



Continuous integration / Continuous delivery (deployment). Lecture 1.



Prerequisites

- Software development methodologies and techniques
- Virtualization and containerization
- Version Control Systems
- Operating systems administration
- Networking

Continuous integration

```
IF  
IT_business.bloods == time && money  
THEN  
IT_business.heart_pumping = CI / CD
```

“CI/CD bridges the gaps between development and operation activities and teams by enforcing automation in building, testing and deployment of applications. DevOps practices involve continuous development, continuous testing, continuous integration, continuous deployment and continuous monitoring of software applications throughout its development life cycle. The CI/CD practice, or CI/CD pipeline, forms the backbone of DevOps operations.”

“**DevOps** is a set of practices that combines **software development** (*Dev*) and **IT operations** (*Ops*). It aims to shorten the **systems development life cycle** and provide **continuous delivery** with high **software quality**. DevOps is complementary with **Agile software development**. Several DevOps aspects came from the Agile methodology.”

Software Development Life Cycle

1. Identify the Current Problems (Requirement Analysis)

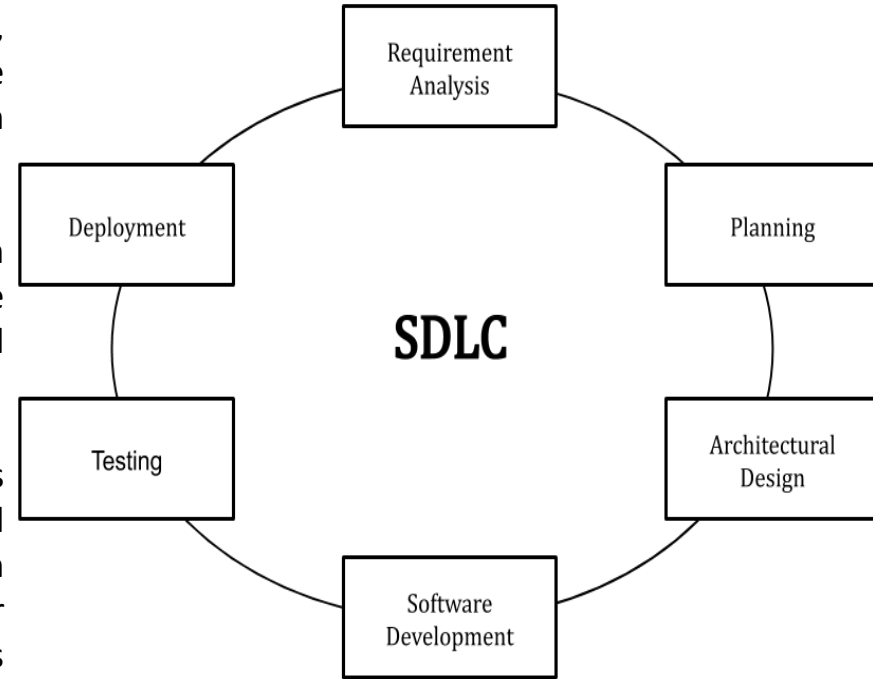
“What are the current problems?” This stage of the SDLC means getting input from all stakeholders, including customers, salespeople, industry experts, and programmers. Learn the strengths and weaknesses of the current system with improvement as the goal.

2. Plan

“What do we want?” In this stage of the SDLC, the team determines the cost and resources required for implementing the analyzed requirements. It also details the risks involved and provides sub-plans for softening those risks.

3. Design

“How will we get what we want?” This phase of the SDLC starts by turning the software specifications into a design plan called the Design Specification. All stakeholders then review this plan and offer feedback and suggestions. It’s crucial to have a plan for collecting and incorporating stakeholder input into this document. Failure at this stage will almost certainly result in cost overruns at best and the total collapse of the project at worst.



<https://stackify.com/what-is-sdlc/>

Software Development Life Cycle

4. Build

“Let’s create what we want.” At this stage, the actual development starts. It’s important that every developer sticks to the agreed blueprint. Also, make sure you have proper guidelines in place about the code style and practices.

5. Code Test

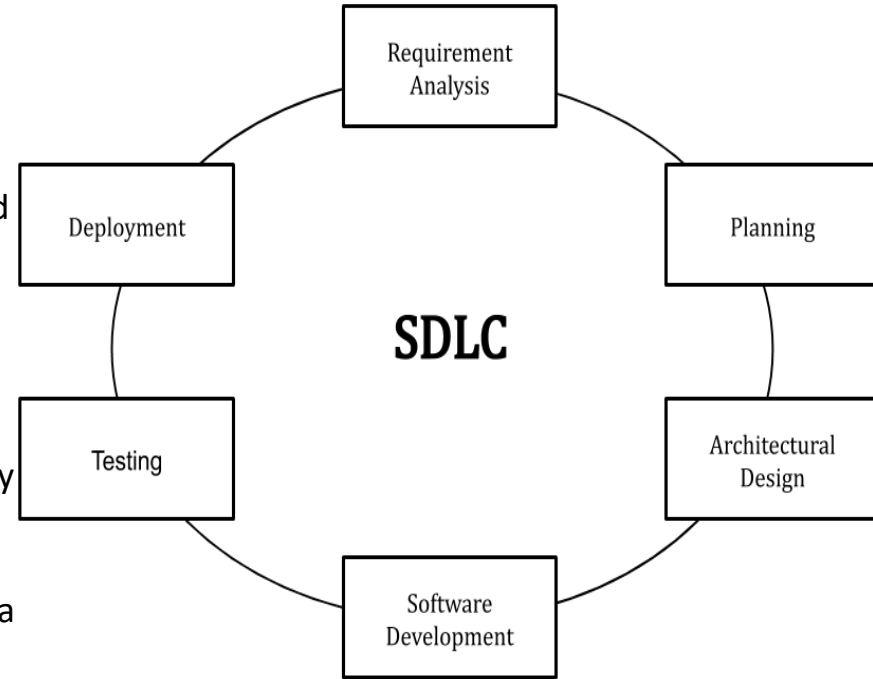
“Did we get what we want?” In this stage, we test for defects and deficiencies. We fix those issues until the product meets the original specifications.

In short, we want to verify if the code meets the defined requirements.

6. Software Deployment

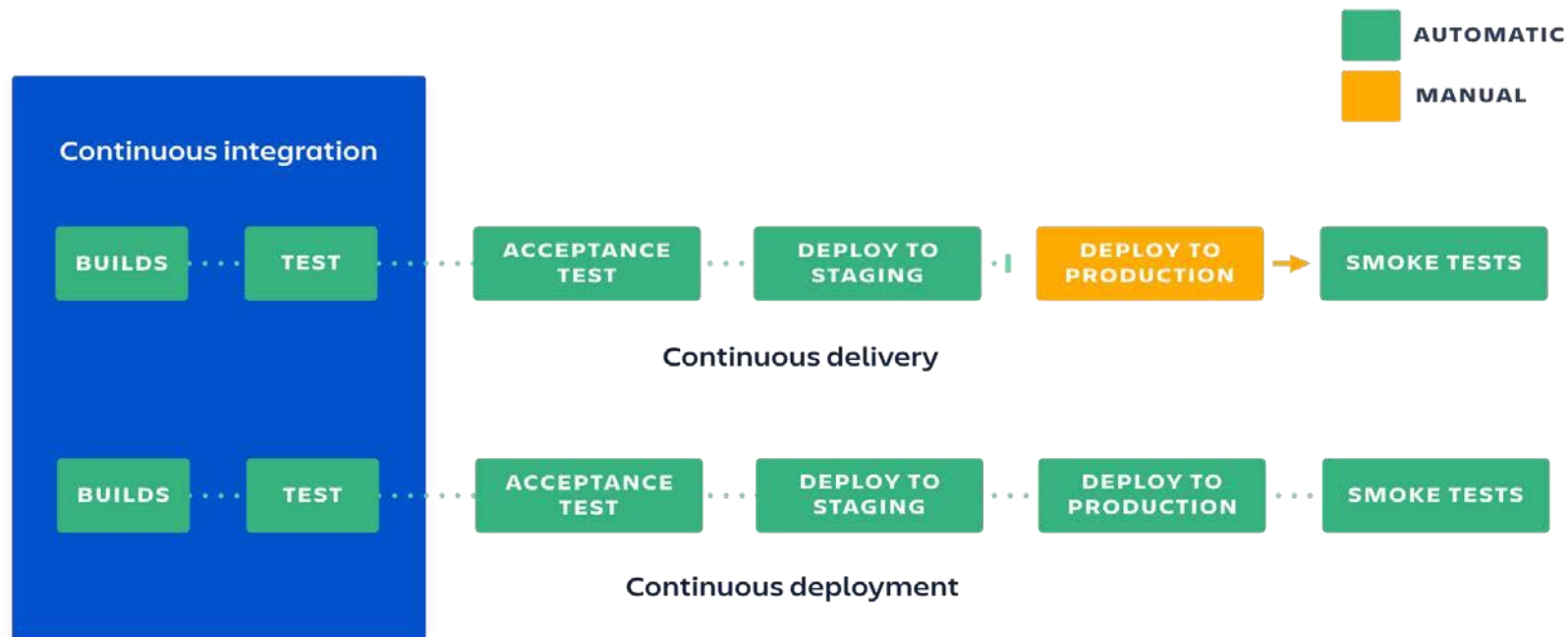
“Let’s start using what we got.” At this stage, the goal is to deploy the software to the production environment so users can start using the product. However, many organizations choose to move the product through different deployment environments such as a testing or staging environment.

