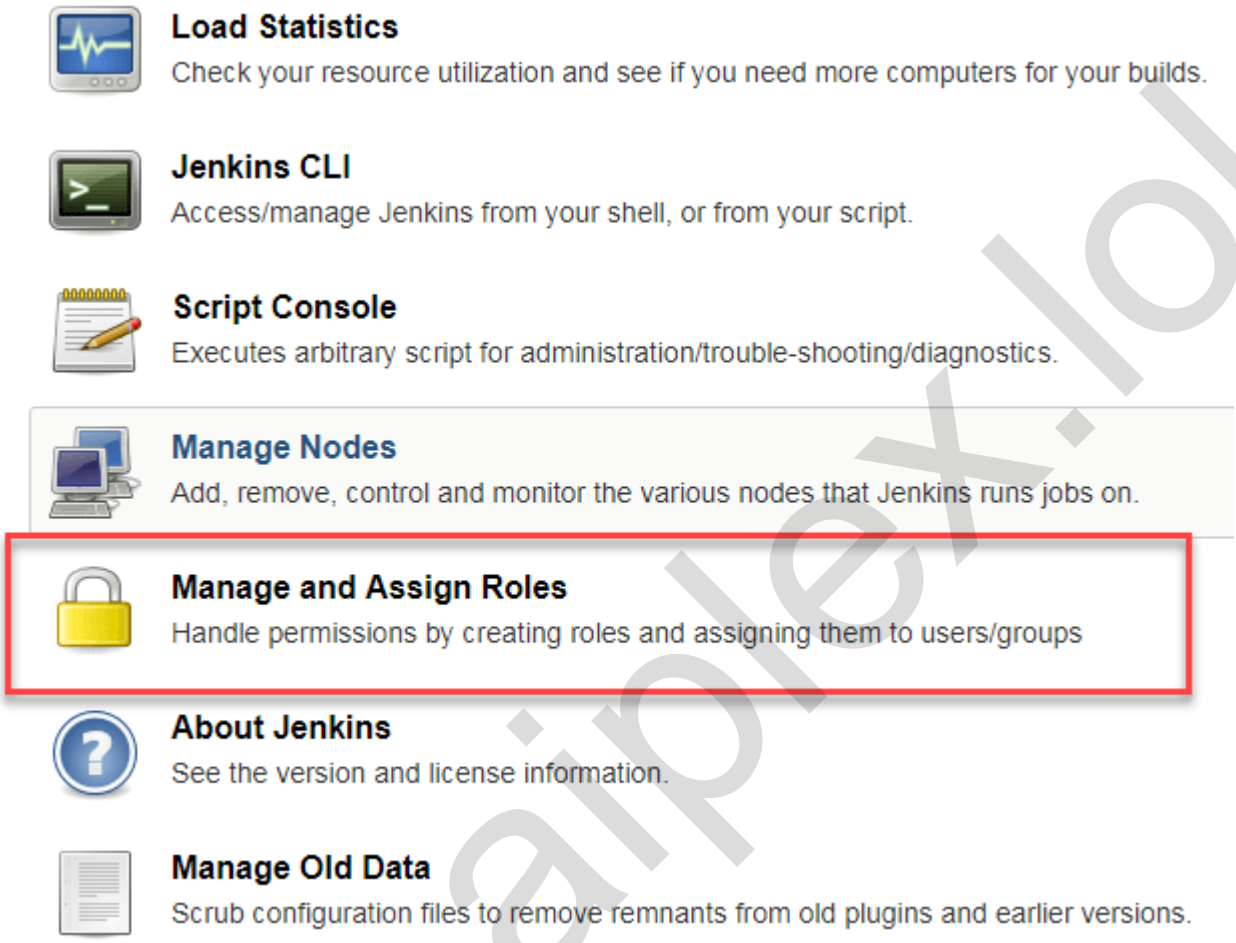
Following are the steps on how to manage and assign roles in Jenkins:

Step 1)

1. Go to **Manage Jenkins**

www.aiplex.lol

2. Select **Manage and Assign Roles**

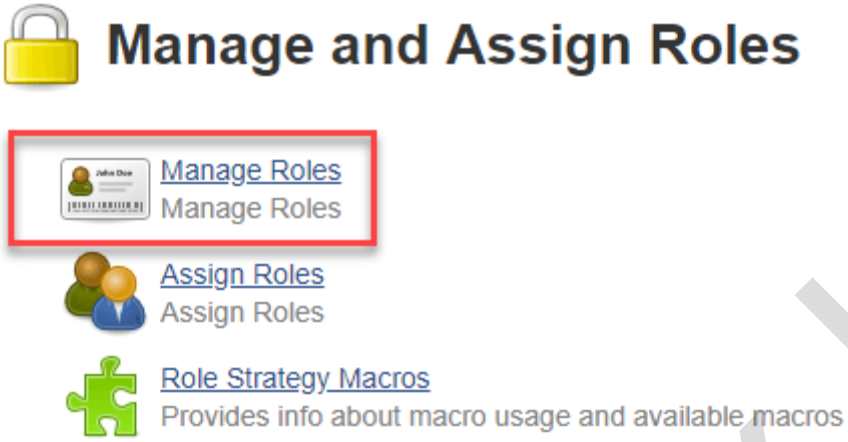


Note: that the **Manage and Assign Roles** option will only be visible if you've installed the role strategy plugin.

Step 2) Click on **Manage Roles** to add new roles based on your organization.

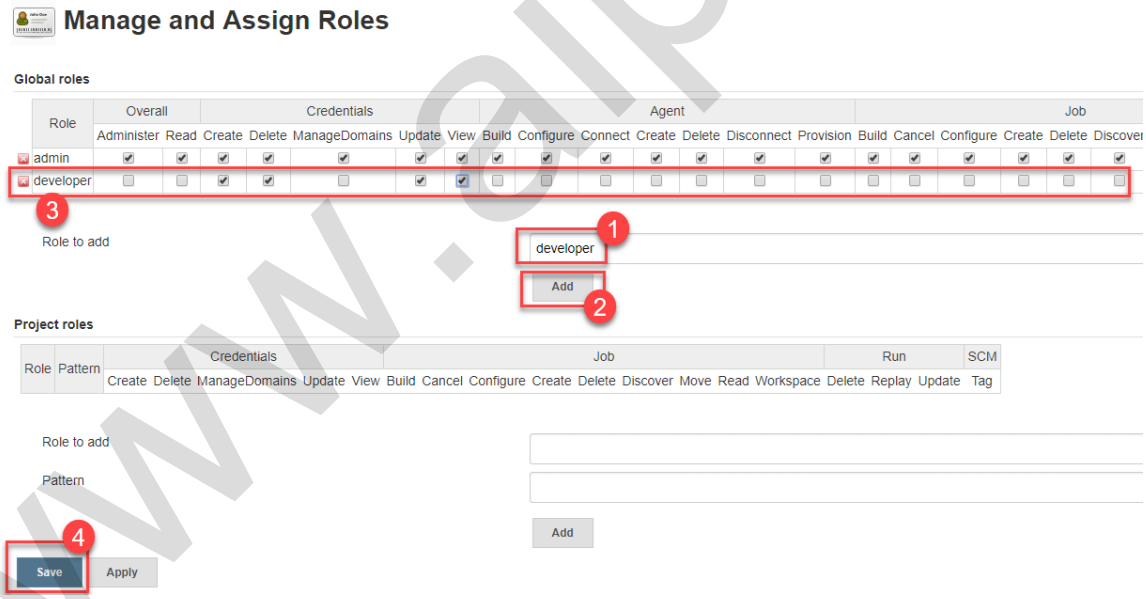
www.aiplex.lol

Get more free courses at www.aiplex.lol



Step 3) To create a new role called "developer",

1. Type "developer" under "role".
2. Click on "Add" to create a new role.
3. Now, select the Jenkins user permissions you want to assign to the "Developer" role.
4. Click Save



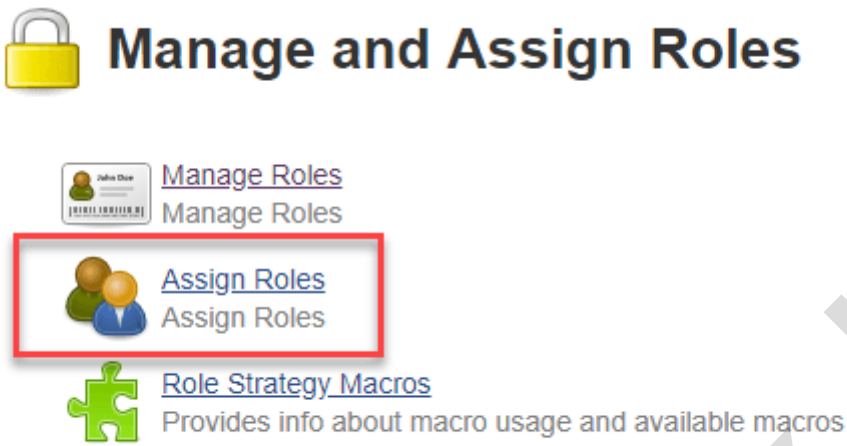
How to Assign Roles in Jenkins

Step 1) Now that you have created roles, let us assign them to specific users.

1. Go to **Manage Jenkins**
2. Select Manage and Assign Roles

www.aiplex.lol

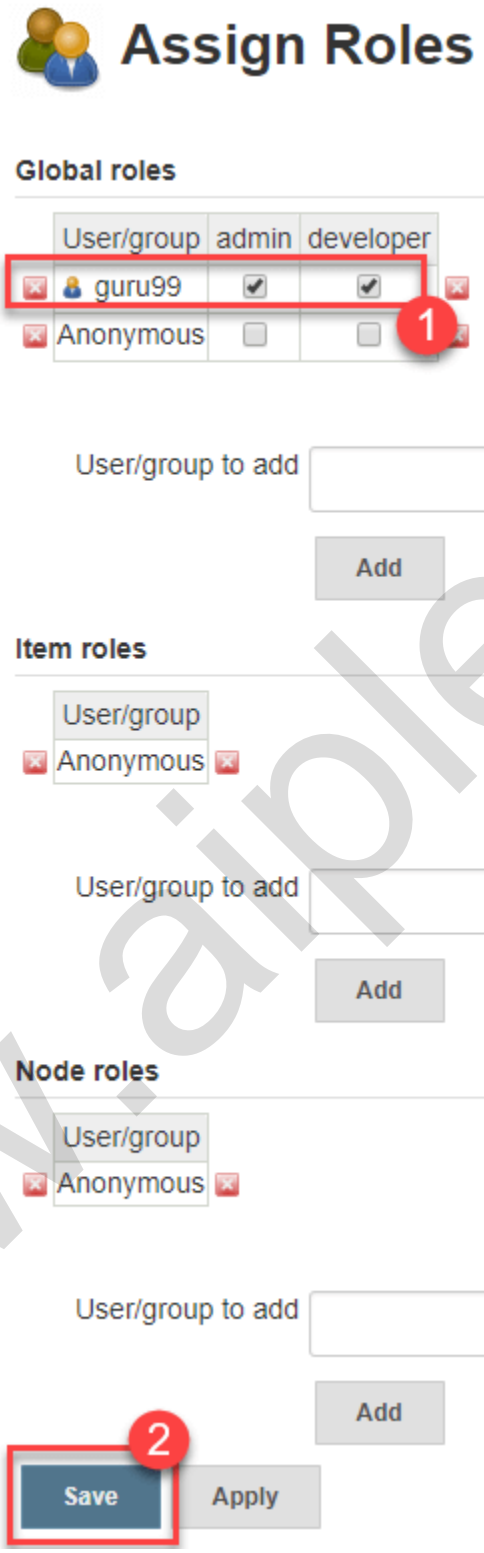
Get more free courses at www.aiplex.lol



Step 2) We shall add the new role "developer" to user "**guru99**"

1. Selector developer role checkbox
2. Click Save

www.aiplex.lol



You can assign any role to any user, as per your need.

