

## Lesson 09 Demo 03

### Installing and Configuring ZAP Plugin on Jenkins

**Objective:** To install and configure the OWASP ZAP plugin on Jenkins to automate security testing of web applications during the build process

**Tools required:** Jenkins

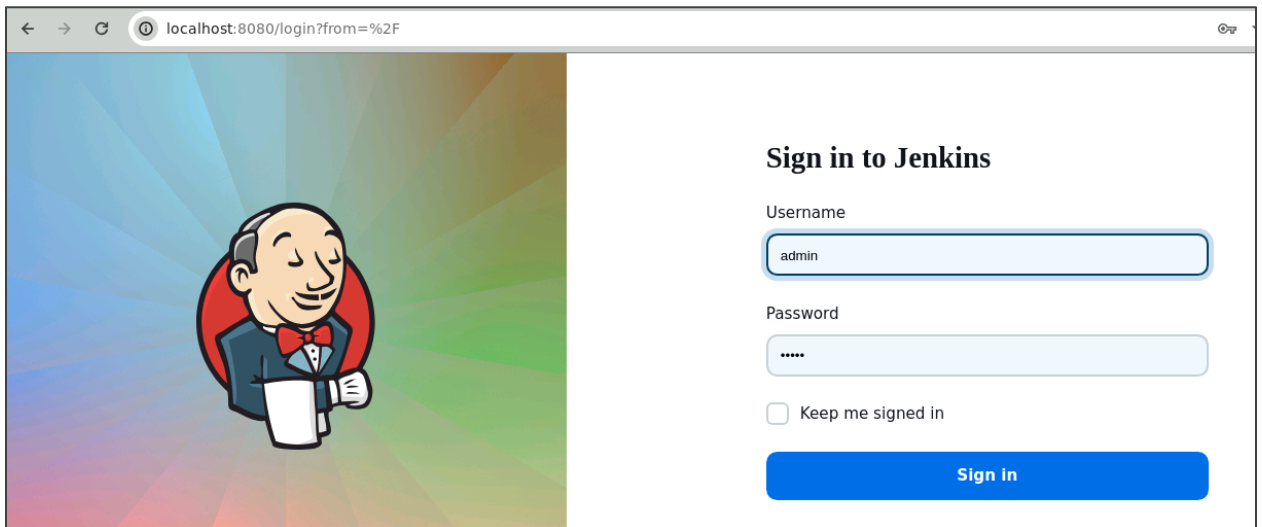
**Prerequisites:** You need to have a Jenkins up and running.

Steps to be followed:

1. Configure OWASP ZAP tool in Jenkins
2. Create a Jenkins pipeline job to integrate the vulnerability scan tool

#### Step 1: Configure OWASP ZAP tool in Jenkins

##### 1.1 Log in to Jenkins using your credentials

A screenshot of a web browser window showing the Jenkins login page. The browser's address bar displays 'localhost:8080/login?from=%2F'. The page features a large illustration of a man in a tuxedo on the left and a login form on the right. The form is titled 'Sign in to Jenkins' and includes fields for 'Username' (containing 'admin') and 'Password' (masked with dots). Below the password field is a checkbox labeled 'Keep me signed in'. A blue 'Sign in' button is at the bottom of the form.

← → ↻ 🔒 localhost:8080/login?from=%2F

**Sign in to Jenkins**

Username  
admin

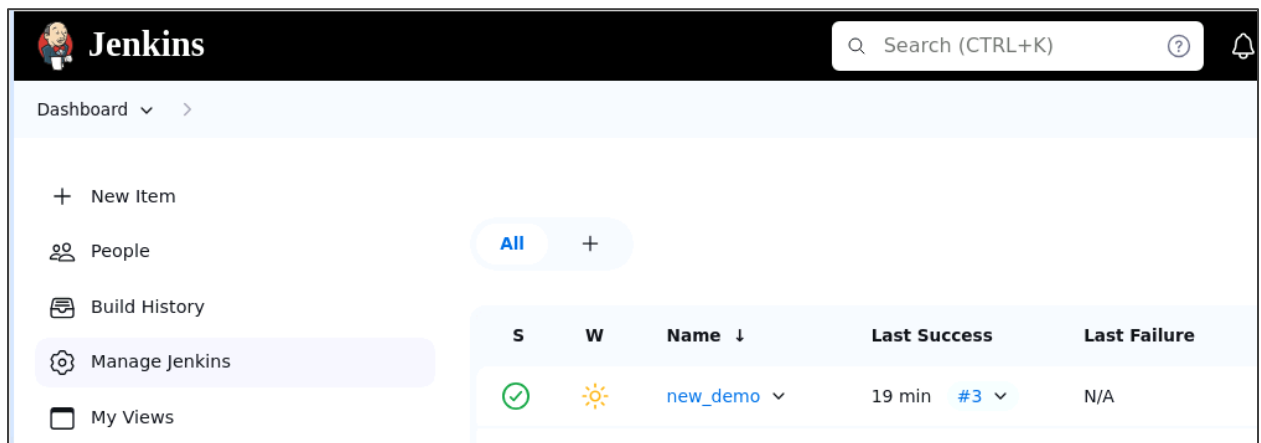
Password  
.....