This allows any stakeholders to safely play with the product before releasing it to the market. Besides, this allows any final mistakes to be caught before releasing the product.



<https://stackify.com/what-is-sdlc/>

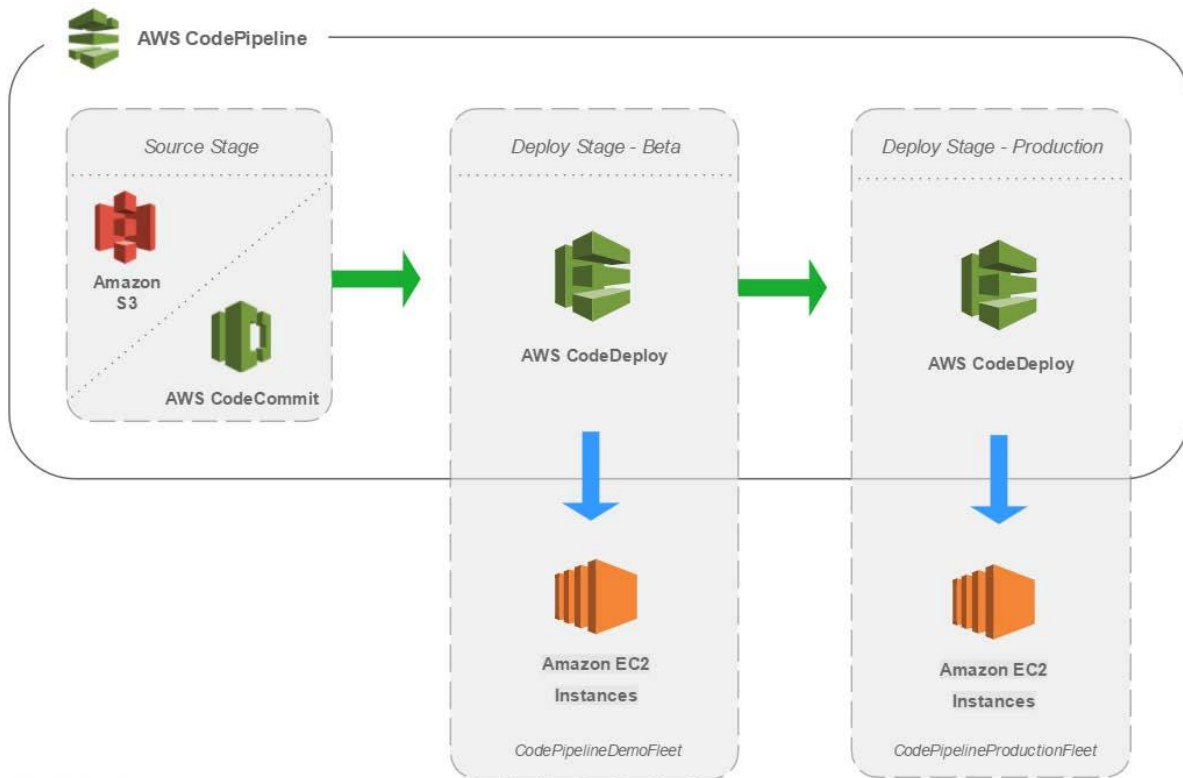
Atlassian vision



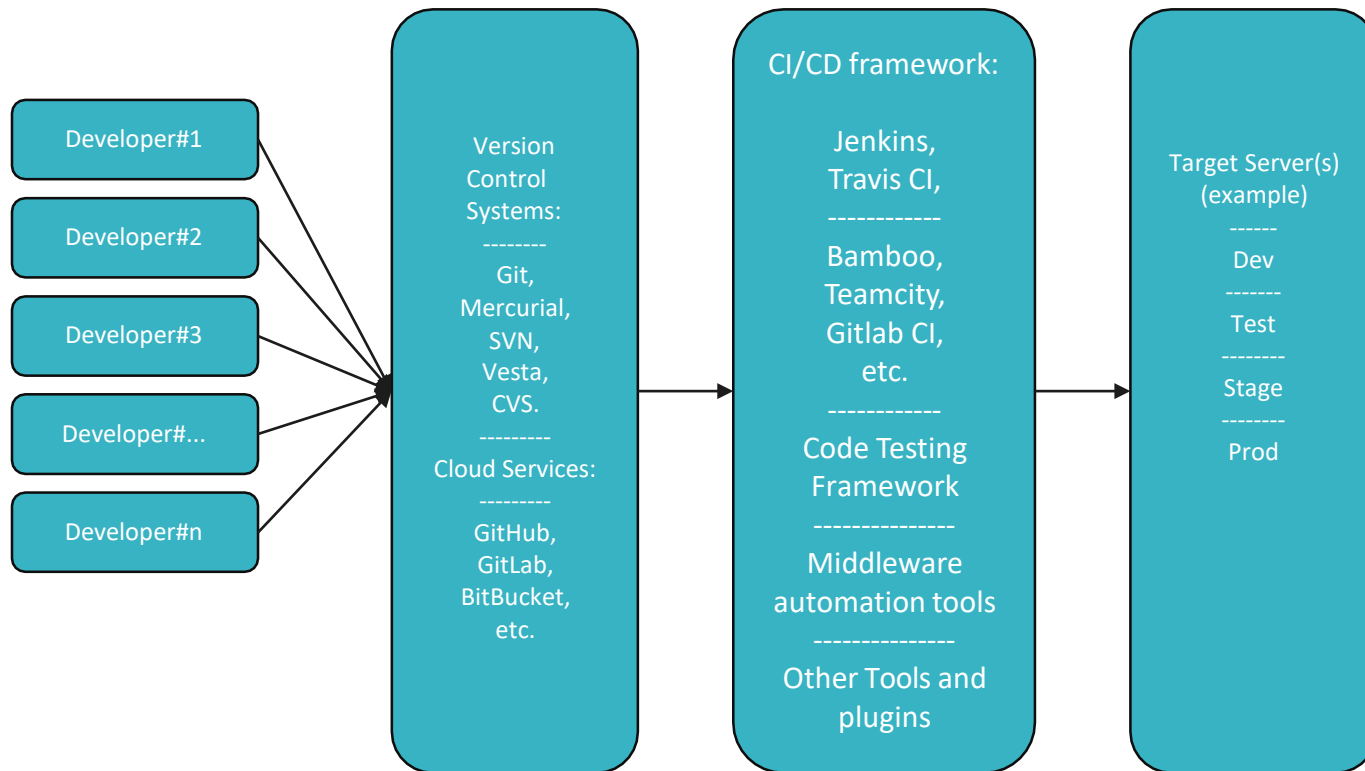
RedHat vision



AWS vision



CI/CD pipeline



Prerequisites

- VirtualBox ver. 6+
- Installed Ubuntu 20.04 with SSH server (server edition) with “Network”-> “Adapter” set to “Bridged Adapter”
- Clone(-s) of Installed Ubuntu 20.04 with SSH server (server edition) with “Network”-> “Adapter” set to “Bridged Adapter” (full clone with reinitialized MAC- addresses)

SSH client :

<https://mobaxterm.mobatek.net/download.html> (for Windows)

or

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html> (Windows, Linux, but simple)

or

• <https://www.termius.com/>

(Windows, Linux, Mac)

Installing Jenkins

Linux

Jenkins installers are available for several Linux distributions.