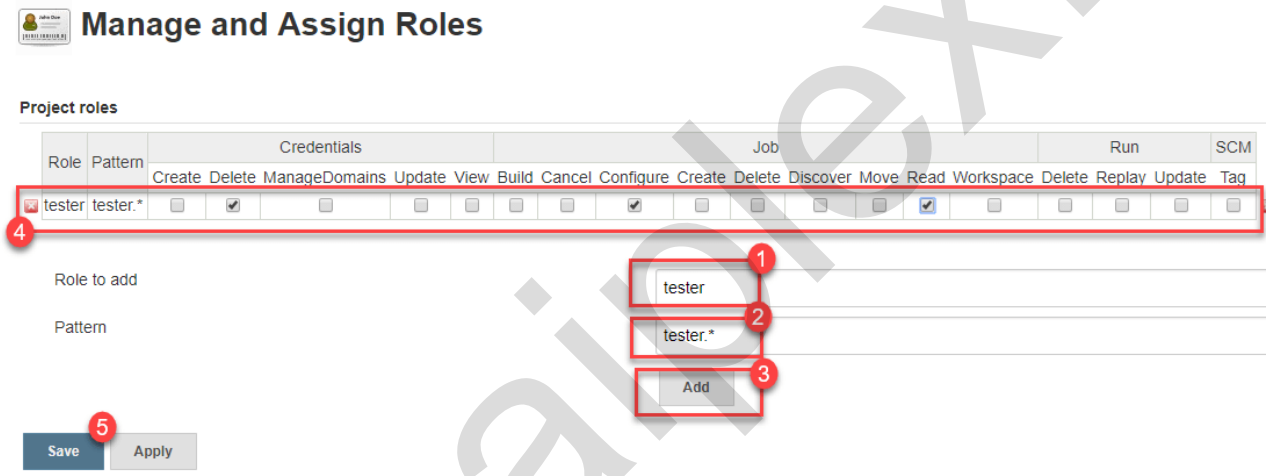
www.aiplex.lol

How to Create Project Roles in Jenkins

You can create project specific roles under **Project Roles.**

Step 1) In Jenkin's Manage and Assign Roles

1. Enter a role as "tester"
2. Add a pattern to this by adding **tester.***, so that any username starting with "tester" will be assigned the project role you specify.
3. Click Add
4. Select privileges
5. Click Save



www.aiplex.lol

Get more free courses at www.aiplex.lol

How to Create a New Build Job in Jenkins Freestyle Project

What is a Jenkins Freestyle Project?

Jenkins Freestyle Project is a repeatable build job, script, or pipeline that contains steps and post-build actions. It is an improved job or task that can span multiple operations. It allows you to configure build triggers and offers project-based security for your Jenkins project. It also offers plugins to help you build steps and post-build actions.

The types of actions you can perform in a Jenkins build step or post-build action are quite limited. There are many standard plugins available within a Jenkins Freestyle Project to help you overcome this problem.

How to Create a Job in Jenkins

How to Create a New Build Job in Jenkins

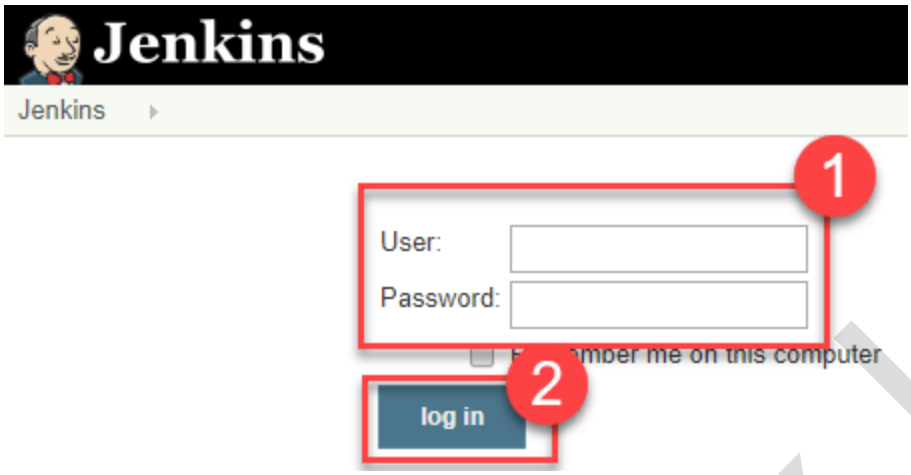
The freestyle build job is a highly flexible and easy-to-use option. You can use it for any type of project; it is easy to set up, and many of its options appear in other build jobs. Below is a step by step process to create job in Jenkin.

Step 1) Login to Jenkins

To create a Jenkins freestyle job, log on to your Jenkins dashboard by visiting your Jenkins installation path. Usually, it will be hosted on localhost at http://localhost:8080 If you have installed Jenkins in another path, use the appropriate URL to access your dashboard as shown in the below Jenkins job creation example.

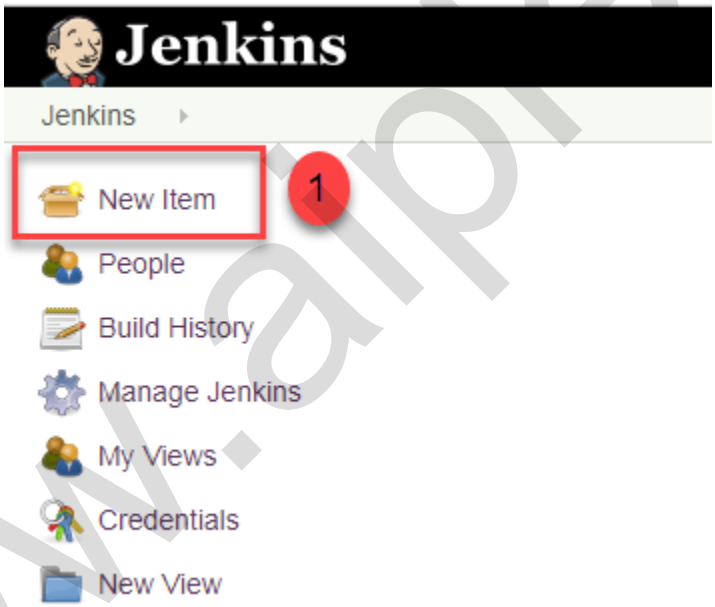
www.aiplex.lol

Get more free courses at www.aiplex.lol



Step 2) Create New Item

Click on "**New Item**" at the top left-hand side of your dashboard.



Step 3) Enter Item details

In the next screen,

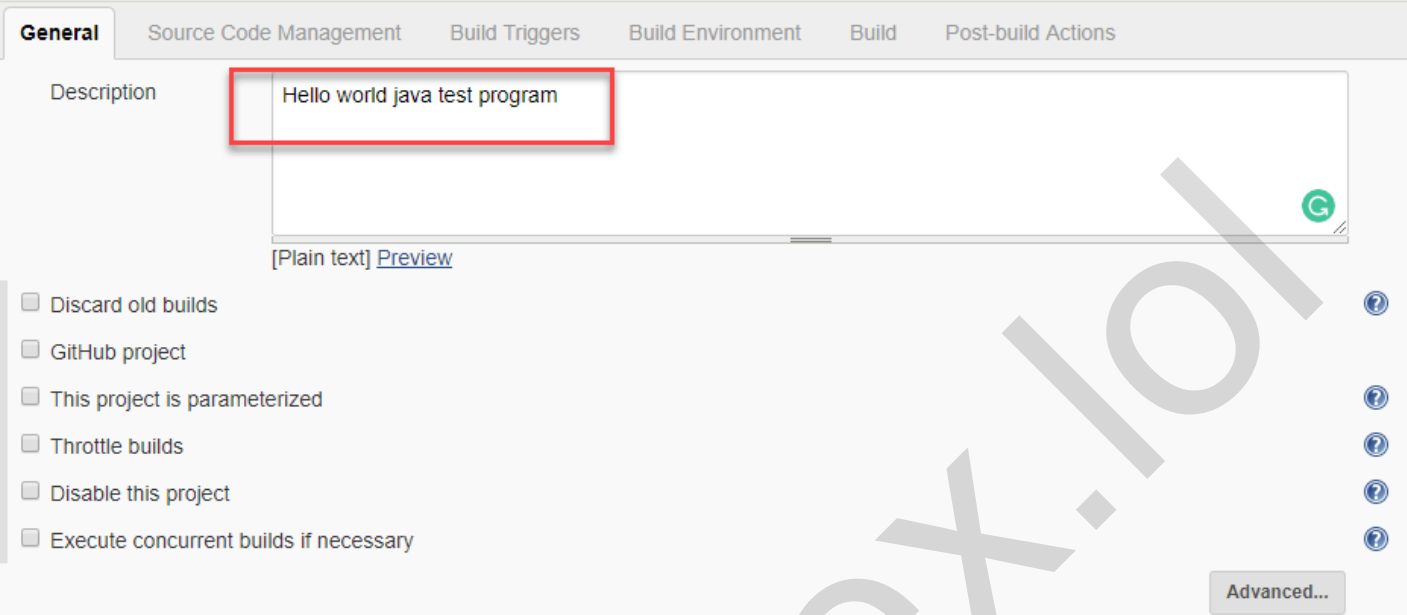
1. Enter the name of the item you want to create. We shall use the "Hello world" for this demo.
2. Select Freestyle project
3. Click Okay

www.aiplex.lol

Step 4) Enter Project details

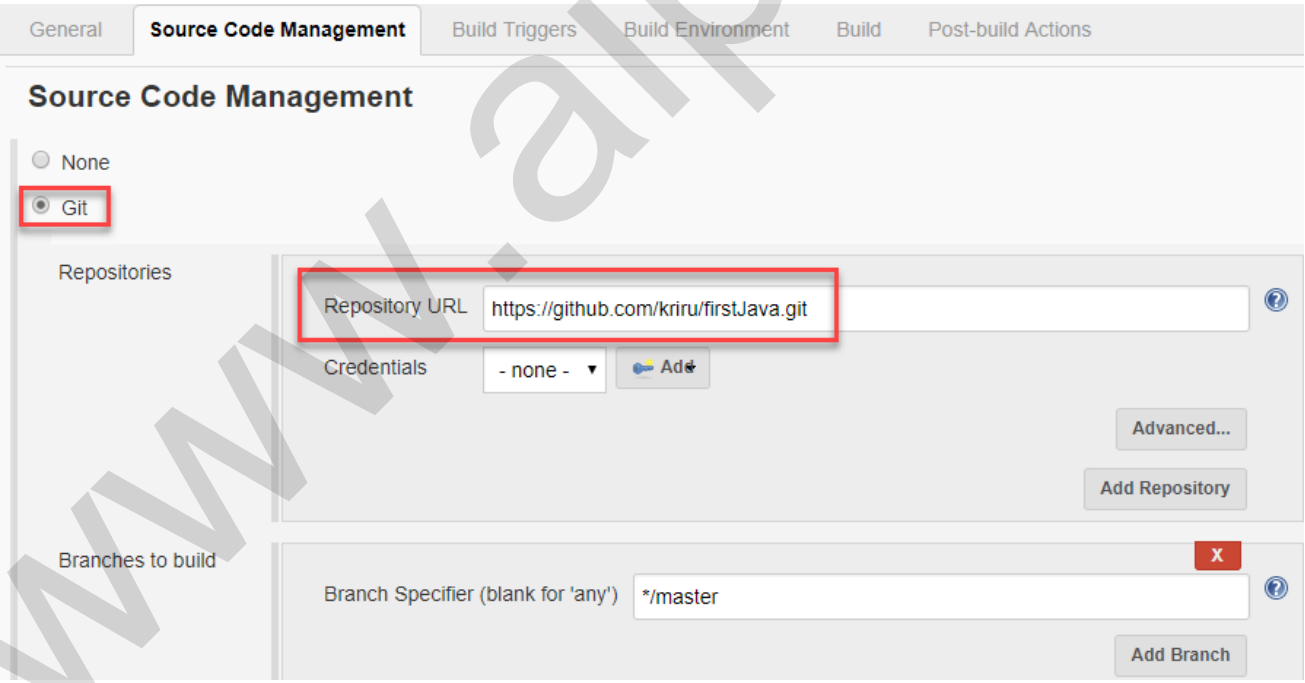
Enter the details of the project you want to test.

www.aiplex.lol



Step 5) Enter repository URL

Under Source Code Management, Enter your repository URL. We have a test repository located at https://github.com/kriru/firstJava.git



It is also possible for you to use a local repository.