☐ Keep me signed in

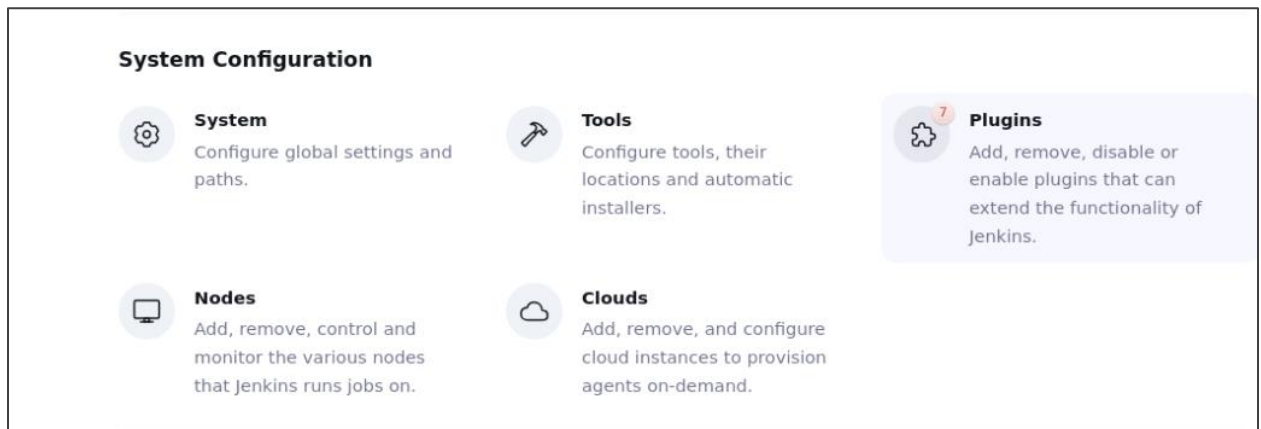
**Sign in**

**Note:** The credentials for accessing Jenkins in the lab are Username: **admin** and Password: **admin**.

1.2 In the Jenkins dashboard, navigate to **Manage Jenkins**, and under **System Configuration**, click on **Plugins**

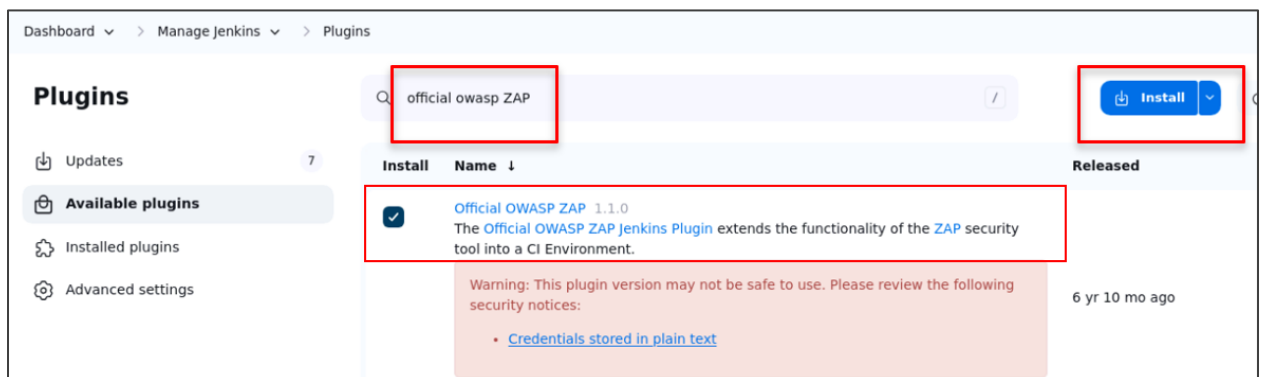


The screenshot shows the Jenkins dashboard. At the top, there's a header with the Jenkins logo and a search bar. Below the header, there's a sidebar with navigation links: Dashboard, New Item, People, Build History, Manage Jenkins (highlighted), and My Views. The main content area shows a table with columns: S, W, Name, Last Success, and Last Failure. A single row is visible with a green checkmark in the S column, a sun icon in the W column, the name 'new\_demo', a last success time of '19 min', and a last failure status of 'N/A'.



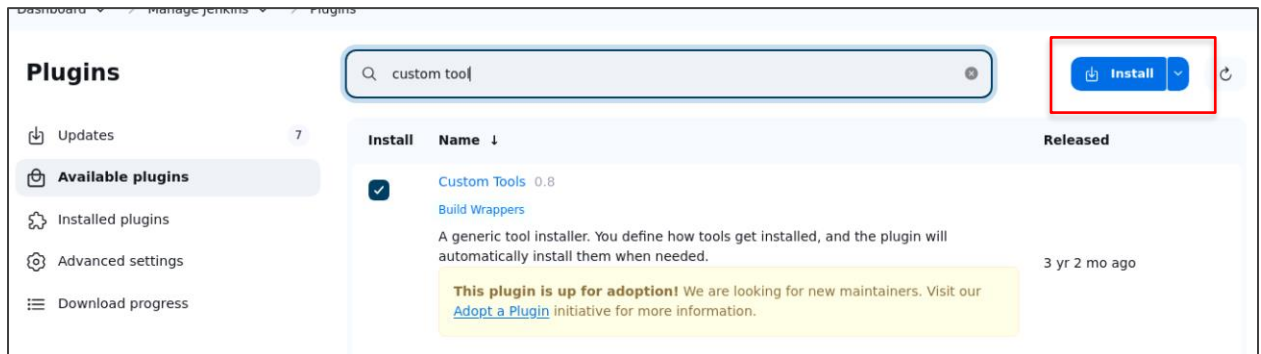
The screenshot shows the 'System Configuration' page. It features five cards: 'System' (Configure global settings and paths), 'Tools' (Configure tools, their locations and automatic installers), 'Plugins' (Add, remove, disable or enable plugins that can extend the functionality of Jenkins), 'Nodes' (Add, remove, control and monitor the various nodes that Jenkins runs jobs on), and 'Clouds' (Add, remove, and configure cloud instances to provision agents on-demand).

