



Hosting a webserver on local host

Prerequisites

Ensure the following are installed and configured **before** creating the Jenkins job:

- **JDK** (Java 8 or higher)
- **Maven**
- **Apache Tomcat**
- **Jenkins**
- Jenkins plugins:
 - Git Plugin
 - Maven Integration Plugin

with all the necessary packages, tools and plugins installed,

1. Creating the pipeline foundation

Head too Jenkins home page and click on New Item/ Job

Name the project as required and select Free style project

Jenkins / All / New Item

New Item

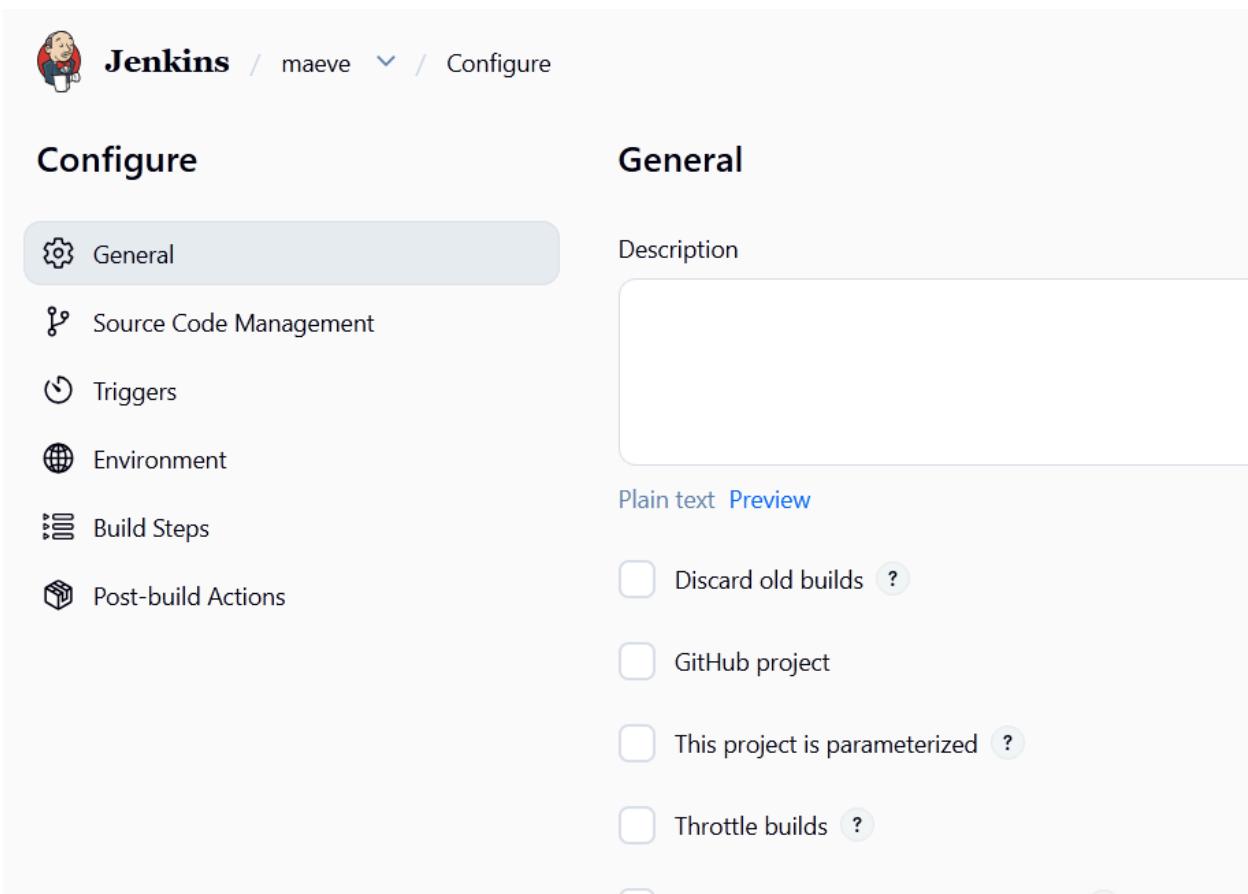
Enter an item name

Select an item type

- Pipeline**
Build, test, and deploy using pipelines. Supports stages, parallel work, and running on multiple agents.
- Freestyle project**
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Maven project**
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
- Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.