- [Debian/Ubuntu](#)
- [Fedora](#)
- [Red Hat / CentOS](#)

Prerequisites

Minimum hardware requirements:

- 256 MB of RAM
- 1 GB of drive space (although 10 GB is a recommended minimum if running Jenkins as a Docker container)

Recommended hardware configuration for a small team:

- 4 GB+ of RAM
- 50 GB+ of drive space

Comprehensive hardware recommendations:

- Hardware: see the [Hardware Recommendations](#) page

Software requirements:

- Java: see the [Java Requirements](#) page
- Web browser: see the [Web Browser Compatibility](#) page
- For Windows operating system: [Windows Support Policy](#)

<https://www.jenkins.io/doc/book/installing/linux/>

Jenkins (<https://jenkins.io/>)

- <https://jenkins.io/download/>



Downloading Jenkins

Jenkins is distributed as WAR files, native packages, installers, and Docker images. Follow these installation steps:

1. Before downloading, please take a moment to review the [Hardware and Software requirements](#) section of the User Handbook.
2. Select one of the packages below and follow the download instructions.
3. Once a Jenkins package has been downloaded, proceed to the [Installing Jenkins](#) section of the User Handbook.
4. You may also want to verify the package you downloaded. [Learn more about verifying Jenkins downloads.](#)

Download Jenkins 2.289.3 LTS for:

Generic Java package (.war)
SHA-256: 996df629d5f933546af9e9f77c29b371b6627b6265b6c9f134ac2a0f1248b87

Docker

Ubuntu/Debian

CentOS/Fedora/Red Hat

Windows

openSUSE

FreeBSD

Gentoo

macOS

OpenBSD

Download Jenkins 2.307 for:

Generic Java package (.war)
SHA-256: e1f73c76f6066db50eca64ac8aaed36d91c747219d35ef3653885dd8ed12c

Docker

Ubuntu/Debian

CentOS/Fedora/Red Hat

Windows

openSUSE

Arch Linux

FreeBSD

Gentoo

macOS

OpenBSD

OpenIndiana Hipster

Jenkins Debian packages

Jenkins Debian Packages

This is the Debian package repository of Jenkins to automate installation and upgrade. To use this repository, first add the key to your system:

```
wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
```

Then add a Jenkins apt repository entry:

```
sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
```

Update your local package index, then finally install Jenkins:

```
sudo apt-get update  
sudo apt-get install jenkins
```

The apt packages were signed using this key:

```
pub  rsa4096 2020-03-30 [SC] [expires: 2023-03-30]  
    62A9756BFD780C377CF24BA8FCE32E745F2C3D5  
uid                Jenkins Project  
sub  rsa4096 2020-03-30 [E] [expires: 2023-03-30]
```

You will need to explicitly install a Java runtime environment, because Oracle's Java RPMs are incorrect and fail to register as providing a java dependency. Thus, adding an explicit dependency requirement on Java would force installation of the OpenJDK JVM.

- 2.164 (2019-02) and newer: Java 8 or Java 11
- 2.54 (2017-04) and newer: Java 8
- 1.612 (2015-05) and newer: Java 7

With that set up, the Jenkins package can be installed with:

See [the installation guide](#) for more information, including how Jenkins is run and where the configuration is stored, etc.

<https://pkg.jenkins.io/debian-stable/>

<https://www.jenkins.io/doc/book/installing/linux/#debianubuntu>

Jenkins Debian packages



If an error is reported, "jenkins : Depends: daemon but it is not installable", add the "universe" apt repository of community maintained free and open source software for Ubuntu by executing this command after `sudo apt-get update`:

```
sudo add-apt-repository universe
```

This package installation will:

- Setup Jenkins as a daemon launched on start. See `/etc/init.d/jenkins` for more details.
- Create a 'jenkins' user to run this service.
- Direct console log output to the file `/var/log/jenkins/jenkins.log`. Check this file if you are troubleshooting Jenkins.
- Populate `/etc/default/jenkins` with configuration parameters for the launch, e.g `JENKINS_HOME`
- Set Jenkins to listen on port 8080. Access this port with your browser to start configuration.



If your `/etc/init.d/jenkins` file fails to start Jenkins, edit the `/etc/default/jenkins` to replace the line `----HTTP_PORT=8080----` with `----HTTP_PORT=8081----` Here, "8081" was chosen but you can put another port available.

<https://www.jenkins.io/doc/book/installing/linux/#weekly-release>

Jenkins Debian packages

Installation of Java

Jenkins requires Java in order to run, yet certain distributions don't include this by default and some Java versions are incompatible with Jenkins.

There are multiple Java implementations which you can use. OpenJDK is the most popular one at the moment, we will use it in this guide.

To install the Open Java Development Kit (OpenJDK) run the following:

- Update the repositories

```
sudo apt update
```

- search of all available packages:

```
sudo apt search openjdk
```

- Pick one option and install it:

```
sudo apt install openjdk-11-jdk
```

- Confirm installation:

```
sudo apt install openjdk-11-jdk
```

- checking installation:

```
java -version
```

- the result must be something like:

```
openjdk version "11.0.9.1" 2020-11-04
OpenJDK Runtime Environment (build 11.0.9.1+1-post-Debian-1deb10u2)
OpenJDK 64-Bit Server VM (build 11.0.9.1+1-post-Debian-1deb10u2, mixed mode, sharing)
```

<https://www.jenkins.io/doc/book/installing/linux/#installation-of-java>

Use SSH clients to copy-paste all the followings commands(except#)

1. `wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -`
2. `sudo nano /etc/apt/sources.list`
3. #Add next string after last row in the editing file
`deb https://pkg.jenkins.io/debian-stable binary/`
#And save changes
4. `sudo apt-get update`
5. `java -version`
6. `sudo apt-get install openjdk-8-jdk`
7. `sudo apt-get install jenkins`
8. `service jenkins status`
9. `sudo cat /var/lib/jenkins/secrets/initialAdminPassword`
#The output of this command should be entered in input field on the next slide

Unlocking Jenkins

Getting Started

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:

```
/var/lib/jenkins/secrets/initialAdminPassword
```

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

Administrator password

Continue

Expand functionality of Jenkins with plugins



Suggested Plugins installation

Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding	<div>Installer</div> <div>** bouncycastle API</div> <div>** JavaScript GUI Lib: ACE Editor bundle</div> <div>** Pipeline: SCM Step</div> <div>** Pipeline: Groovy</div> <div>** Pipeline: Job</div> <div>** Apache HttpComponents Client 4.x API</div> <div>Mailer</div> <div>** Pipeline: Basic Steps</div> <div>Gradle</div> <div>** Pipeline: Milestone Step</div> <div>** Pipeline: Input Step</div> <div>** Pipeline: Stage Step</div> <div>** Pipeline Graph Analysis</div> <div>** Pipeline: REST API</div> <div>** JavaScript GUI Lib: Handlebars bundle</div> <div>** JavaScript GUI Lib: Moment.js bundle</div> <div>Pipeline: Stage View</div> <div>** Pipeline: Build Step</div> <div>** Pipeline: Model API</div> <div>** - required dependency</div>
✓ Timestampers	✓ Workspace Cleanup	✓ Ant	✓ Gradle	
🔄 Pipeline	🔄 GitHub Branch Source	🔄 Pipeline: GitHub Groovy Libraries	✓ Pipeline: Stage View	
🔄 Git	🔄 SSH Build Agents	🔄 Matrix Authorization Strategy	🔄 PAM Authentication	
🔄 LDAP	🔄 Email Extension	✓ Mailer		


Jenkins 2.303.1

Create user with admin privileges


Getting Started

Create First Admin User


Username:



Password:



Confirm password:



Full name:

E-mail address:

Jenkins 2.303.1

[Skip and continue as admin](#)

[Save and Continue](#)

Language will be set to English as default using plugin
Locale (later)

Jenkins instance configuration

“Grey” or behind NAT IP makes some difficulties. We’ll talk about it.

Getting Started

Instance Configuration

Jenkins URL:

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.303.1

Not now

Save and Finish

Getting Started


Jenkins is ready!