1.3 In the **Available plugins**, search for the **Official OWASP ZAP** plugin and click on **Install**

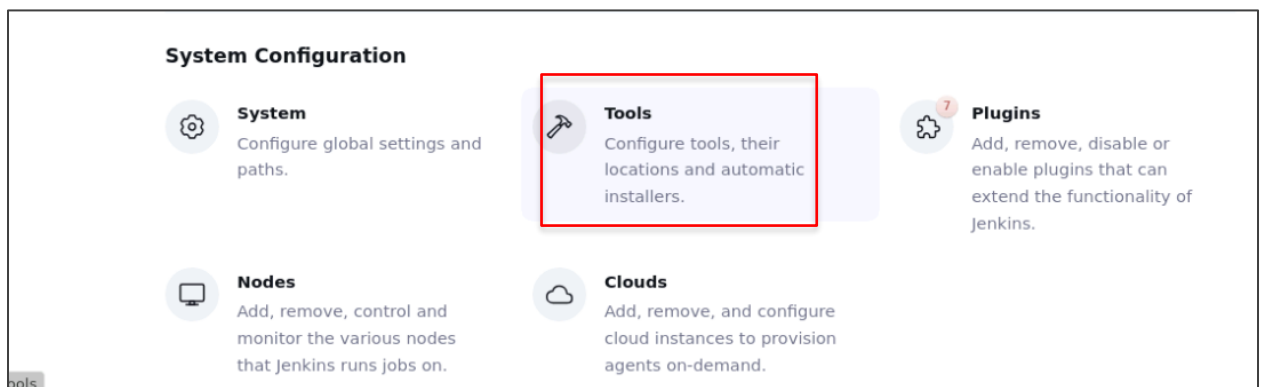
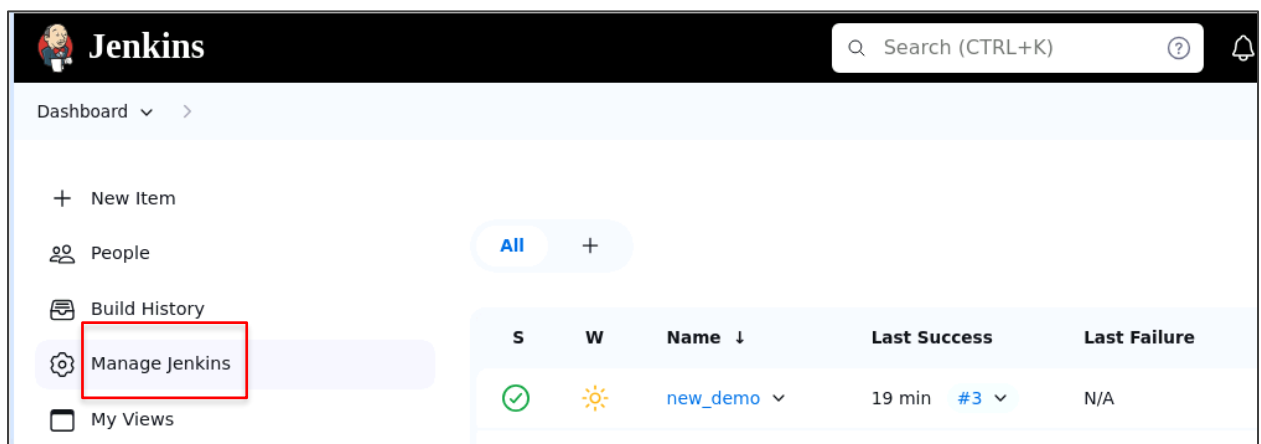


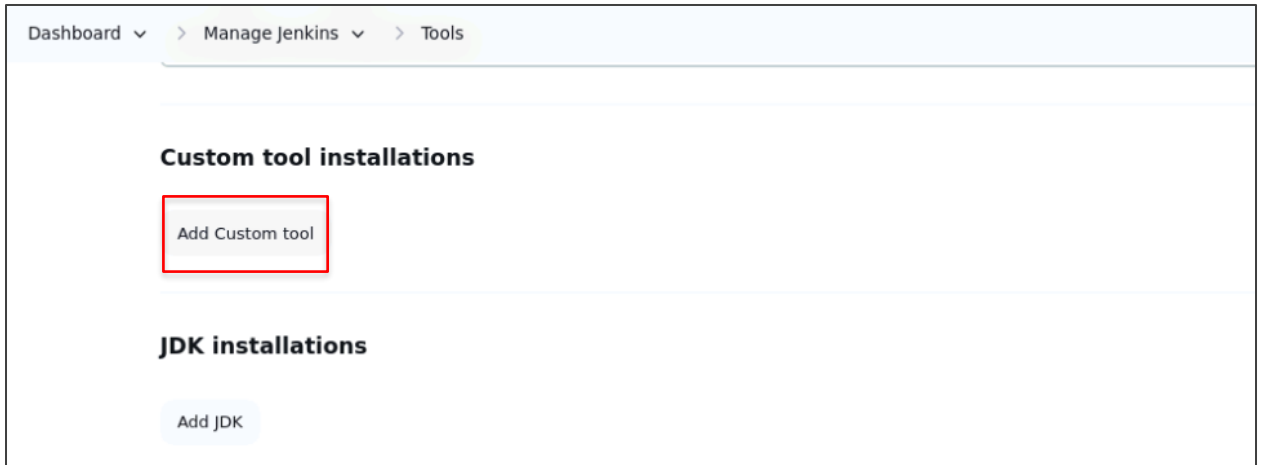
The screenshot shows the 'Plugins' page in Jenkins. The breadcrumb trail at the top reads 'Dashboard > Manage Jenkins > Plugins'. On the left, there's a sidebar with links: Updates, Available plugins (highlighted), Installed plugins, and Advanced settings. The main content area has a search bar with 'official owasp ZAP' entered. To the right of the search bar is an 'Install' button. Below the search bar, there's a table with columns: Install, Name, and Released. A single row is visible for the 'Official OWASP ZAP 1.1.0' plugin. The 'Install' column has a checkmark. The 'Name' column contains the text 'Official OWASP ZAP 1.1.0' and a description: 'The Official OWASP ZAP Jenkins Plugin extends the functionality of the ZAP security tool into a CI Environment.' The 'Released' column shows '6 yr 10 mo ago'. A warning message is displayed below the table: 'Warning: This plugin version may not be safe to use. Please review the following security notices: Credentials stored in plain text'.