2. Pipeline

After the creation, we get to see this screen



The screenshot shows the Jenkins 'Configure' screen for a project named 'maeve'. The left sidebar lists configuration sections: General (selected), Source Code Management, Triggers, Environment, Build Steps, and Post-build Actions. The main area is titled 'General' and contains a 'Description' field (empty) and a 'Plain text Preview' section. Under 'Plain text Preview', there are four checkboxes: 'Discard old builds', 'GitHub project', 'This project is parameterized', and 'Throttle builds'. Each checkbox has a question mark icon next to it.

Leave most of it as is and under the---

Source code management

- We need to link the source code, Let's say we have it in the git repository, link the code here

Credentials

- Leave empty if public repo
- If private: add GitHub username + Personal Access Token

Source Code Management

Connect and manage your code repository to automatically pull the latest code for your builds.

None

Git [?](#)

Repositories [?](#)

Repository URL [?](#)

`https://github.com/Msocial123/EcommerceApp.git`

! Please enter Git repository.

Credentials [?](#)

- none -

Advanced [▼](#)

The screenshot shows a configuration interface for a build system. Under 'Source Code Management', the 'Git' option is chosen. In the 'Repository URL' field, the URL 'https://github.com/Msocial123/EcommerceApp.git' is entered. A red exclamation mark icon indicates an error: 'Please enter Git repository.'. The 'Credentials' section shows no entries. An 'Advanced' button is visible at the bottom right.

Now select the branch where the code exists. here the code exists in the master folder

Branches to build

Use explicit reference:

`refs/heads/main`

(or `refs/heads/master` if applicable)

Branches to build [?](#)

Branch Specifier (blank for 'any') [?](#)

*/master

[+ Add Branch](#)

leave the triggers and environment as is

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

- Trigger builds remotely (e.g., from scripts) [?](#)
- Build after other projects are built [?](#)
- Build periodically [?](#)
- GitHub hook trigger for GITScm polling [?](#)
- Poll SCM [?](#)

Environment

Configure settings and variables that define the context in which your build runs, like credentials, paths, and global parameters.

- Delete workspace before build starts
- Use secret text(s) or file(s) [?](#)
- Provide Configuration files [?](#)
- Add timestamps to the Console Output
- Inspect build log for published build scans
- Terminate a build if it's stuck
- With Ant [?](#)

3. Build Steps

Add build step → **Invoke top-level Maven targets**

Goals

clean package

This command:

- Compiles the project
- Generates a WAR file

The screenshot shows a user interface for managing build steps. At the top, there is a red box around the 'Build Steps' section. Below it, a large red box highlights the 'Invoke top-level Maven targets' option in a list of available steps.

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

+ Add build step

Filter

- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets**
- Provide Configuration files
- Run with timeout
- Set build status to "pending" on GitHub commit

select maven version

Build Steps

Automate your build process with ordered tasks like coc

⋮ Invoke top-level Maven targets ?

Maven Version

(Default)

(Default)

maven_

and define goals

Build Steps

Automate your build process

⋮ Invoke top-level lifecycle phase

Maven Version

maven_

Goals

clean project

4. Post build steps

click on add post build and select this option

Post-build Actions

Define what happens after a build completes, like

+ Add post-build action

The screenshot shows a list of Jenkins post-build actions. At the top is a 'Filter' input field. Below it is a list of actions: 'Aggregate downstream test results', 'Archive the artifacts', 'Build other projects', 'Publish JUnit test result report', 'Publish Javadoc', 'Record fingerprints of files to track usage', 'Git Publisher', 'Deploy war/ear to a container' (which is highlighted with a red rectangular border), 'E-mail Notification', 'Editable Email Notification', 'Set GitHub commit status (universal)', 'Set build status on GitHub commit [deprecated]', and 'Delete workspace when build is done'. At the bottom of the list is a button labeled '+ Add post-build action'.