Your Jenkins setup is complete.


Start using Jenkins


Jenkins 2.303.1


First look at Jenkins GUI. Let's make it in English


 **Jenkins**


search ?  1  jenkins  log out


 New Item


 People

 Build History

 Manage Jenkins

 My Views

 Lockable Resources

 New View


Build Queue ^

No builds in the queue.

Build Executor Status ^

1 Idle

2 Idle

 add description

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job →

Set up a distributed build

Set up an agent →

Configure a cloud →

Learn more about distributed builds ↗

REST API Jenkins 2.303.1

Install first plugin manually

The screenshot shows the Jenkins web interface. The left sidebar contains a 'Manage Jenkins' link, which is highlighted with a red rectangle. The main content area is titled 'Manage Jenkins' and contains several sections: 'System Configuration', 'Security', 'Status Information', 'Troubleshooting', and 'Tools and Actions'. The 'Manage Plugins' option is highlighted with a red rectangle in the 'System Configuration' section. The top of the dashboard shows the Jenkins logo, a search bar, and user information. The bottom of the dashboard shows the REST API and Jenkins version (2.303.1).

Jenkins Dashboard

Manage Jenkins

Building on the controller node can be a security issue. You should set up distributed builds. See [the documentation](#).

[Set up agent](#) [Set up cloud](#) [Diagnosis](#)

System Configuration

- Configure System**
Configure global settings and paths.
- Global Tool Configuration**
Configure tools, their locations and automatic installers.
- Manage Plugins**
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- Manage Nodes and Clouds**
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Security

- Configure Global Security**
Secure Jenkins; define who is allowed to access/use the system.
- Manage Credentials**
Configure credentials
- Configure Credential Providers**
Configure the credential providers and types
- Manage Users**
Create/delete/modify users that can log in to this Jenkins

Status Information

- System Information**
Displays various environmental information to assist trouble-shooting.
- System Log**
System log captures output from java, url1, logging output related to Jenkins.
- Load Statistics**
Check your resource utilization and see if you need more computers for your builds.
- About Jenkins**
See the version and license information.

Troubleshooting

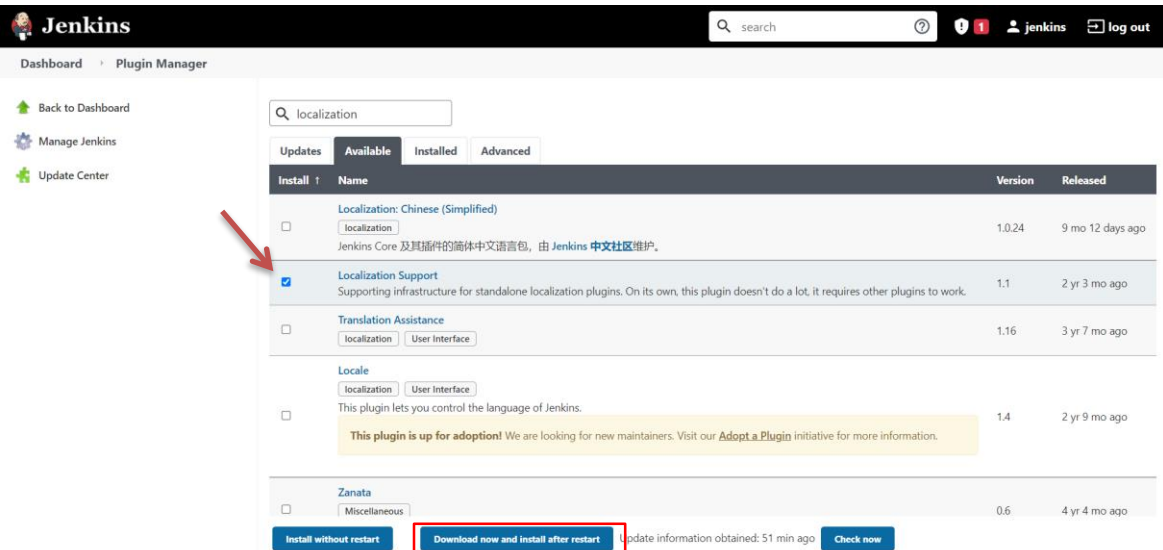
- Manage Old Data**
Scrub configuration files to remove remnants from old plugins and earlier versions.

Tools and Actions

- Reload Configuration from Disk**
Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.
- Jenkins CLI**
Access/manage Jenkins from your shell, or from your script.
- Script Console**
Executes arbitrary script for administration/trouble-shooting/diagnostics.
- Prepare for Shutdown**
Stops executing new builds, so that the system can be eventually shut down safely.

REST API Jenkins 2.303.1

Plugin Locale installation



Jenkins

Dashboard > Plugin Manager

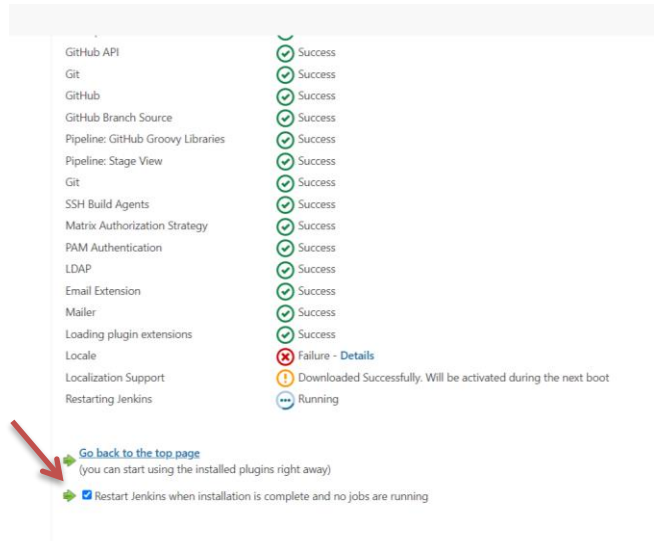
Back to Dashboard Manage Jenkins Update Center

Search: localization

Updates Available Installed Advanced

Install	Name	Version	Released
<input type="checkbox"/>	Localization: Chinese (Simplified) localization Jenkins Core 及其插件的简体中文语言包, 由 Jenkins 中文社区维护。	1.0.24	9 mo 12 days ago
<input checked="" type="checkbox"/>	Localization Support Supporting infrastructure for standalone localization plugins. On its own, this plugin doesn't do a lot, it requires other plugins to work.	1.1	2 yr 3 mo ago
<input type="checkbox"/>	Translation Assistance localization User Interface	1.16	3 yr 7 mo ago
Locale localization User Interface This plugin lets you control the language of Jenkins. This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.			
<input type="checkbox"/>	Zanata Miscellaneous	0.6	4 yr 4 mo ago

Install without restart Download now and install after restart Update information obtained: 51 min ago Check now



Plugin	Status
GitHub API	Success
Git	Success
GitHub	Success
GitHub Branch Source	Success
Pipeline: GitHub Groovy Libraries	Success
Pipeline: Stage View	Success
Git	Success
SSH Build Agents	Success
Matrix Authorization Strategy	Success
PAM Authentication	Success
LDAP	Success
Email Extension	Success
Mailer	Success
Loading plugin extensions	Success
Locale	Failure - Details
Localization Support	Downloaded Successfully. Will be activated during the next boot
Restarting Jenkins	Running

Go back to the top page
(you can start using the installed plugins right away)

☒ Restart Jenkins when installation is complete and no jobs are running

Chrome Language setting

Settings

Lang

You and Google

Autofill

Safety check

Privacy and security

Appearance

Search engine

Default browser

On startup

Advanced

Extensions

About Chrome

Languages

Language

Order languages based on your preference

English

This language is used to display the Google Chrome

Ukrainian

Russian

This language is used when translating pages

English (United States)