If your GitHub repository is private, Jenkins will first validate your login credentials with GitHub and only then pull the source code from your GitHub repository.

www.aiplex.lol

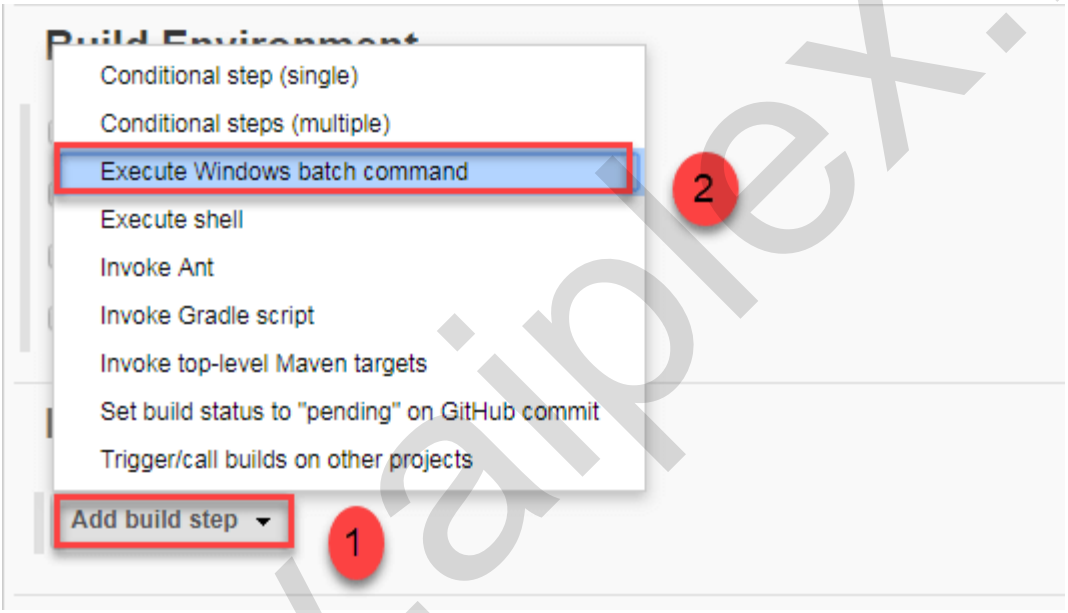
Step 6) Tweak the settings

Now that you have provided all the details, it's time to build the code. Tweak the settings under the **build** section to build the code at the time you want. You can even schedule the build to happen periodically, at set times.

Under **build**,

1. Click on "**Add build step**"

2. Click on "**Execute Windows batch command**" and add the commands you want to execute during the build process.



Here, I have added the java commands to compile the java code.

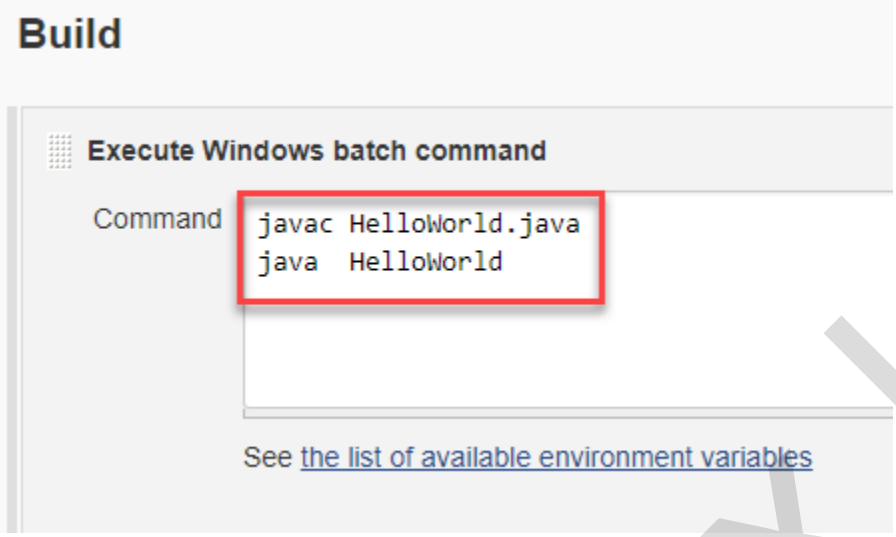
I have added the following windows commands:

javac HelloWorld.java

java HelloWorld

www.aiplex.lol

Get more free courses at www.aiplex.lol



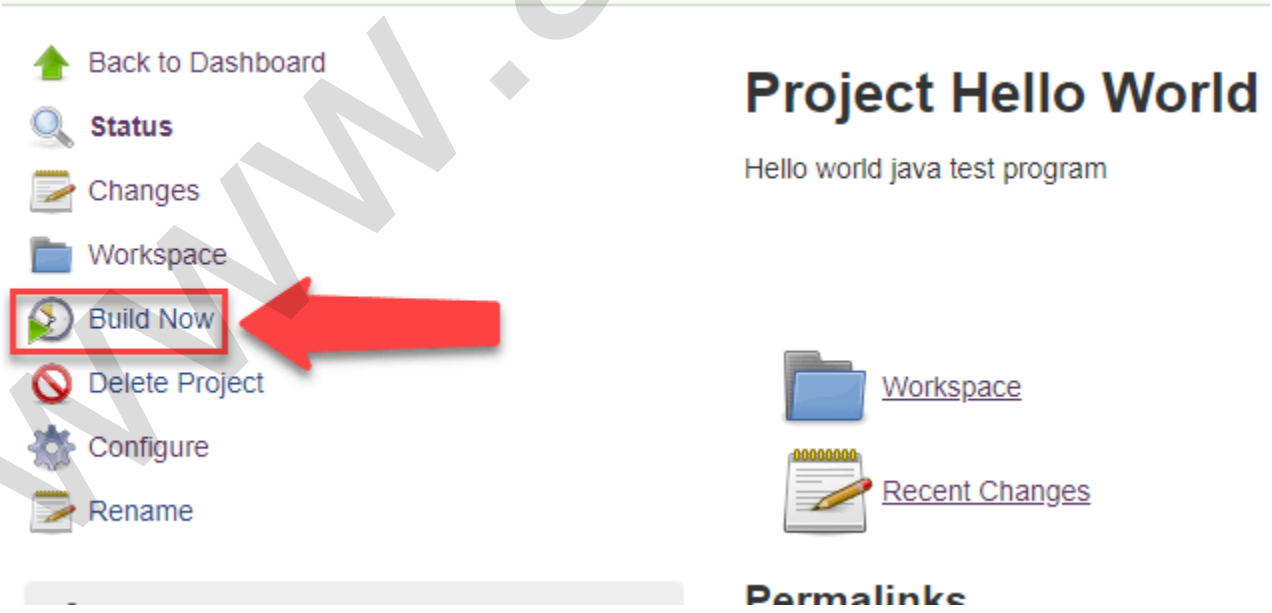
Step 7) Save the project

When you have entered all the data,

1. Click **Apply**
2. **Save** the project.

Step 8) Build Source code

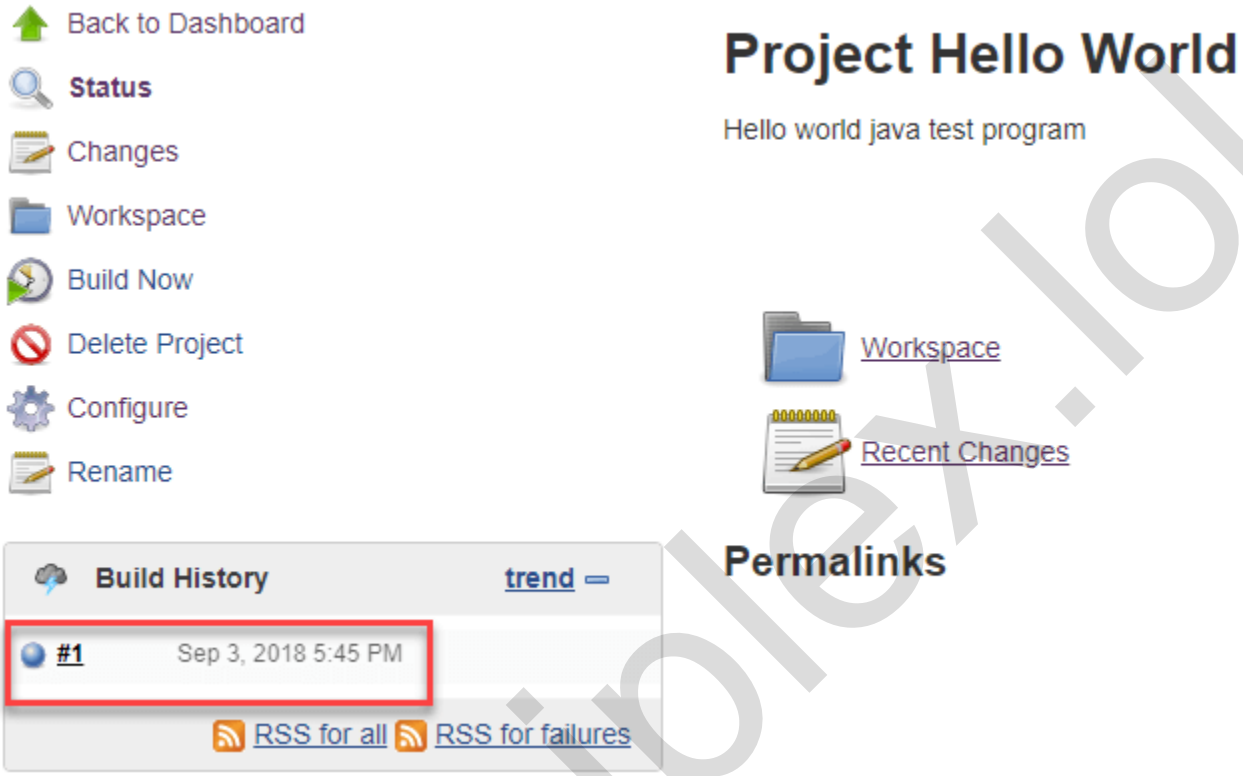
Now, in the main screen, Click the **Build Now** button on the left-hand side to build the source code.



Step 9) Check the status

www.aiplex.lol

After clicking on **Build now,** you can see the status of the build you run under **Build History**.



Step 10) See the console output

Click on the **build number** and then Click on **console output** to see the status of the build you run. It should show you a success message, provided you have followed the setup properly as shown in the below Jenkins create new job example.

www.aiplex.lol

In sum, we have executed a HelloWorld program hosted on GitHub. Jenkin pulls the code from the remote repository and builds continuously at a frequency you define.

Summary

- Jenkins Freestyle Project is a repeatable build job, script, or pipeline that contains steps and post-build actions. It is an improved job or task that can span multiple operations.
- The types of actions you can perform in a build step or post-build action are quite limited. There are many standard plugins available within a Jenkins Freestyle Project to help you overcome this problem.
- Freestyle build Jenkins jobs are highly flexible and easy-to-use. You can use it for any type of project; it is easy to set up, and many of its options appear in other build Jenkins jobs.
- If your GitHub repository is private, Jenkins will first validate your login credentials with GitHub and only then pull the source code from your GitHub repository.

www.aiplex.lol

Get more free courses at www.aiplex.lol

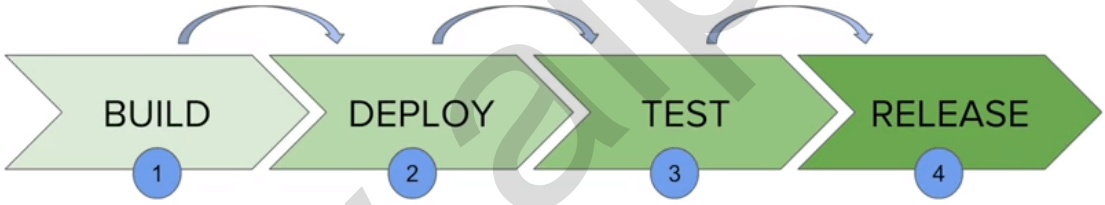
Jenkins Pipeline Tutorial: How to Create JenkinsFile (Example)

What is Jenkins Pipeline?