- Aggregate downstream test results
- Archive the artifacts
- Build other projects
- Publish JUnit test result report
- Publish Javadoc
- Record fingerprints of files to track usage
- Git Publisher
- Deploy war/ear to a container
- E-mail Notification
- Editable Email Notification
- Set GitHub commit status (universal)
- Set build status on GitHub commit [deprecated]
- Delete workspace when build is done

+ Add post-build action

define the war/ear files as so

Post-build Actions

Define what happens after a build completes, like sending notifi

⋮ Deploy war/ear to a container

WAR/EAR files ?

**/*.war

War/ear files to deploy. Relative to the workspace root. You

Click on add container

Containers

+ Add Container



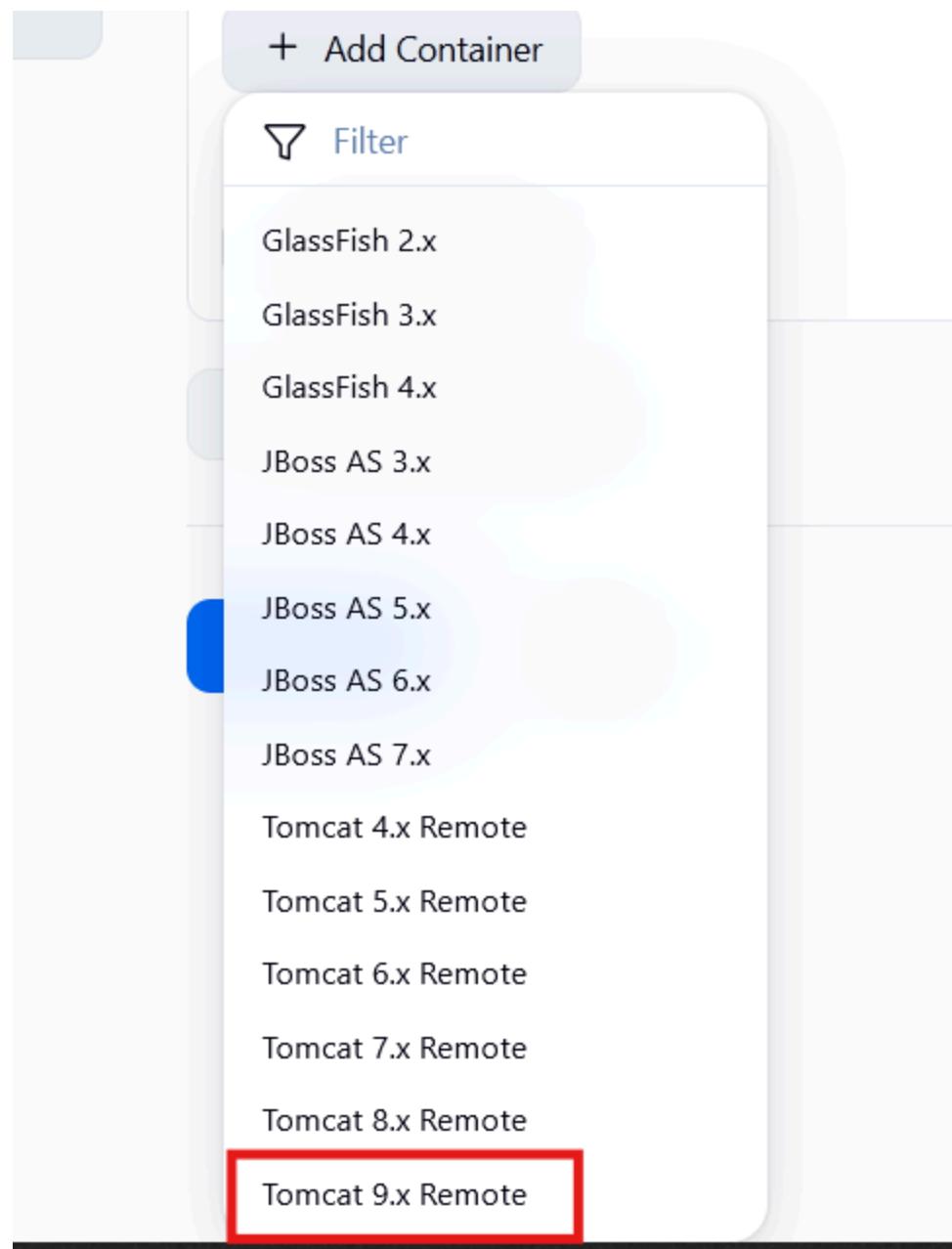
Deploy on failure

+ Add post-build action

Save

Apply

select tomcat version



select tomcat credentials

Containers

Tomcat 9.x Remote

Credentials

ash/********

- none -

ash/********

ashtwt/******** (tomcat_credentials)

Advanced ▾

and finally give the tomcat local host url

Containers

Tomcat 9.x Remote

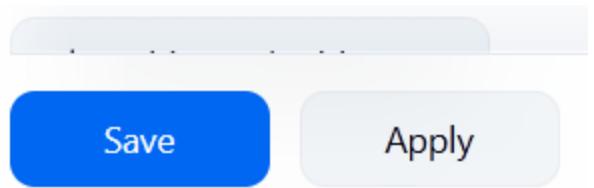
Credentials

ash/********

Tomcat URL ?

http://localhost:8082

Save and apply



click on build now

A screenshot of the Jenkins web interface for the "maeve" project. The top navigation bar shows the Jenkins logo and the project name "maeve". Below the navigation, there is a sidebar with several options: "Status", "Changes", "Workspace", "Build Now" (which is highlighted with a red rectangular box), "Configure", "Delete Project", and "Rename". The main content area is titled "Builds >" and displays the message "No builds".

Jenkins / maeve

- Status
- </> Changes
- Workspace
- Build Now
- Configure
- Delete Project
- Rename

Builds > ...

No builds

Verification

After successful build:

1. Jenkins console output shows:

BUILD SUCCESS

```