Add languages

Offer to translate pages that aren't in a language you read

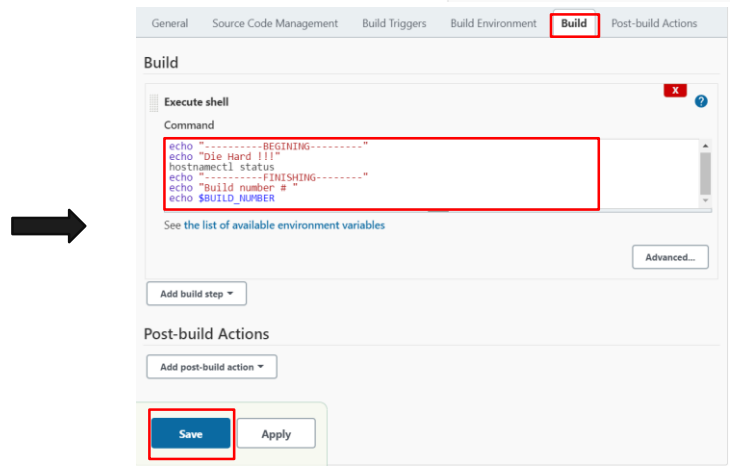
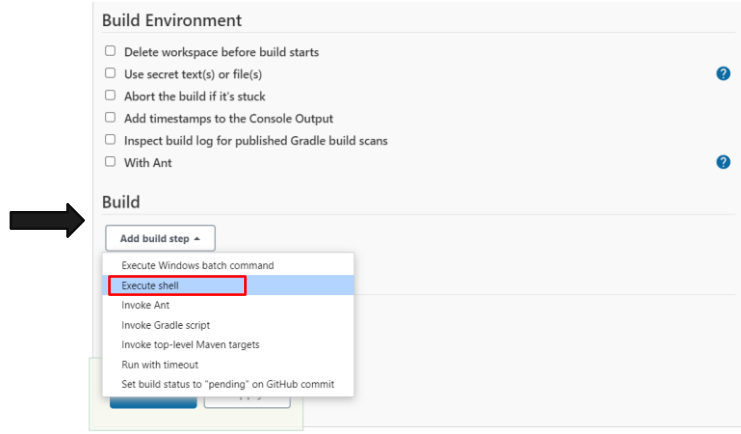
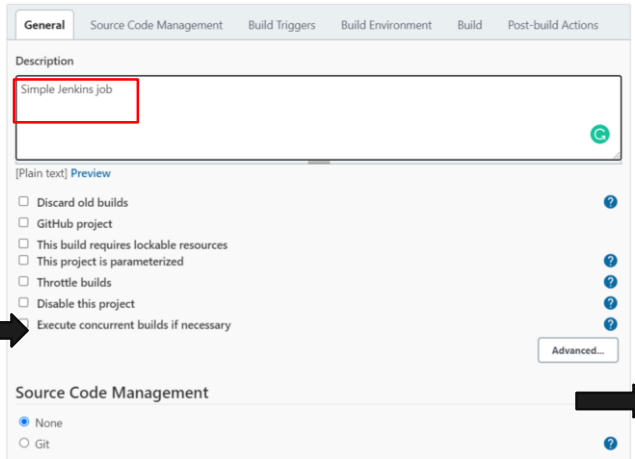
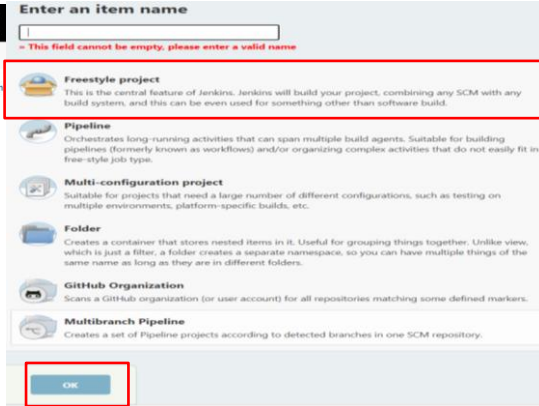
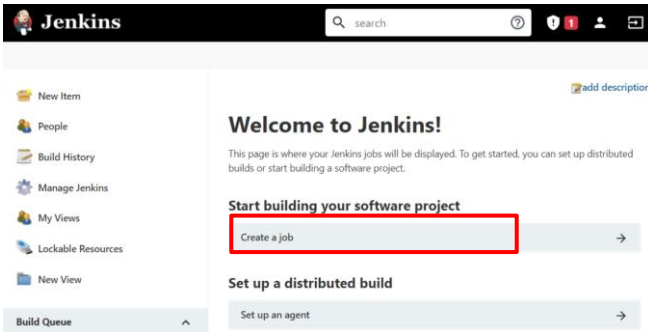
Spell check

Basic spell check

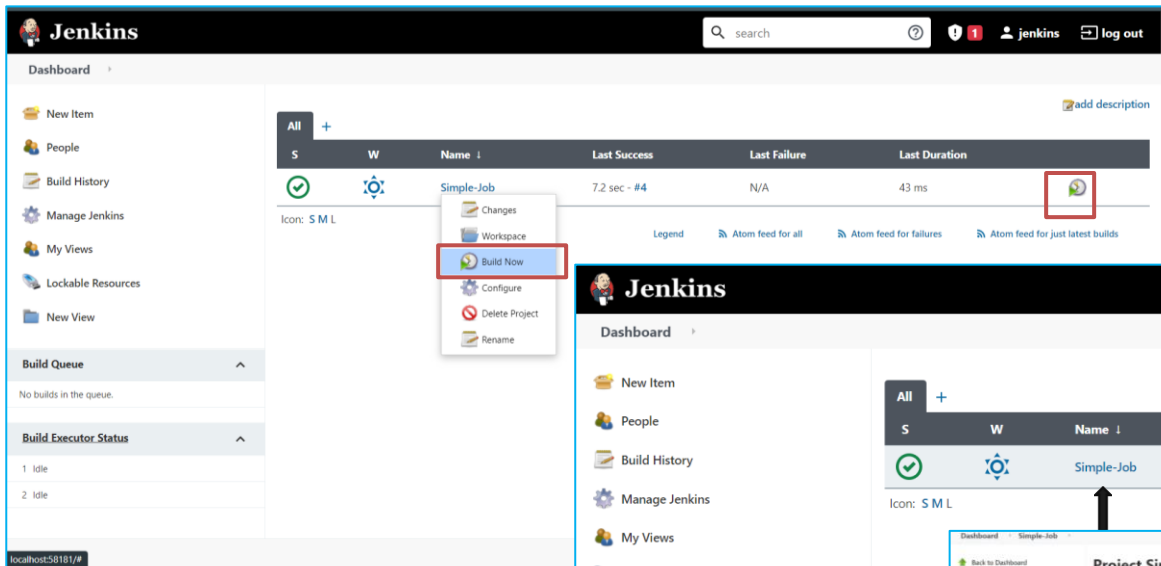
Enhanced spell check

Uses the same spell checker that's used in Google search. Text you type in the browser is sent to Google.

Simple job example in Jenkins



Simple job example in Jenkins

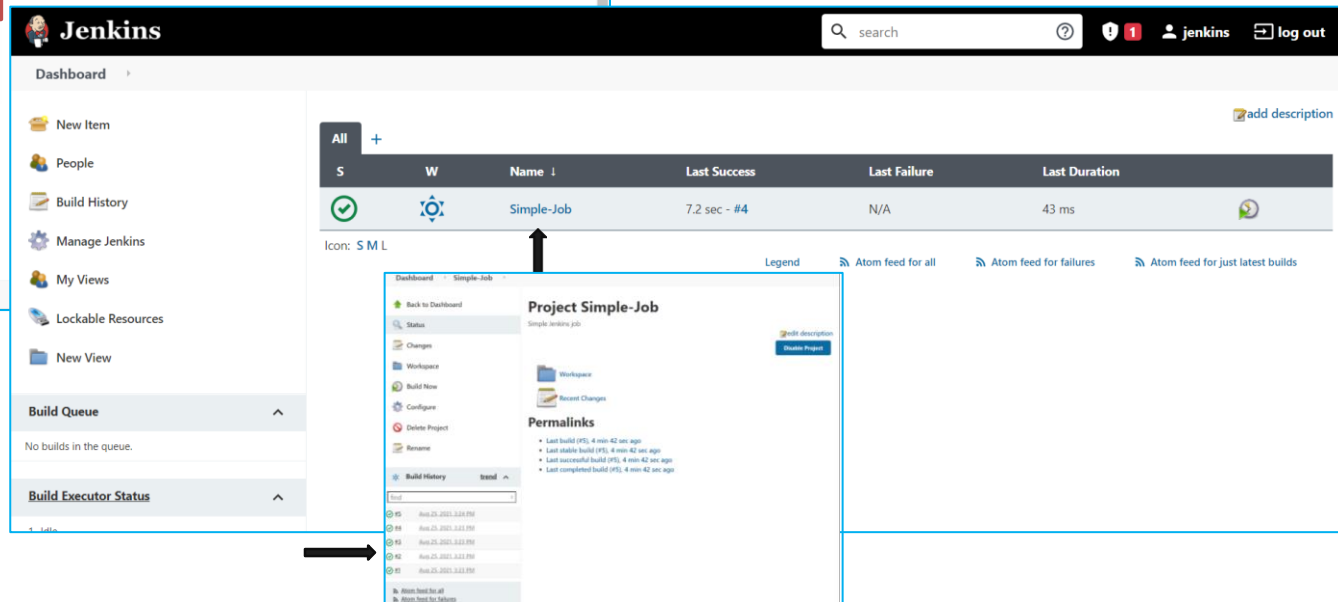


The screenshot shows the Jenkins Dashboard with a table of jobs. The 'Simple-Job' is listed with a status of 'Success' (green checkmark), a last success time of '7.2 sec - #4', and a last duration of '43 ms'. A red box highlights the 'Build Now' button in the context menu.