Jenkins Pipeline is a combination of plugins that supports integration and implementation of continuous delivery pipelines. It has an extensible automation server to create simple and complex delivery pipelines as code via pipeline DSL. A Pipeline is a group of events interlinked with each other in a sequence.

What is Continuous Delivery Pipelines? How it Works?

In a Jenkins pipeline, every job or event has some sort of dependency on at least one or more events.



Working of Jenkins Continuous Delivery Pipelines

The picture above represents a continuous delivery pipeline in Jenkins. It contains a group of states called build, deploy, test and release. These events are interlinked with each other. Every state has its events, which work in a sequence called a continuous delivery pipeline.

A continuous delivery pipeline is an automated expression to display your process for getting software for version control. Thus, every change made in your software goes through a number of complex processes on its way to being released. It also involves developing the software in a reliable and repeatable manner, and progression of the built software through multiple stages of testing and deployment.

In this Jenkins pipeline tutorial, you will learn

- What is Jenkins Pipeline?
- What is a JenkinsFile?
- Why Use Jenkin's Pipeline?

Get more free courses at www.aiplex.lol

- Jenkins Pipeline Concepts
- Install Build Pipeline Plugin in Jenkins
- How to Create Jenkins Pipeline
- Running Jenkins pipeline
- Best Practices using Jenkins Pipeline

What is a JenkinsFile?

Jenkins pipelines can be defined using a text file called **JenkinsFile.** You can implement pipeline as code using JenkinsFile, and this can be defined by using a domain specific language (DSL). With JenkinsFile, you can write the steps needed for running a Jenkins pipeline.

The benefits of using **JenkinsFile are**:

- You can create pipelines automatically for all branches and execute pull requests with just one **JenkinsFile.**
- You can review your Jenkins code on the pipeline
- You can audit your Jenkins pipeline
- This is the singular source for your pipeline and can be modified by multiple users.

JenkinsFile can be defined by either Web UI or with a Jenkins File.

Declarative versus Scripted pipeline syntax:

There are two types of Jenkins pipeline syntax used for defining your JenkinsFile.

1. Declarative
2. Scripted

Declarative:

Declarative pipeline syntax offers an easy way to create pipelines. It contains a predefined hierarchy to create Jenkins pipelines. It gives you the ability to control all aspects of a pipeline execution in a simple, straight-forward manner.

Scripted:

Scripted Jenkins pipeline runs on the Jenkins master with the help of a lightweight executor. It uses very few resources to translate the pipeline into atomic commands. Both declarative and scripted syntax are different from each other and are defined totally differently.

www.aiplex.lol

Get more free courses at www.aiplex.lol

Why Use Jenkin's Pipeline?

Jenkins is an open continuous integration server which has the ability to support the automation of software development processes. You can create multiple automation jobs with the help of use cases, and run them as a Jenkins pipeline.

Here are the reasons why you use should use Jenkins pipeline:

- Jenkins pipeline is implemented as a code which allows multiple users to edit and execute the pipeline process.
- Pipelines are robust. So if your server undergoes an unforeseen restart, the pipeline will be automatically resumed.
- You can pause the pipeline process and make it wait to resume until there is an input from the user.
- Jenkins Pipelines support big projects. You can run multiple jobs, and even use pipelines in a loop.

Jenkins Pipeline Concepts

Term	Description
Pipeline	The pipeline is a set of instructions given in the form of code for continuous delivery and consists of instructions needed for the entire build process. With pipeline, you can build, test, and deliver the application.
Node	The machine on which Jenkins runs is called a node. A node block is mainly used in scripted pipeline syntax.
Stage	A stage block contains a series of steps in a pipeline. That is, the build, test, and deploy processes all come together in a stage. Generally, a stage block is used to visualize the Jenkins pipeline process.
Step	A step is nothing but a single task that executes a specific process at a defined

www.aiplex.lol

Get more free courses at www.aiplex.lol

time. A pipeline involves a series of steps.

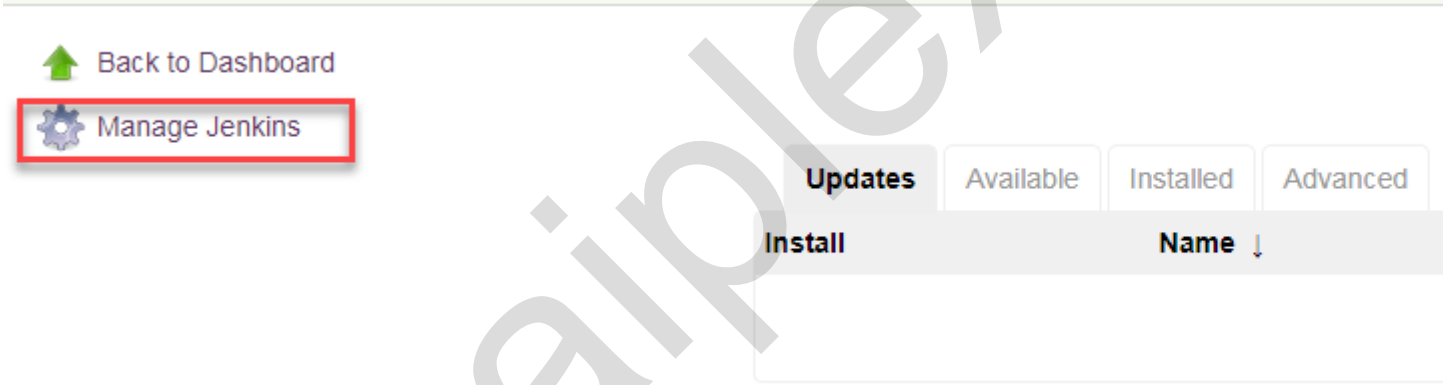
Install Build Pipeline Plugin in Jenkins

With the **build pipeline** plugin, you can create a pipeline view of incoming and outgoing jobs, and create triggers which require manual intervention.

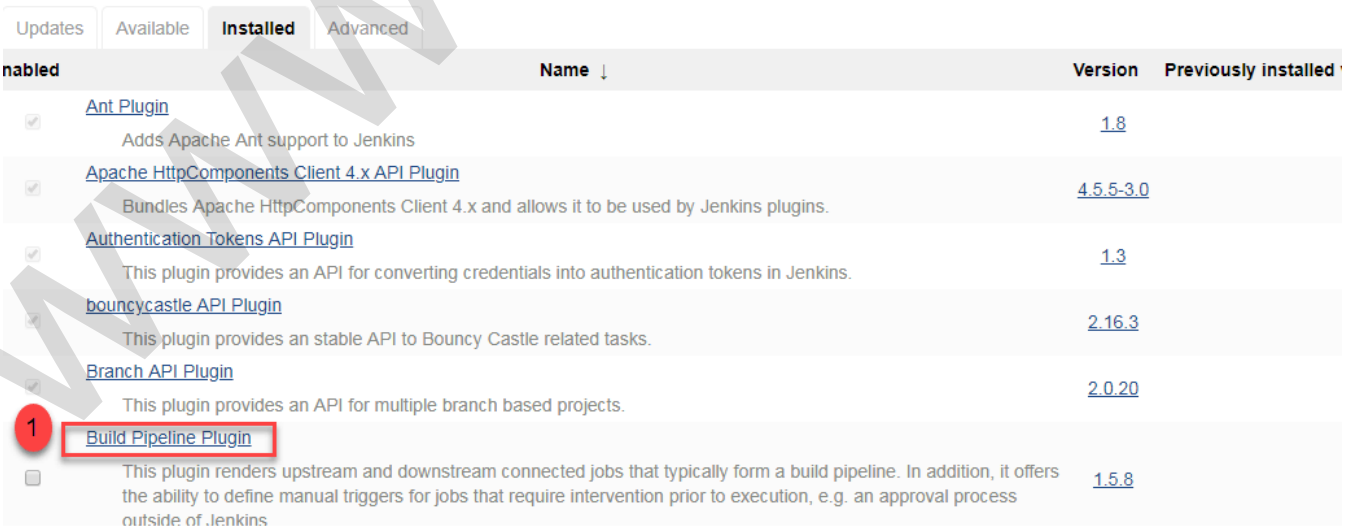
Here is how you can install the **build pipeline** plugin in your Jenkins:

Step 1) The settings for the plugin can be found under,

Manage Jenkins > Manage Plugins.



If you have already installed the plugin, it is shown under the installed tab.



www.aiplex.lol

Get more free courses at www.aiplex.lol

Step 2) If you do not have the plugin previously installed,

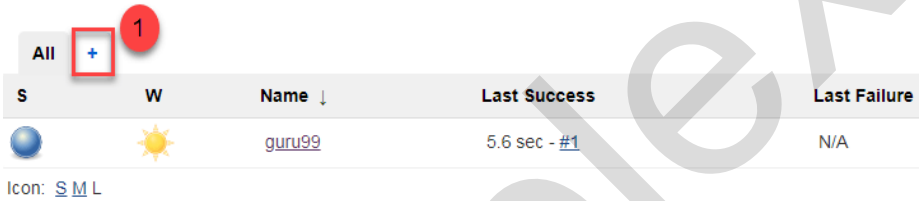
it shows up under the **Available** tab.

Once you have successfully installed the **build pipeline** plugin in your Jenkins, follow these steps to create your Jenkins pipeline:

How to Create Jenkins Pipeline

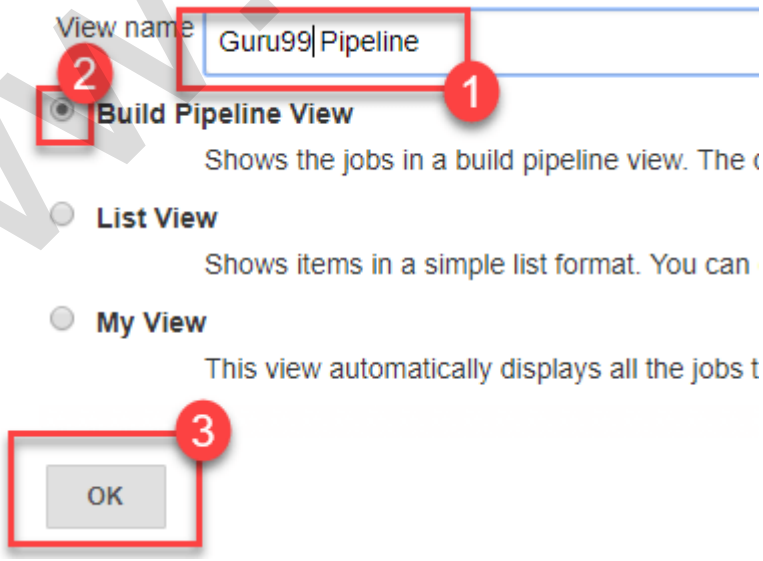
Once you are logged in to your Jenkins dashboard:

Step 1) Click on the "+" button on the left-hand side of your Jenkins dashboard to create a pipeline.