[INFO] Packaging webapp
[INFO] Assembling webapp [maven-web-application]
[INFO] Processing war project
[INFO] Copying webapp resources [C:\ProgramData\
[INFO] Building war: C:\ProgramData\Jenkins\.jen
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  5.946 s
[INFO] Finished at: 2026-01-28T11:19:56+05:30
[INFO] -----
[DeployPublisher][INFO] Attempting to deploy 1 w
[DeployPublisher][INFO] Deploying C:\ProgramData\
Tomcat 9.x Remote with context null
[C:\ProgramData\Jenkins\.jenkins\workspace\mav
Deploying [C:\ProgramData\Jenkins\.jenkins\wor
Finished: SUCCESS
```

The screenshot shows the Tomcat Web Application Manager interface. At the top left is a cartoon cat icon, and at the top right is the "THE ASF" logo. Below the header is a message box containing "Message: OK". The main area is titled "Tomcat Web Application Manager" and contains a table of applications:

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/EcommerceApp	None specified		true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/maven-web-application	None specified	maven-web-application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/petclinic	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

The screenshot shows the homepage of an online shopping website for electronic products. At the top center is the logo "Online Electronic Shopping" with a shopping cart icon showing "0". To the right are links for Customer Registration, Customer Login, Admin Login, and Contact Us. Below the header is a navigation bar with links: Home, View Category, Tv, Laptop, Mobile, Watch, View All Product, and About Us.

The main content area has a light gray background and features a large "Home" title. On the left, there is promotional text: "Work, Play, Learn," and "Explore our wide range of laptops". On the right, there is an image of an Acer laptop displaying a video game scene.