S	W	Name	Last Success	Last Failure	Last Duration
✓	⚙️	Simple-Job	7.2 sec - #4	N/A	43 ms

Icon: S M L

- Changes
- Workspace
- Build Now
- Configure
- Delete Project
- Rename



The screenshot shows the Jenkins Dashboard with a table of jobs. The 'Simple-Job' is listed with a status of 'Success' (green checkmark), a last success time of '7.2 sec - #4', and a last duration of '43 ms'. A red box highlights the 'Build Now' button in the context menu.

S	W	Name	Last Success	Last Failure	Last Duration
✓	⚙️	Simple-Job	7.2 sec - #4	N/A	43 ms

Icon: S M L

Project Simple-Job

Simple Jenkins job

Workspace

Recent Changes

Permalinks

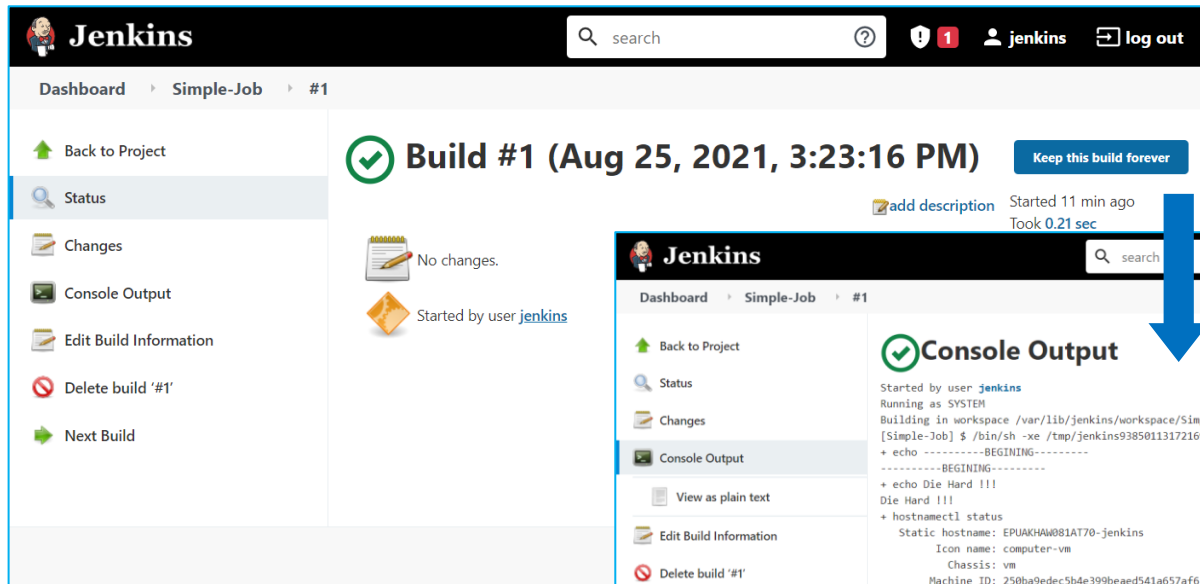
- Last build (PS), 4 min 42 sec ago
- Last stable build (PS), 4 min 42 sec ago
- Last successful build (PS), 4 min 42 sec ago
- Last completed build (PS), 4 min 42 sec ago

Build History

Build Queue

Build Executor Status

Simple job example in Jenkins (Finished: SUCCESS)



The screenshot shows the Jenkins Dashboard for a job named 'Simple-Job' with build #1. The top navigation bar includes the Jenkins logo, a search bar, and user information. The left sidebar contains links to 'Back to Project', 'Status', 'Changes', 'Console Output', 'Edit Build Information', 'Delete build '#1'', and 'Next Build'. The main content area displays 'Build #1 (Aug 25, 2021, 3:23:16 PM)' with a green checkmark icon, indicating success. It also shows 'No changes' and 'Started by user jenkins'. A blue arrow points from the 'Console Output' link in the sidebar to the right-hand screenshot.

Jenkins

Dashboard > Simple-Job > #1

Back to Project

Status

Changes

Console Output

Edit Build Information

Delete build '#1'

Next Build

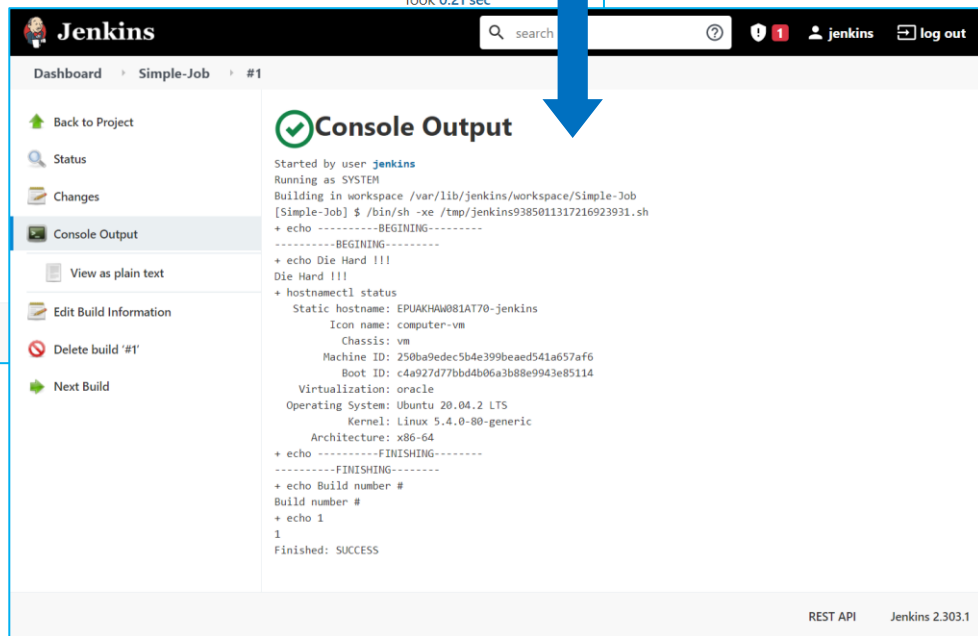
Build #1 (Aug 25, 2021, 3:23:16 PM)

Keep this build forever

add description Started 11 min ago Took 0.21 sec

No changes.

Started by user [jenkins](#)



The screenshot shows the Jenkins Console Output for the same job. The left sidebar is similar to the first screenshot, but the 'Console Output' link is highlighted. The main content area displays the console output, which includes system information and build details. A blue arrow points from the 'Console Output' link in the sidebar to this screenshot.