Step 2)

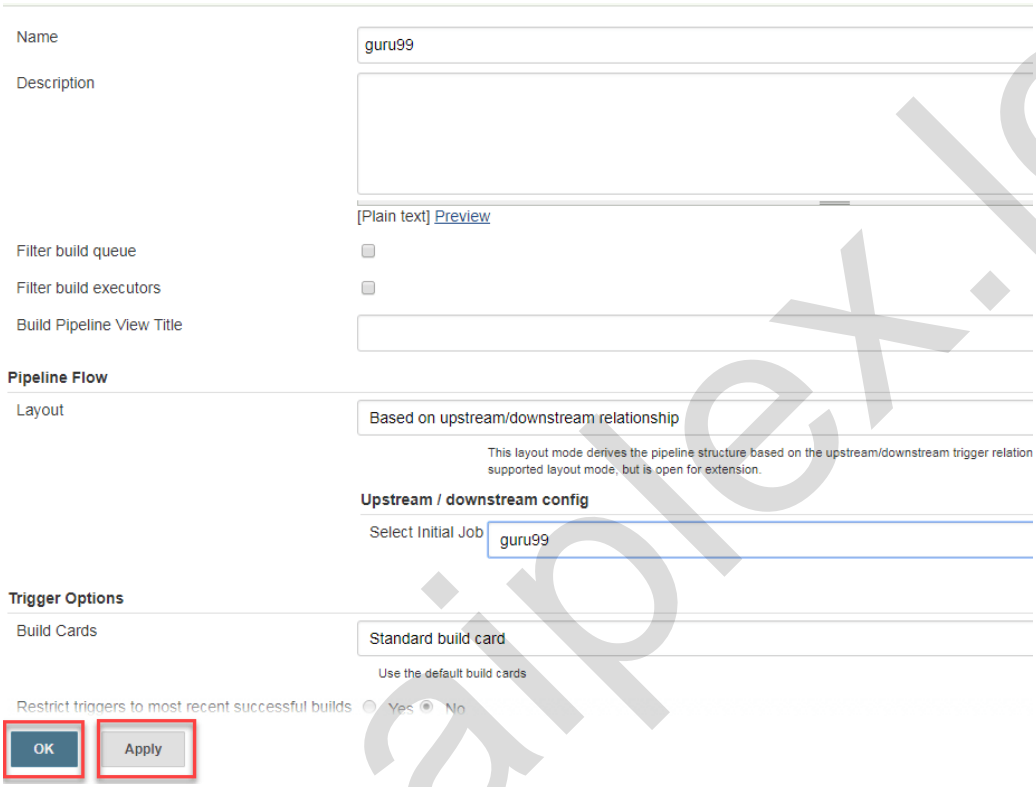
1. You will be asked to give a name to the pipeline view. We shall call it "**Guru99 Pipeline**" for the duration of this demo.
2. Select **Build a pipeline view** under **options**
3. Click **ok**



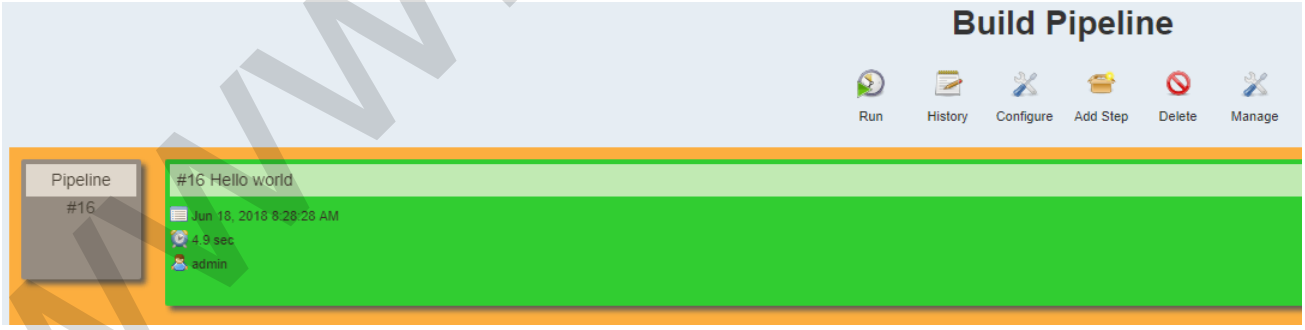
www.aiplex.lol

Step 3) In the next page, you will be asked for some more details to configure your Jenkins pipeline. Just accept the default settings, and make sure you choose the first job under the settings.

Click on **Apply** and then **OK**.



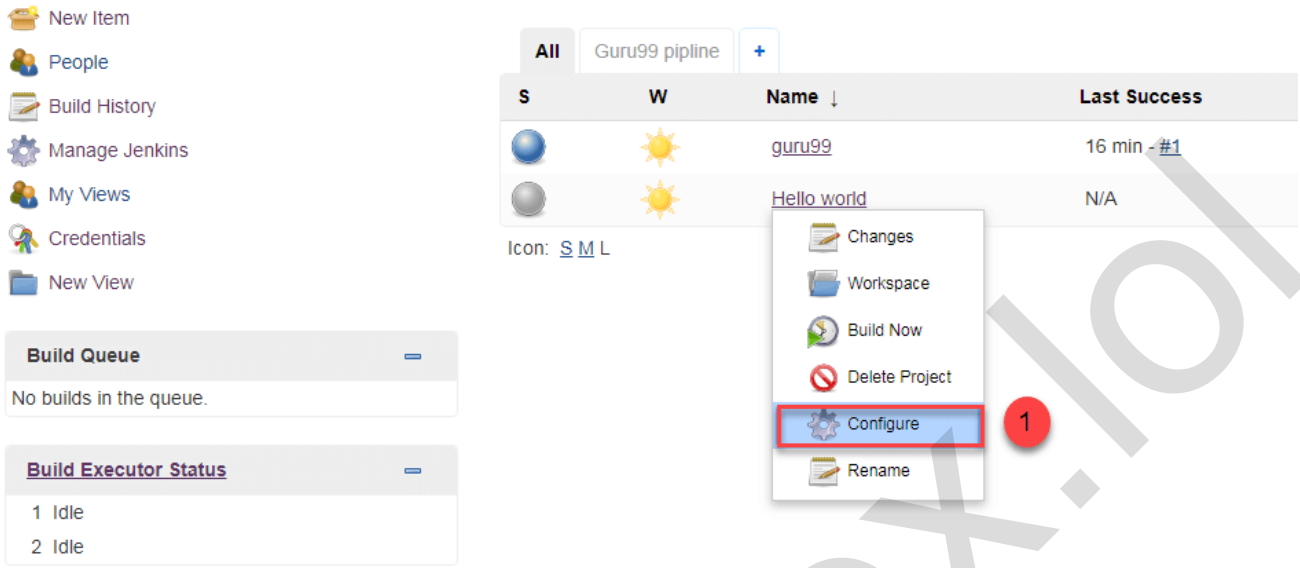
This will show you the sample pipeline view of your item, as given below:



Running a Pipeline build

Step 1) For running a pipeline build, you need to chain your jobs first. For this, go to your first job and click on configure.

www.aiplex.lol



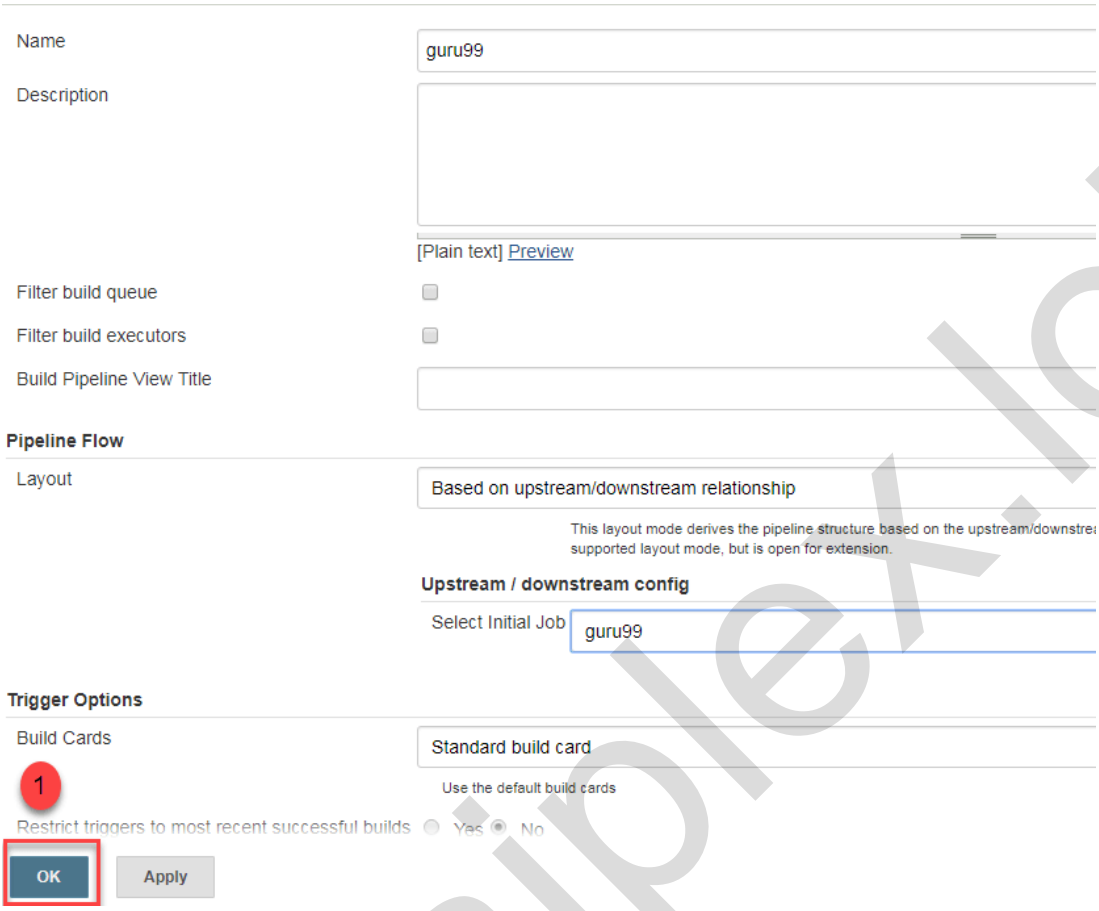
Step 2) Now, under **Build Triggers**, check the **Build after other projects are built** option.

Thus, a chain for all your jobs has been created.

Step 3) Install the **Build Pipeline view** plugin if you don't have it installed already.

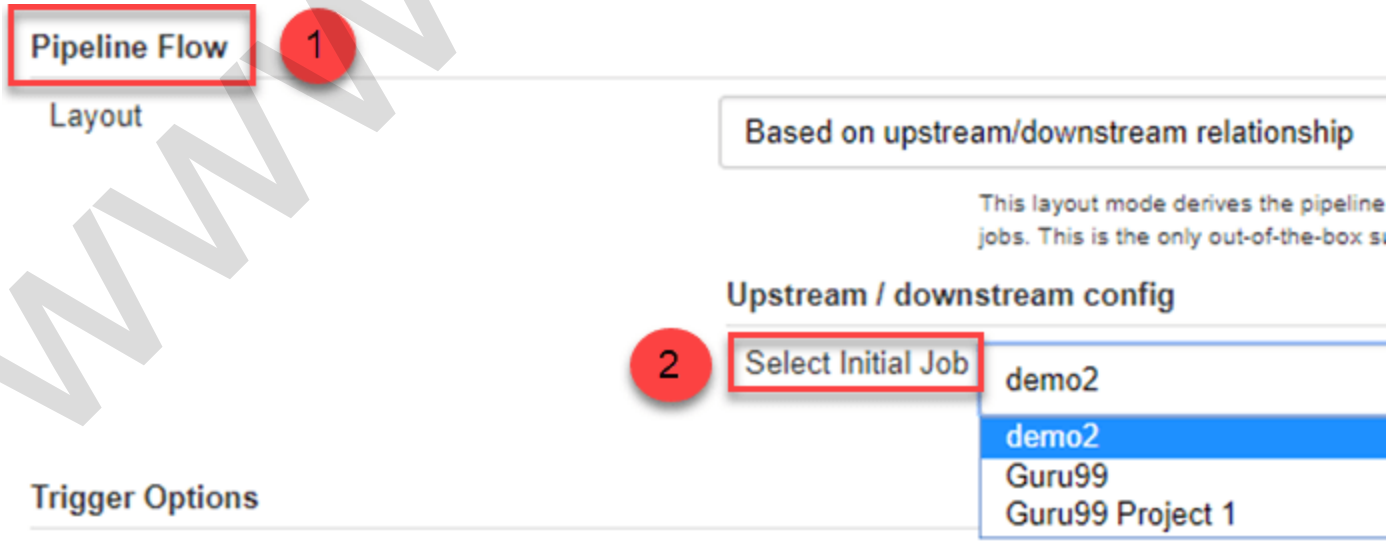
Step 4) Go to your Jenkins dashboard and create a view by clicking on the "+" button. Select the **Build Pipeline View** option and click **OK**.

www.aiplex.lol



Step 5) Under **Pipeline view configuration**, locate **Pipeline Flow**.

Under **Pipeline flow**, select the initial job to run. Now choose the job which has chains to other jobs, as shown in **Step 1** and **Step 2**.

