What is jenkins?

Jenkins is defined as an open-source Java-based program comprising an automation server to enable continuous integration and continuous delivery (CI/CD), automating the various stages of software development such as test, build, and deployment*.*

Pipeline in relation to CI/CD:

A pipeline in the context of CI/CD is a set of processes or stages through which an application or software artifact progresses from development to production. It defines the steps, such as building, testing, and deploying, required to deliver software reliably and efficiently.

Purpose of a continuous integration tool:

The purpose of a continuous integration tool like Jenkins is to automate the integration of code changes from multiple developers into a shared repository. It helps identify and address integration issues early on by continuously building, testing, and validating the codebase, ensuring that changes are compatible with the rest of the system.

Capabilities of Jenkins:

- Automated builds: Jenkins can trigger builds automatically whenever changes are pushed to the repository.

- Testing and reporting: It supports various testing frameworks and generates reports to provide insights into test results.

- Continuous delivery/deployment: Jenkins enables the automated deployment of applications to different environments, such as staging or production.

- Integration with version control systems: It integrates seamlessly with popular version control systems like Git, allowing for easy tracking of code changes.

- Plugin ecosystem: Jenkins has a vast plugin ecosystem, providing extensibility and integration with various tools and technologies.

Custom use cases and where it performs best:
   - Web application development: Jenkins is commonly used for web application development, automating build, test, and deployment processes.
   - DevOps and CI/CD pipelines: It excels in implementing and managing complex CI/CD pipelines, enabling teams to automate the delivery of software changes.
   - Scalable and distributed environments: Jenkins can handle large-scale projects and supports distributed builds across multiple machines or agents.

Comparison with other CI tools:
   - Travis CI: While Jenkins is a highly customizable and versatile CI tool, Travis CI focuses on simplicity and ease of use. Travis CI provides a cloud-based solution, reducing the need for manual setup and configuration.
   - Bamboo: Compared to Jenkins, Bamboo is a commercial CI/CD tool developed by Atlassian. Bamboo offers a more integrated and streamlined experience for teams already using Atlassian products like Jira and Bitbucket. Jenkins, being open-source, provides more flexibility and community support.

Feature	Jenkins	GitLab	CircleCI	Semaphore
Pricing	Open source	Free for open source projects, paid plans for businesses	Paid plans	Paid plans
Platforms	Windows, Linux, macOS	Windows, Linux, macOS	Linux, macOS, Windows, Docker	Linux, macOS, Windows, Docker
Languages	Java, Python, Ruby, PHP, etc.	Java, Python, Ruby, PHP, etc.	Java, Python, Ruby, PHP, etc.	Java, Python, Ruby, PHP, etc.
Plugins	Large plugin ecosystem	Large plugin ecosystem	Smaller plugin ecosystem	Smaller plugin ecosystem
Reporting	Comprehensive reporting	Comprehensive reporting	Comprehensive reporting	Comprehensive reporting
Community	Active community	Active community	Active community	Active community
Ease of use	Can be complex to set up and configure	Easy to use	Easy to use	Easy to use
Customization	Highly customizable	Highly customizable	Less customizable	Less customizable
Deployment automation	Supports a wide range of deployment tools	Supports a wide range of deployment tools	Supports a wide range of deployment tools	Supports a wide range of deployment tools
Test automation	Supports a wide range of testing frameworks	Supports a wide range of testing frameworks	Supports a wide range of testing frameworks	Supports a wide range of testing frameworks

JENKINS SET UP

- Install java jdk version 17 from **https://www.oracle.com/java/technologies/downloads/#java17**

- Configure path under environment variables

- Download Jenkins stable version from https://www.jenkins.io/download/

● Install Jenkins with the following steps below

Jenkins 2.263.4 Setup

Destination Folder
Click Next to install to the default folder or click Change to choose another.

Jenkins

Install Jenkins 2.263.4 to:

C:\Program Files\Jenkins\

Change...

Back   Next   Cancel

Jenkins 2.263.4 Setup

Port Selection

Choose a port for the service.

Jenkins

Please choose a port.

Port Number (1-65535):

8080

Test Port ✔

It is recommended that you accept the selected default port.

Back   Next   Cancel

Jenkins 2.263.4 Setup

Jenkins 2.263.4 Setup

Select Java home directory (JDK or JRE)

Jenkins

Please select the path of a Java Development Kit or Java Runtime Environment. Only Java 1.8 and 11 are supported by Jenkins.