Jenkins

Dashboard > Simple-Job > #1

Back to Project

Status

Changes

Console Output

View as plain text

Edit Build Information

Delete build '#1'

Next Build

Console Output

Started by user [jenkins](#)

Running as SYSTEM

Building in workspace /var/lib/jenkins/workspace/Simple-Job

[Simple-Job] \$ /bin/sh -xe /tmp/jenkins9385011317216923931.sh

```
+ echo -----BEGINNING-----  
-----BEGINNING-----  
+ echo Die Hard !!!  
Die Hard !!!  
+ hostnamectl status  
  Static hostname: EPUAKHAW081AT70-jenkins  
    Icon name: computer-vm  
  Chassis: vm  
  Machine ID: 250ba9edec5b4e399beaed541a657af6  
    Boot ID: c4a927d77bbd4b06a3b88e9943e85114  
  Virtualization: oracle  
  Operating System: Ubuntu 20.04.2 LTS  
    Kernel: Linux 5.4.0-80-generic  
  Architecture: x86_64  
+ echo -----FINISHING-----  
-----FINISHING-----  
+ echo Build number #  
Build number #  
+ echo 1  
1  
Finished: SUCCESS
```

REST API Jenkins 2.303.1

Simple job example in Jenkins (Finished: FAILED)

General Source Code Management Build Triggers Build Environment **Build** Post-build Actions

Build

Execute shell ✖ ?

Command

```
echo "-----BEGINNING-----"  
echo "Die Hard !!!"  
hostnamectl status  
echo "-----FINISHING-----"  
echo "Build number # "  
echo $BUILD_NUMBER  
cat file-student.log
```

See the list of available environment variables

Advanced...

Add build step ▾

Post-build Actions

Add post-build action ▾

Save Apply

Jenkins

Dashboard Simple-Job #6

Back to Project

Status

Changes

Console Output

Edit Build Information

Delete build '#6'

Previous Build

✖ Build #6 (Aug 25, 2021, 3:42:39 PM)

Keep this build forever

[add description](#) Started 5.2 sec ago Took 0.15 sec

No changes.

Started by user jenkins

Dashboard Simple-Job #6

Back to Project

Status

Changes

Console Output

View as plain text

Edit Build Information

Delete build '#6'

Previous Build

✖ Console Output

Started by user jenkins

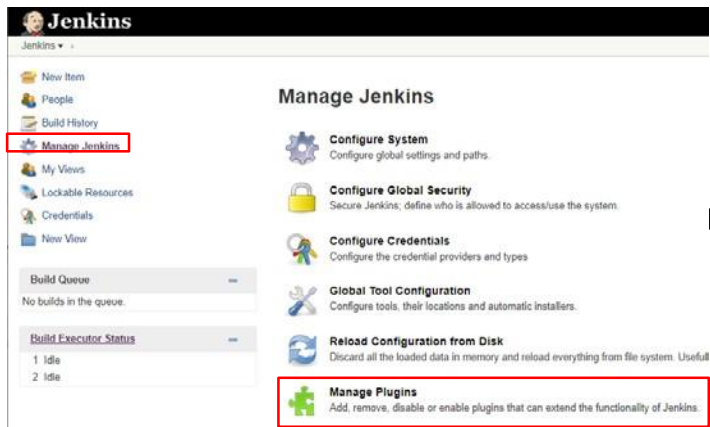
Running as SYSTEM

Building in workspace /var/lib/jenkins/workspace/Simple-Job

[Simple-Job] \$ /bin/sh -xe /tmp/jenkins9931874526030706927.sh

```
+ echo -----BEGINNING-----  
-----BEGINNING-----  
+ echo Die Hard !!!  
Die Hard !!!  
+ hostnamectl status  
  Static hostname: EPUAKHw081AT70-jenkins  
    Icon name: computer-vm  
  Chassis: vm  
  Machine ID: 250ba9edec5b4e399beaed541a657af6  
    Boot ID: c4a927d77bbd4b06a3b88e9943e85114  
  Virtualization: oracle  
  Operating System: Ubuntu 20.04.2 LTS  
    Kernel: Linux 5.4.0-80-generic  
  Architecture: x86_64  
+ echo -----FINISHING-----  
-----FINISHING-----  
+ echo Build number #  
Build number #  
+ echo 6  
6  
+ cat file-student.log  
cat: file-student.log: No such file or directory  
Build step 'Execute shell' marked build as failure  
Finished: FAILURE
```

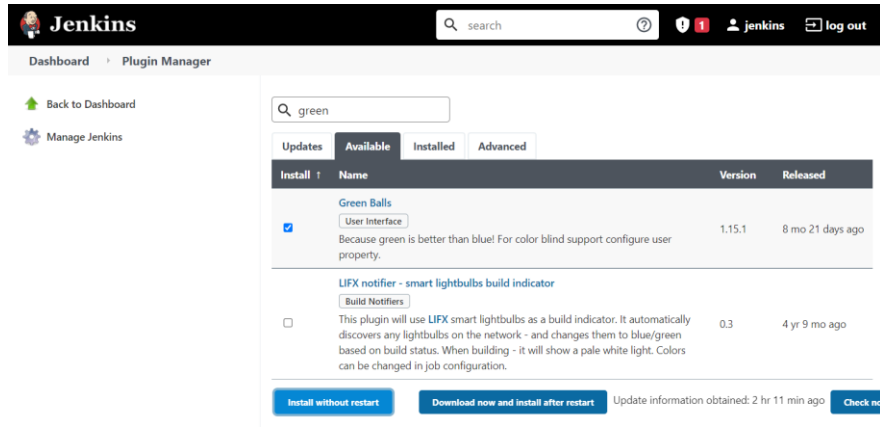
Frequently used plugins make GUI better: Green Balls (green better than blue :-)



The image shows the Jenkins 'Manage Jenkins' page. On the left sidebar, 'Manage Jenkins' is highlighted with a red box. In the main content area, 'Manage Plugins' is also highlighted with a red box. A large black arrow points from this page towards the right.

Manage Jenkins

- Configure System**
Configure global settings and paths.
- Configure Global Security**
Secure Jenkins; define who is allowed to access/use the system.
- Configure Credentials**
Configure the credential providers and types.
- Global Tool Configuration**
Configure tools, their locations and automatic installers.
- Reload Configuration from Disk**
Discard all the loaded data in memory and reload everything from file system. Usefull
- Manage Plugins**
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.



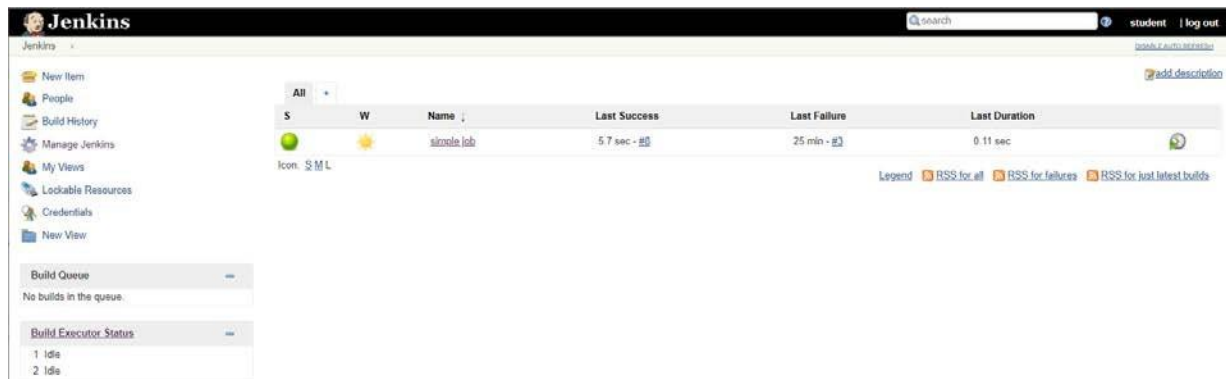
The image shows the Jenkins 'Plugin Manager' page. The 'Available' tab is selected. A search bar contains the text 'green'. The 'Green Balls' plugin is listed and checked for installation. The 'LIFX notifier' plugin is also listed but not checked.