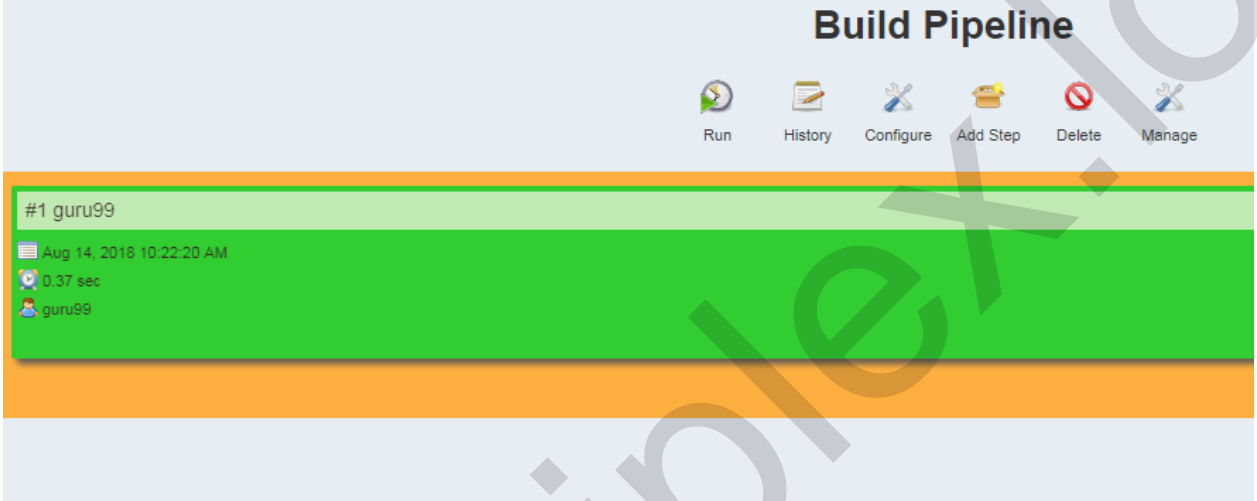


www.aiplex.lol

Here we have selected **Guru99 Project 1** as the initial job, chained to other jobs. So, one by one, the jobs will run in the pipeline.

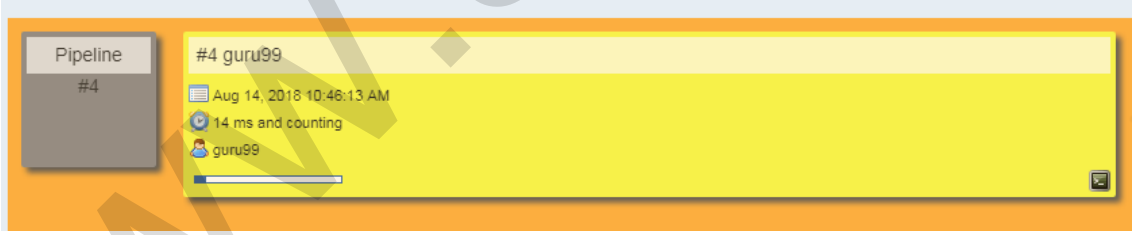
When the Jenkins pipeline is running, you can check its status with the help of Red and Green status symbols. Red means the pipeline has failed, while green indicates success.

In this Jenkins pipeline example, we see that the button is green. Hence, the pipeline is successful.



Running Jenkins pipeline

Click on **Run** to run the Jenkins pipeline. It will look something like this:



In the Jenkins pipeline script example above, we are demonstrating a simple "helloworld.java" program. But in real time projects, you will be responsible for creating and building complex pipelines in Jenkins. See below for a sample pipeline view.

www.aiplex.lol

Best Practices using Jenkins Pipeline:

- Use the genuine Jenkins Pipeline
- Develop your pipeline as code
- Any non-setup work in your pipeline should occur within a stage block.
- Any material work in a pipeline must be performed within a node block.
- Don't use input within a node block.
- Never set environment variables with env global variable
- Wrap your inputs in a timeout

www.aiplex.lol

Get more free courses at www.aiplex.lol

CI/CD Pipeline: Learn with Example

What is a CI/CD pipeline?

A CI/CD pipeline automates the process of software delivery. It builds code, runs tests, and helps you to safely deploy a new version of the software. CI/CD pipeline reduces manual errors, provides feedback to developers, and allows fast product iterations.

CI/CD pipeline introduces automation and continuous monitoring throughout the lifecycle of a software product. It involves from the integration and testing phase to delivery and deployment. These connected practices are referred as CI/CD pipeline.

What is Continuous Integration, Continuous Delivery, and Continuous Deployment?

- **Continuous integration** is a software development method where members of the team can integrate their work at least once a day. In this method, every integration is checked by an automated build to search the error.
- **Continuous delivery** is a software engineering method in which a team develops software products in a short cycle. It ensures that software can be easily released at any time.
- **Continuous deployment** is a software engineering process in which product functionalities are delivered using automatic deployment. It helps testers to validate whether the codebase changes are correct, and it is stable or not.

In this CI/CD Pipeline tutorial, you will learn:

- Stages of a CI/CD pipeline
- Example of CI/CD Pipeline
- CI/CD pipeline Best Practices
- Advantages of CI/CD pipelines
- Important CI/CD tools
- Why Does the CI/CD Pipeline Matter for IT Leaders?
- Ci/CD Pipeline KPI

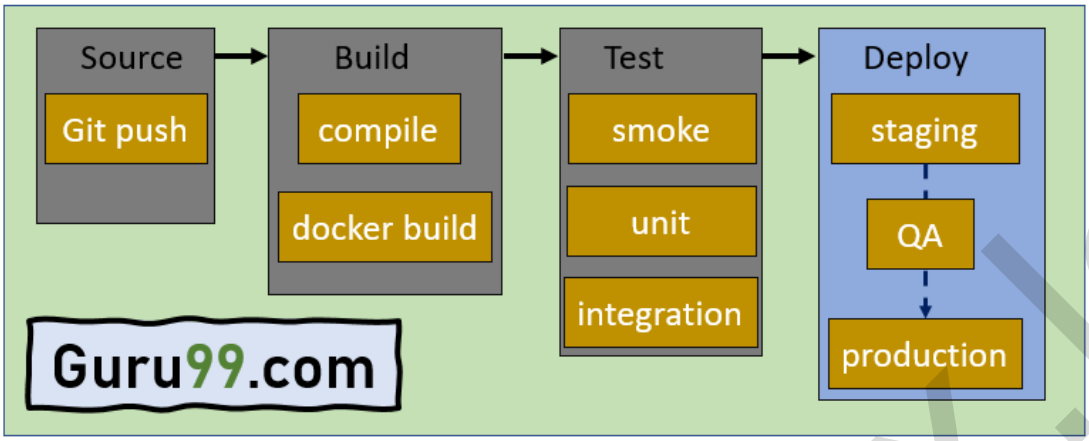
Stages of a CI/CD pipeline

A CI/CD pipeline is a runnable specification of the steps that any developer should perform to deliver a new version of any software. Failure in each and every stage triggers a notification via email, Slack, or other communication platforms. It enables responsible developers to know about the important issues.

www.aiplex.lol

Get more free courses at www.aiplex.lol

Here are the important Stages of CI/CD pipeline:



Stages of CI/CD pipeline

Source Stage

In the source stage, CI/CD pipeline is triggered by a code repository. Any change in the program triggers a notification to the CI/CD tool that runs an equivalent pipeline. Other common triggers include user-initiated workflows, automated schedules, and the results of other pipelines.

Build Stage

This is the second stage of the CI/CD Pipeline in which you merge the source code and its dependencies. It is done mainly to build a runnable instance of software that you can potentially ship to the end-user.

Programs that are written in languages like C++, Java, C, or Go language should be compiled. On the other hand, JavaScript, Python, and Ruby programs can work without the build stage.

Failure to pass the build stage means there is a fundamental project misconfiguration, so it is better that you address such issue immediately.

Test Stage

Test Stage includes the execution of automated tests to validate the correctness of code and the behaviour of the software. This stage prevents easily reproducible bugs from reaching the clients. It is the responsibility of developers to write automated tests.

Deploy Stage

This is the last stage where your product goes live. Once the build has successfully passed through all the required test scenarios, it is ready to deploy to live server.

www.aiplex.lol

Get more free courses at www.aiplex.lol

Example of CI/CD Pipeline

Here is example of CI/CD pipeline:

- **Source Code Control:** Host code on GitHub as a private repository. This will help you to integrate your application with major services and software.
- **Continuous integration:** Use continuous integration and delivery platform CircleCI and commit every code. When the changes notify, this tool will pull the code available in GitHub and process to build and run the test.
- **Deploy code to UAT:** Configure CitcleCI to deploy your code to AWS UAT server.
- **Deploy to production:** You have to reuse continuous integration steps for deploying code to UAT.

CI/CD pipeline Best Practices

Here is a CI/CD pipeline best practices:

- Write up the current development process therefore, you can know the procedures that require to change and one that can be easily automated.
- Start off with a small proof of project before going ahead and complete whole development process at once.
- Set up a pipeline with more than one stage in which fast fundamental tests run first.
- Start each workflow from the same, clean, and isolated environment.
- Run open source tools that cover everything from code style to security scanning.
- Setup a better code hub to continuously check the quality of your code by running the standard set of tests against every branch.
- Peer code review each pull request to solve a problem in a collaborative manner.
- You have to define success metrics before you start the transition to CD automation. This will help you to consistently analyze your software, developing progress help refining where needed.

Advantages of CI/CD pipelines

Here are the pros/ benefits of CI/CD Pipeline:

- Builds and testing can be easily performed manually.
- It can improve the consistency and quality of code.
- Improves flexibility and has the ability to ship new functionalities.
- CI/CD pipeline can streamline communication.
- It can automate the process of software delivery.
- Helps you to achieve faster customer feedback.
- CI/CD pipeline helps you to increase your product visibility.
- It enables you to remove manual errors.
- Reduces costs and labour.
- CI/CD pipelines can make the software development lifecycle faster.

www.aiplex.lol

Get more free courses at www.aiplex.lol

- It has automated pipeline deployment.
- A CD pipeline gives a rapid feedback loop starting from developer to client.
- Improves communications between organization employees.
- It enables developers to know which changes in the build can turn to the brokerage and to avoid them in the future.
- The automated tests, along with few manual test runs, help to fix any issues that may arise.

Important CI/CD tools

Here are the important CI/CD tools:

Jenkins

Jenkins is an open-source Continuous Integration server that helps to achieve the Continuous Integration process (and not only) in an automated fashion. Jenkins is free and is entirely written in Java. Jenkins is a widely used application around the world that has around 300k installations and growing day by day.

Features:

- Jenkin will build and test code many times during the day.
- Automated build and test process, saving timing, and reducing defects.
- The code is deployed after every successful build and test.
- The development cycle is fast.

Link: https://www.jenkins.io/download/

Bambo

Bamboo is a continuous integration build server that performs - automatic build, test, and releases in a single place. It works seamlessly with JIRA software and Bitbucket.

Features:

www.aiplex.lol

Get more free courses at www.aiplex.lol

- Run parallel batch tests
- Setting up Bamboo is pretty simple
- Per-environment permissions feature allows developers and QA to deploy to their environments
- Built-in Git branching and workflows. It automatically merges the branches.

Link: https://www.atlassian.com/software/bamboo

CircleCI

Circle CI is a flexible CI tool that runs in any environment like a cross-platform mobile app, Python API server, or Docker cluster. This tool reduces bugs and improves the quality of the application.

Features:

- Allows to select Build Environment
- Supports many languages including C++, JavaScript, NET, PHP, Python, and Ruby
- Support for Docker lets you configure a customized environment.
- Automatically cancel any queued or running builds when a newer build is triggered.

Link: https://circleci.com/

Why Does the CI/CD Pipeline Matter for IT Leaders?