C:\Program Files\Java\jdk1.8.0_261\

Change...

Back   Next   Cancel

Jenkins 2.263.4 Setup — □ ✕

Ready to install Jenkins 2.263.4

Jenkins

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.
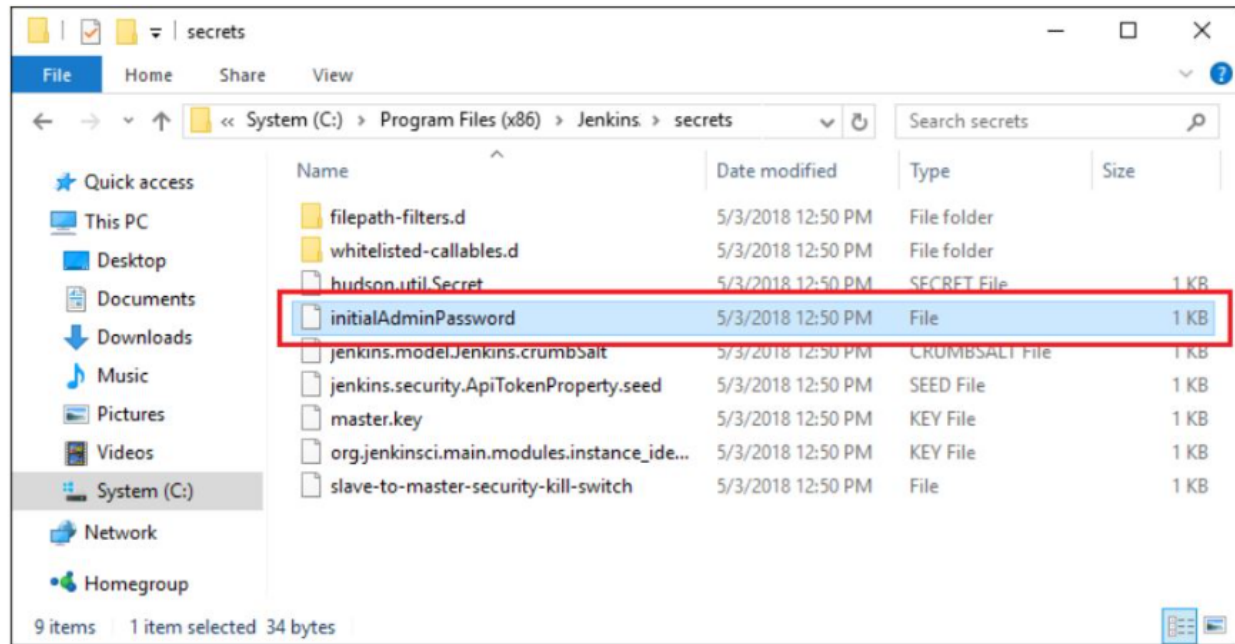
Back   🛡 Install   Cancel

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`C:\Program Files (x86)\Jenkins\secrets\initialAdminPassword`

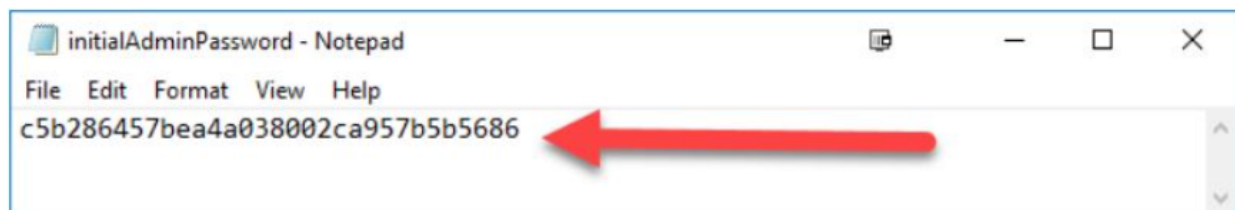Please copy the password from either location and paste it below.

Administrator password

[   ]

Continue

Step 3

Open the highlighted file and copy the content of the **initialAdminPassword** file.

- When the **Create First Admin User** page appears, specify the details for your administrator user in the respective fields and click Save and Finish.

- When the Jenkins is ready page appears, click Start using Jenkins and type in your browser localhost:8080 to access your dashboard.