Plugin Manager

Search: green

Install	Name	Version	Released
<input checked="" type="checkbox"/>	Green Balls User Interface Because green is better than blue! For color blind support configure user property.	1.15.1	8 mo 21 days ago
<input type="checkbox"/>	LIFX notifier - smart lightbulbs build indicator Build Notifiers This plugin will use LIFX smart lightbulbs as a build indicator. It automatically discovers any lightbulbs on the network - and changes them to blue/green based on build status. When building - it will show a pale white light. Colors can be changed in job configuration.	0.3	4 yr 9 mo ago

Buttons: Install without restart, Download now and install after restart, Update information obtained: 2 hr 11 min ago, Check for updates

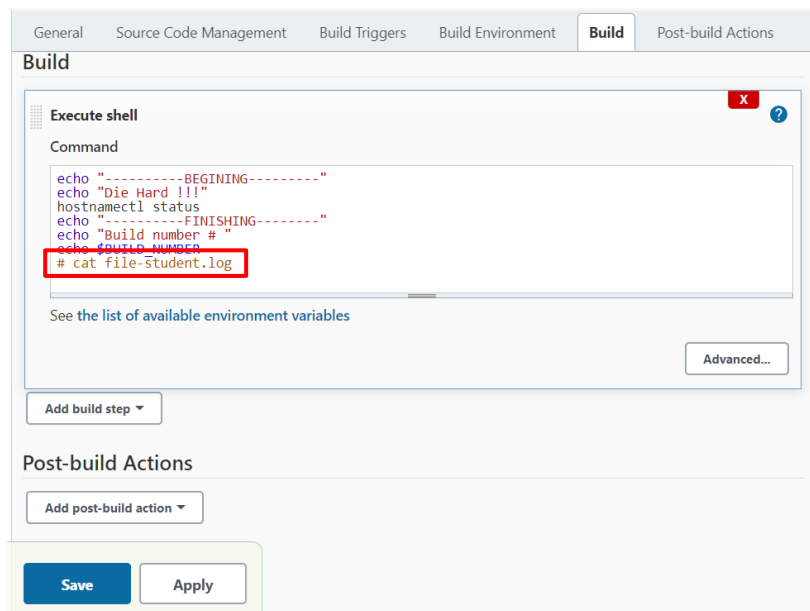


The image shows the Jenkins 'Build Queue' page. The 'Build Queue' section shows 'No builds in the queue.' The 'Build Executor Status' section shows 1 Idle and 2 Idle executors.

Build Queue
No builds in the queue.

Build Executor Status
1 Idle
2 Idle

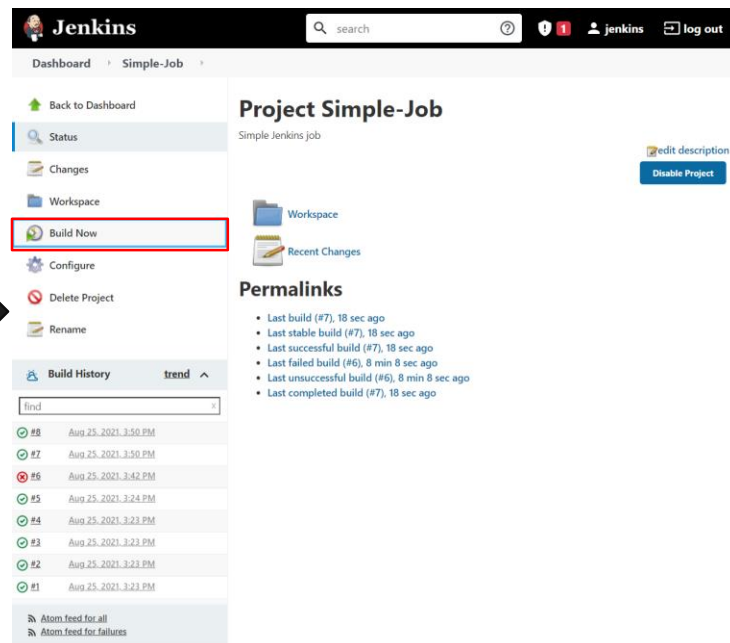
Simple job example in Jenkins (Finished: SUCCESS) with GreenBalls plugin enabled



The screenshot shows the Jenkins configuration page for a job named 'Simple-Job'. The 'Build' tab is selected. Under the 'Execute shell' section, the command is:

```
echo "-----BEGINNING-----"  
echo "Die Hard !!!"  
hostnamectl status  
echo "-----FINISHING-----"  
echo "Build number #"  
echo "BUILD NUMBER"  
# cat file-student.log
```

The command is enclosed in a red box. Below the command field, there is a link to 'See the list of available environment variables' and an 'Advanced...' button. At the bottom, there are 'Save' and 'Apply' buttons.



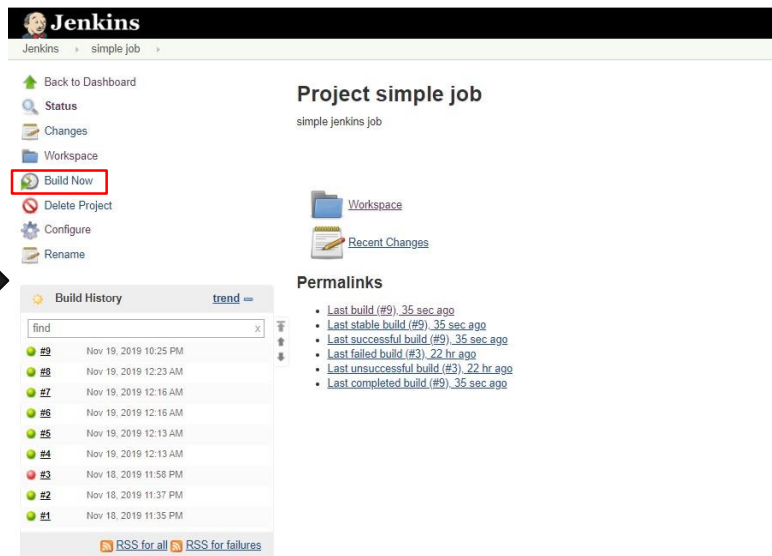
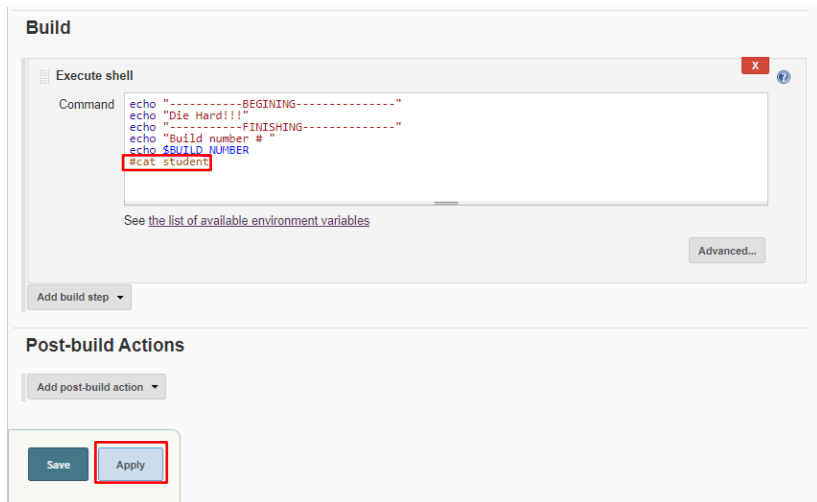
The screenshot shows the Jenkins 'Project Simple-Job' status page. The 'Build Now' button is highlighted with a red box. The 'Build History' section shows a list of builds with their status and timestamps:

Build Number	Status	Timestamp
#8	Success	Aug 25, 2021 3:50 PM
#7	Success	Aug 25, 2021 3:50 PM
#6	Failure	Aug 25, 2021 3:42 PM
#5	Success	Aug 25, 2021 3:24 PM
#4	Success	Aug 25, 2021 3:23 PM
#3	Success	Aug 25, 2021 3:23 PM
#2	Success	Aug 25, 2021 3:23 PM
#1	Success	Aug 25, 2021 3:23 PM

The 'Permalinks' section shows a list of links to various build details:

- Last build (#7), 18 sec ago
- Last stable build (#7), 18 sec ago
- Last successful build (#7), 18 sec ago
- Last failed build (#6), 8 min 8 sec ago
- Last unsuccessful build (#6), 8 min 8 sec ago
- Last completed build (#7), 18 sec ago

Simple job example in Jenkins (Finished: SUCCESS) with GreenBalls plugin enabled



References

- <https://jenkins.io/doc/book/installing/>
- <https://jenkins.io/doc/tutorials/>
- <https://www.javatpoint.com/jenkins>
- <https://www.edureka.co/blog/jenkins-tutorial/>
- <https://wiki.jenkins.io/display/JENKINS/Home>

Q&A

A light blue world map is centered in the background, showing the outlines of continents and countries. The map is slightly faded and serves as a backdrop for the text.

Thank you!