- CI/CD pipeline can improve reliability.
- It makes IT team more attractive to developers.
- CI/CD pipeline helps IT leaders, to pull code from version control and execute software build.
- Helps to move code to target computing environment.
- Enables project leaders to easily manage environment variables and configure for the target environment.
- Project managers can publish push application components to services like web services, database services, API services, etc.
- Providing log data and alerts on the delivery state.
- It enables programmers to verify code changes before they move forward, reducing the chances of defects ending up in production.

www.aiplex.lol

Get more free courses at www.aiplex.lol

Ci/CD Pipeline KPI

- **Cycle or Deployment Time:** Cycle time is the time taken to go from the build stage to production. You can obtain average life cycle time by measuring the development process phases. This metric will give insight into bottlenecks in your process and the overall speed of development time.
- **Development Frequency:** Development frequency allows you to analyse bottlenecks you find during automation. The more frequent smaller releases reduce the risk of defects and fix them when found. Such a metric is an overall measure of your team efficiency.
- **Change Lead Time:** It measures the start time of the development phase to deployment. This metric is an indicator of the entire development process and how well the team works together.
- **Change Failure Rate:** It focuses on the number of times development get succeeds vs. the number of times it fails.
- **MTTR vs. MTTF:** MTTR (Mean Time to Recovery) is the amount of time required by your team to recover from failure. MTTF (Mean Time to Failure) measures the amount of time between fixes and outages. These metrics are a reflection of the team's ability to respond and fix issues.

Summary

- A CI/CD pipeline automates the process of software delivery.
- CI/CD pipeline introduces automation and continuous monitoring throughout the lifecycle of a software product.
- Continuous integration is a software development method where members of the team can integrate their work at least once a day.
- Continuous delivery is a software engineering method in which a team develops software products in a short cycle.
- Continuous deployment is a software engineering process in which product functionalities are delivered using automatic deployment.
- There are four stages of a CI/CD pipeline 1) Source Stage, 2) Build Stage, 3) Test Stage, 4) Deploy Stage.
- Important CI/CD tools are Jenkins, Bambo, and Circle CI.
- CI/CD pipeline can improve reliability.
- CI/CD pipeline makes IT team more attractive to developers.
- Cycle time is the time taken to go from the build stage to production.
- Development frequency allows you to analyse bottlenecks you find during automation.
- Change Lead Time measures the start time of the development phase to deployment.
- Change Failure Rate focuses on the number of times development get succeeds vs. the number of times it fails.
- MTTR (Mean Time to Recovery) is the amount of time required by your team to recover from failure.
- MTTF (Mean Time to Failure) measures the amount of time between fixes and outages.

www.aiplex.lol

Get more free courses at www.aiplex.lol

Jenkins vs Travis-CI: What is the difference?

What is CI?

Continuous integration is a software development method where members of the team can integrate their work at least once in a day. In this method, every integration is checked by an automated build to search the error. The CI concept was first introduced over two decades ago to avoid "integration hell," which happens when integration is put off till the end of a project.

In this tutorial, you will learn

- What is CI?
- What is Travis CI?
- What is Jenkins?
- What does Travis do?
- What did Jenkin do?
- Travis CI Features:
- Jenkin Features:
- Travis vs. Jenkins
- Popularity Index
- Which is better?

How CI works?

- Developers write code and commit changes to the shared repository
- After that, the CI server monitors the repository and evaluates all the changes
- CI builds the system and conduct integration and unit tests
- The server releases deployable artifacts
- The Continous integration server assigns a build tag to the version and building code
- Then the CI server reports the team about the successful build. If the tests fail, the server alerts about the event to the development team. The team will fix the issues as fast as is possible.

www.aiplex.lol

Get more free courses at www.aiplex.lol

VS

KEY DIFFERENCE

- Travis CI is a commercial CI tool whereas Jenkins is an open-source tool.
- Travis CI takes very less time to get started while Jenkins needs elaborate setup.
- Travis CI offers less customization option whereas Jenkins offers vast customization options.
- Travis CI has a YAML configuration file whereas Jenkins provides a full configuration option to the user.

What is Travis CI?

Travis CI was the first CI as a Service tool. It introduced a new approach to building code in the cloud. This CI tool allows the user to sign up, link their repository, build, as well as test their apps.

Travis CI tool can easily integrate with the common cloud repositories like GitHub and Bitbucket. It offers many automated CI options which cut out the need for a dedicated server as the Travis CI server is hosted in the cloud. This allows you to test in different environments, on various machines, running on different Operating Systems.

Travis CI is free for open source projects. For commercial projects, you need to purchase an enterprise plan.

What is Jenkins?

Jenkins is an award-winning continuous integration tool that monitors executions of deployment cycles. It started as a side project by Sun's software engineers group. Later it was expanded as one of the popular open source CI tools which help software development teams to automate their deployments.

www.aiplex.lol

Get more free courses at www.aiplex.lol

Jenkins is a Java-based tool, which means you only need Java Runtime Environment to operate it. Hence, Jenkins can be installed on any operating system where Java runs.

In this tool, Developers can also specify conditions for customized builds. Jenkins supports a massive plugin archive. This allows developers to alter how Jenkin looks and operates.

Moreover, the Jenkins Pipeline suite of plugins comes with special tools that allow developers to model easy-to-complex delivery pipelines using DSL (Digital Subscribe line) method.

What does Travis do?

Travis CI offers following benefits:

- You can monitor GitHub projects
- Runs Test and generate results quickly. Parallel test execution is possible.
- Build artifacts & check code quality
- Easy Deployment to cloud services
- It can identify small as well as large code changes.
- Developers can use Travis CI to watch the tests when they are running.
- The tool integrates with Slack, HipChat, Email, etc.

What does Jenkin do?

Jenkins allows you to automate your build, test, and deploy tasks. The tool provides support for different OS like Windows, Mac OSX, and Linux systems.

Moreover, Jenkins gives you an ability to quickly build and test your code to get early feedback on whether it's ready for production or not. In most cases, Jenkin will require few modifications according to your team's custom requirements.

Travis CI Features:

- Automatic integration with GitHub
- Repository access to build pull requests
- Support for 21 languages like Android, C, C#, C++, Java, JavaScript (with Node.js), Perl, PHP, Python, R, Ruby, etc
- Pre-installed build & test tools
- Available services - databases, message queues, etc.
- Deployment to multiple cloud services
- Encrypt secure environment variables or files
- Virtual machines recreated after every build
- CLI client and API for scripting
- Comes with free cloud-based hosting which does not require maintenance or administration.

www.aiplex.lol

Get more free courses at www.aiplex.lol

Jenkin Features:

- Easy to install, upgrade, and configure
- Distributed Builds
- Monitoring external jobs
- More than 600 plugins to customize your Jenkins environment
- Over 1000+ public repositories on Github, 500+ contributors, strong commit activity
- Support for various authentication methods, version control systems, notification, etc.
- Jenkins provides remote access API and its functionalities.
- Provide Powerful CI/CD tool for big projects
- It supports various job models like Freestyle, Pipeline, etc.,
- Allows developers to add their extensions
- Compatible with Docker, Libvirt, Kubernetes, and many other programs

Travis vs. Jenkins

Parameter	Jenkin	Travis
Cost	Jenkins is free. But development team need to run and maintain their dedicated server. This could be considered an extra expense.	Travis CI enterprise suites start at $129 per month. Cost increase based on the level of support you require.
Set up Time	Jenkins needs elaborate setup. So you'll have a very long wait time for the complete installation.	It takes very less time to get started. Create a config file and start integrating.
Performance	If you're looking for a CI tool with unlimited customization options, then Jenkins is the best choice for you.	Travis CI is the best choice If you are working in an open source project.
Tool Type	It is an open-source free to use the tool.	It is a commercial CI Tool
Usage	Easy to use	Flexible to use
Github	Good for Github	Excellent for Github

www.aiplex.lol

Get more free courses at www.aiplex.lol

Parameter	Jenkin	Travis
Support	Extensive support from the community.	Limited support for the community.
Pros	• Customization is biggest benefit of the Jenkins CI server • An extensive plugin archive can be configured to change the look of Jenkins. • You can also add new functionality like authentication, alerts, and credentials.	• Integration with GitHub & cloud • Unlimited open source projects with full functionality • Extensive project configuration via .travis.ymi file • Allows cluster tests and run them in parallel • Multiple build environments and target platforms (i.e. Node 0.10,0.8,0.6, Li on).
Cons	• One major drawback of Jenkins for continuous integration is that it is highly configurable. There is no out-of-the-box setup. That's why it may take two-three hours to days to get everything	• The biggest cons of installing Travis CI is that it's Commercial plans start at $129/m which is quite expensive. • Not suitable for high-security projects • Unlike other CI tools, it does not

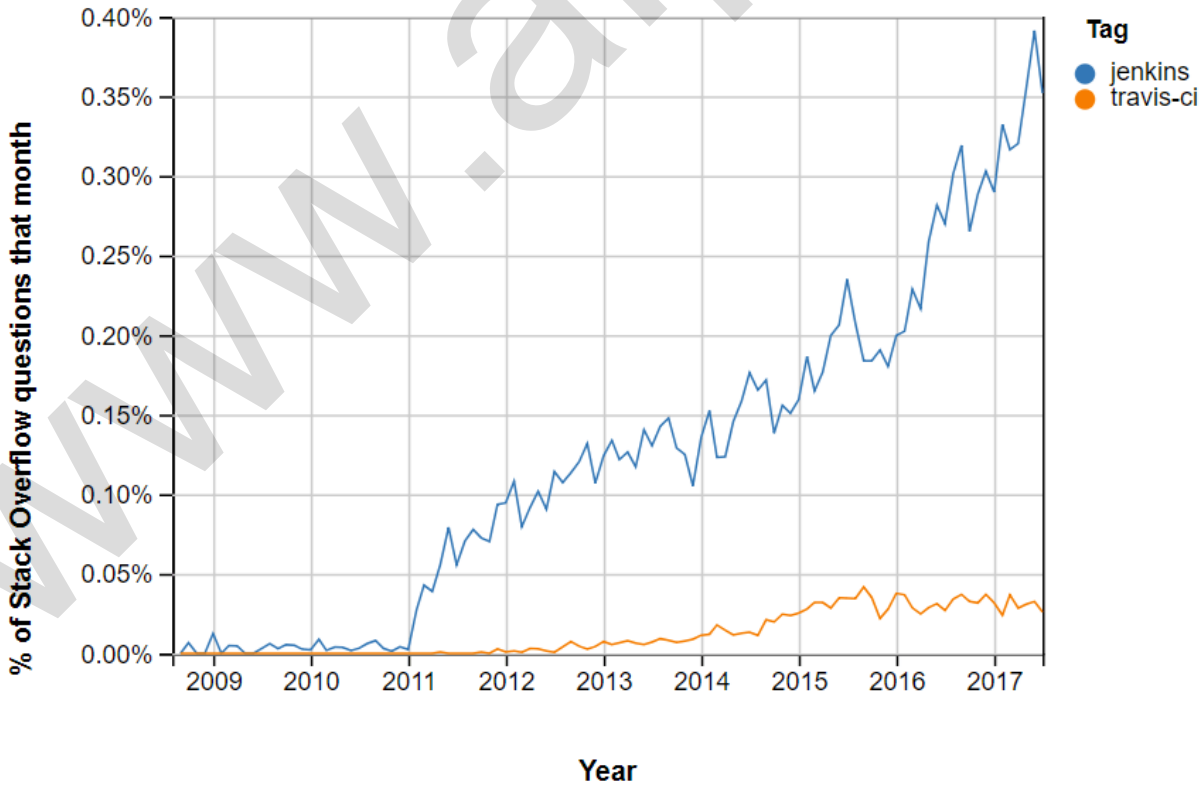
www.aiplex.lol

Get more free courses at www.aiplex.lol

Parameter	Jenkin	Travis
	configured in the system.	offer Bitbucket Support.
Usage Plans	Free	Free for open source projects. However, Paid for Enterprise.
Server Machine	Server-based	Cloud-based
Customization Options	More	Less
Configuration	Fully customizable	YAML
Control on system	Full	Very less

Popularity Index

The number of questions labeled Jenkins and Travis in Stack Overflow.



www.aiplex.lol

Get more free courses at www.aiplex.lol

Which is better?

Thus, with the above discussion, we can get ay that Travis and Jenkins both offer wonderful features. However, small open source projects are best suited for Travis CI as it is easy to run and quick to set up. On the other hand, large enterprise is best suited to Jenkins as it offers free licensing for a private project and a wide range of customizable feature. So, we can say that both of these continuous integration tools are good in their way.

www.aiplex.lol

Get more free courses at www.aiplex.lol

Continuous Integration vs Continuous Delivery vs Continuous Deployment

What is Continuous Integration?

Continuous integration is a software development method where members of the team can integrate their work at least once a day. In this method, every integration is checked by an automated build to search the error.

In continuous integration after a code commit, the software is built and tested immediately. In a large project with many developers, commits are made many times during a day. With each commit code is built and tested. If the test is passed, build is tested for Deployment. If the Deployment is a success, the code is pushed to production. This commit, build, test, and deploy is a continuous process, and hence the name continuous integration/deployment.

What is Continuous Delivery?

Continuous delivery is a software engineering method in which a team develops software products in a short cycle. It ensures that software can be easily released at any time.

The main aim of continuous delivery is to build, test, and release software with good speed and frequency. It helps you to reduce the cost time and risk of delivering changes by allowing for frequent updates in production.

What is Continuous Deployment

Continuous deployment is a software engineering process in which product functionalities are delivered using automatic deployment. It helps testers to validate whether the codebase changes are correct and stable or not.

The team can achieve continuous deployment by relying on infrastructure that automates different testing steps. Once each integration meets this release criteria, the application is updated with a new code.

www.aiplex.lol

Get more free courses at www.aiplex.lol

KEY DIFFERENCES:

- CI is an approach of testing each change to codebase automatically whereas Continuous Delivery is an approach to obtain changes of new features, configuration, and bug fixes. On the other hand, Continuous Deployment is an approach to develop software in a short cycle.
- CI is performed immediately after the developer checks- in. While in Continuous Delivery, developed code is continuously delivered until the programmer considers it is ready to ship and in Continuous Deployment, developers deploy the code directly to the production stage when it is developed.
- CI uses unit tests on the contrary Continuous Delivery uses business logic tests. In Continuous Deployment any testing strategy is used.
- CI refers to the versioning of source code whereas Continuous Delivery refers to the logical evolution of CI and Continuous Deployment refers to automated implementations of the source code.

Difference between CI vs CD vs CD

Here is an important difference between CI vs CD vs CD.

Continuous integration	**Continuous Delivery**	**Continuous Deployment**
CI is an approach of testing each change to codebase automatically.	CD is an approach to obtain changes of new features, configuration, and bug fixes.	CD is an approach to develop software in a short cycle.
CI refers to the versioning of source code.	CD refers to the logical evolution of CI.	CD refers to automated implementations of the source code.
CI focuses on automation testing to determine that the software has no errors or bugs.	Focuses on releasing new changes to your clients properly.	Emphasis on the change in all stages of your production pipeline.
CI is performed immediately after the	In CD, developed code is continuously	In CD, developers deploy the code directly to the

www.aiplex.lol

Get more free courses at www.aiplex.lol

Continuous integration	Continuous Delivery	Continuous Deployment
developer checks-in.	delivered until the programmer considers it is ready to ship.	production stage when it is developed.
It helps you to identify and rectify issues early.	It allows developers to check software updates.	It enables you to rapidly deploy and validate new features and ideas.
It uses unit tests.	It uses business logic tests.	Any testing strategy is performed.
Development team sends continuous code merging requests even when the testing process is running.	You deliver code for review that can be batched for release.	Deploy code using an automated process.
You require a continuous integration server to monitor the main repository.	You require a strong foundation in continuous integration.	You need a good testing culture.

Advantages of Continuous Integration

Here are the pros/benefits of continuous integration:

- Helps you to build better quality software
- It enables you to conduct repeatable testing.
- CI allows software developers to work independently on features in parallel.
- It can increase visibility and enable greater communication.
- CI process help to scale up Headcount and delivery output of engineering teams.
- Continuous integration helps you to develop a potentially shippable product for a fully automated build.
- Helps you to redue risks by making deployment faster and more predictable
- immediate feedback when an issue arrives.

www.aiplex.lol

Get more free courses at www.aiplex.lol

- Avoid last-minute confusion at the release date, and timing automates the build.
- It reduces risks and makes the deployment process more predictable.
- CI provides instant feedback when there is an issue.
- You can see the integration process in real time.
- It can avoid last-minute hassle at release dates.
- The current build is available constantly.
- Provides shippable products on a regular base.
- It is relatively easy to find a history of the software build.
- CI offers code stability.

Advantages of Continuous Delivery

Here are the pros/benefits of continuous delivery:

- Automate the software release process for making delivery more efficient, rapid, and secure.
- CD practices increase productivity by freeing developers from manual work and complex dependencies.
- It helps you to discover software bugs early in the delivery process.
- CD helps your business team to deliver updates to clients immediately and frequently.
- It ensures the software is always ready to go to production.
- You can release software more frequently, which helps you to get fast feedback from your clients.
- There is less pressure on decisions for small changes.

Advantages of Continuous Deployment

Here are the pros/benefits of continuous Deployment:

- It helps you to automate the repetitive tasks.
- CD makes your deployment flawless without compromising security.
- Easily scale from a single software application to an enterprise IT portfolio.
- You can ship cloud-native as well as traditional applications.
- It gives a single view across all environments and applications.
- You can connect your existing DevOps tools and scripts into a proper workflow.
- CD enables you to increase overall productivity.
- You can integrate processes and teams with a unified pipeline.

Disadvantages of Continuous Integration

Here are the cons/ disadvantages of continuous integration:

- Initial setup time and training is required to get acquainted with CI server
- Well-developed test-suite required many resources for CI server.
- It requires additional servers and environments.
- You need a conversion of familiar processes in one project.

www.aiplex.lol

Get more free courses at www.aiplex.lol

- It goes for waiting when multiple developers integrate their code around the same time.
- Your team should write automated tests for each and every new feature or bug fix.
- You require a CI server that monitors the main repository and run the tests for new code commits.
- Developers should merge their changes as more often as possible.
- The unit testing procedure should pass for the Deployment.

Disadvantages of Continuous Delivery

Here are the cons/disadvantages of continuous delivery:

- You should know continuous integration practices before you go for continuous delivery.
- Deployments are still manual, and hence it takes lots of time to deliver the software product.
- The automated tests should be written and function properly.
- Faulty tests can lead to damage while quality testing.
- It requires team coordination because code changes should be collected regularly in an efficient way.
- Continuous delivery requires a reliable and strong integration server for the automation test that is costly.

Disadvantages of Continuous Deployment

Here are the cons/disadvantages of continuous Deployment:

- Your testing culture should be good as the quality of the suite determines how good software releases are.
- Documentation procedures need to keep up with deployment pace.
- Releasing significant changes needs assurance by marketing, help, and support, and other departments.

Continuous Integration Best Practices

Here are some important best practices while implementing Continuous Integration.

- Automate your software build.
- Keep the build as fast as possible.
- Every commit should result in a build
- Automate Deployment
- Commit early and often.
- You should never commit broken code
- Fix build failures immediately.
- Build-in every target environment Create artifacts from every build
- The build of the software needs to be carried out in a manner so that it can be automated
- Do not depend on an IDE
- Build and test everything when it changes

www.aiplex.lol

Get more free courses at www.aiplex.lol

- The database schema counts as everything
- Helps you to find out key metrics and track them visually
- Check-in often and early.
- Stronger source code control.
- Continuous integration is running unit tests whenever you commit code.
- Automate the build and test everyone.
- Keep the build fast with automated Deployment.

Continuous Delivery Best Practices

Here are some important best practices while implementing continuous delivery:

- The first stage must be triggered upon every check-in.
- Each stage should trigger the next one quickly upon successful completion.
- Maintain the version of the source code.
- Perform automated build and Deployment.
- Deploy to one instance of a virtual machine at a time.
- Perform unit and integration tests.
- You have to build your library only once.
- The team should use the same automated release method for each and every environment.
- This method enables you to eliminate conflicts and last-minute problems.
- In case any state fails, you should automatically pause the process and fix the issues.

Continuous Deployment Best Practices

Here are some important best practices while implementing continuous Deployment:

- You should use an issue tracker for the development task.
- In your version controlling system, you should create a branch that contains the issue number and description of any change you have made.
- When software is ready for the Deployment, you can create a pull request for the branch.
- Deployment software to pre-production staging servers.
- Promote your software once you are happy with its quality.

Challenges of Continuous Integration

Here are the challenges of continuous integration:

- It makes the developing process slow.
- Exposes problems and sharing of issues.
- It may lead to a lack of maintenance of version control.
- It can force you to deal with problems.
- Difficulty in building automated code repository.
- Untested or broken code must not be committed.

www.aiplex.lol

Get more free courses at www.aiplex.lol

Challenges of Continuous Delivery

Here are the challenges of continuous delivery:

- You need to keep the continuous delivery efficient without bothering the time.
- You need to cope up with tight deadlines release plan.
- Poor product-specific communication of teams may lead to revisions as well as deployment delays.
- The business team should have the budget to have the infrastructure needed to build more impressive software.
- Monitoring data/ information should be utilized by the research and development team.
- The organization should ensure that how open source software fits into the current workflow.

Challenges of Continuous Deployment

Here are the challenges of continuous deployment:

- CD requires continuous planning to achieve frequent and fast releases.

- Ensure the alignment between the requirement of the business context and application development.
- Rapid delivery must not be isolated to the software development process alone.
- The flow should go with the overall software development cycle.
- Experimental results must be continuously linked with the software roadmap.

www.aiplex.lol

Get more free courses at www.aiplex.lol

Top 12 JENKINS Interview Questions & Answers

Download PDF

1) Mention what is Jenkins?

Jenkins is an open source tool with plugin built for continuous integration purpose. The principle functionality of Jenkins is to keep a track of version control system and to initiate and monitor a build system if changes occur. It monitors the whole process and provides reports and notifications to alert.

2) Explain what is continuous integration?

In software development, when multiple developers or teams are working on different segments of same web application, we need to perform integration test by integrating all modules. In order to do that an automated process for each piece of code is performed on daily bases so that all your code get tested.

3) What is the requirement for using Jenkins?

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

4) Mention what are the advantages of Jenkins?

Advantage of Jenkins include

- At integration stage, build failures are cached
- For each code commit changes an automatic build report notification generates
- 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

5) Explain how you can move or copy Jenkins from one server to another?

- Slide a job from one installation of Jenkins to another by copying the related job directory
- Make a copy of an already existing job by making clone of a job directory by a different name
- Renaming an existing job by renaming a directory.

www.aiplex.lol

Get more free courses at www.aiplex.lol

6) Mention what are the commands you can use to start Jenkins manually?

To start Jenkins manually, you can use either of the following

- (Jenkins_url)/restart: Forces a restart without waiting for builds to complete
- (Jenkin_url)/safeRestart: Allows all running builds to complete

7) Mention some of the useful plugins in Jenkin?

Some of the important plugins in Jenkin includes

- Maven 2 project
- Amazon EC2
- HTML publisher
- Copy artifact
- Join
- Green Balls

8) Explain how you can deploy a custom build of a core plugin?

To deploy a custom field of a core plugin, you have to do following things

- Stop Jenkins
- Copy the custom HPI to $Jenkins_Home/plugins
- Delete the previously expanded plugin directory
- Make an empty file called <plugin>.hpi.pinned
- Start Jenkins

9) Explain how can create a backup and copy files in Jenkins?

Jenkins saves all the setting, build artifacts and logs in its home directory, 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.

www.aiplex.lol

Get more free courses at www.aiplex.lol

10) Explain how you can clone a Git repository via Jenkins?

To clone a Git repository via Jenkins, you have to enter the e-mail and user name for your Jenkins system. For that, you have to switch into your job directory and execute the "git config" command.

11) Explain how you can set up Jenkins job?

To create a project that is handled via jobs in Jenkins. Select New item from the menu, once this done enter a name for the job and select free-style job. Then click OK to create new job in Jenkins. The next page enables you to configure your job.

12) Mention what are the two components Jenkins is mainly integrated with?

Jenkin is mainly integrated with two components

- Version Control system like GIT, SVN
- And build tools like Apache Maven.

www.aiplex.lol

Get more free courses at www.aiplex.lol

www.aiplex.lol

Get more free courses at www.aiplex.lol