1.4 In the **Available plugins**, search for **Custom Tools** and click on **Install**

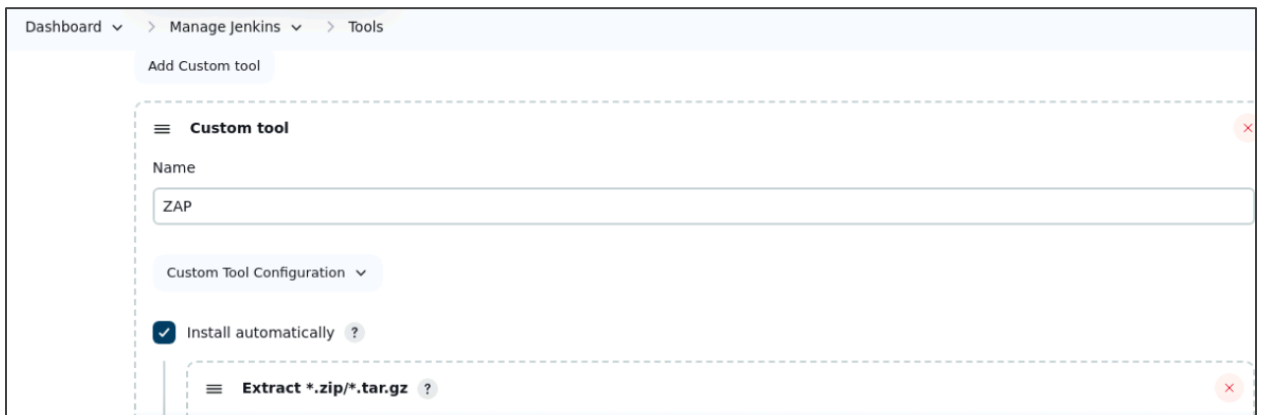


1.5 Navigate back to the **Manage Jenkins**, click on **Tools**, and under **Custom tools installations**, click on **Add Custom tool**





1.6 Under **Custom tool**, provide **ZAP** as the **Name**



- 1.7 Navigate to <https://github.com/zaproxy/zaproxy/releases>, copy the URL highlighted in the screenshot, paste it into the **Download URL for binary archive**, enter **ZAP\_2\_15** for the **Subdirectory of extracted archive** field, and then click **Save**

2 days ago  
zapbot  
v2.15.0  
124b037  
Compare

## v2.15.0 Latest

Release notes: <https://www.zaproxy.org/docs/desktop/releases/2.15.0/>

File	Checksum (SHA-256)
<a href="#">ZAP_2.15.0.dmg</a>	ae025403e46cdefff013cd0c3b88d8edc5a183a76daa63cb62c7c629005337a5
<a href="#">ZAP_2.15.0_aarch64.dmg</a>	4426253f4702bbd5fb4779bcf4d62490b2c10ec851c4ebc94ced8f156d2e5509
<a href="#">ZAP_2.15.0_Core.zip</a>	e3cf30ad526e4f3fb8a228e1d5e02da0389b2ec7436b989bb28f959703380bf5
<a href="#">ZAP_2.15.0_Crossplatform.zip</a>	05d3932a1affb0ab7987664677134709982ca3837a0b0f0e16f9aeb391933341
<a href="#">ZAP_2.15.0_Linux.tar.gz</a>	6410e196baab458a9204e29aafb5745fca003a2a6c0386f2c6e5c04b67621fa7
<a href="#">ZAP_2.15.0_unix.sh</a>	3d976a197b7f71c52c8b8b2e9f06b988384845fb854b637ebb8be0e4cb38112f
<a href="#">ZAP_2.15.0_windows-x32.exe</a>	114953f29647a5e4e5774b338f2271d6149711e9222e0b92b11be3a35b812478
<a href="#">ZAP_2.15.0_windows.exe</a>	28b348dd65116ddabbdbd98b7f84864a0bb0f98d656266f2f08bfd010ae51c57
<a href="#">bom.json</a>	04f3e148aaa406cd7f31f2be8d3ef282e16e852233da85de13aacb3a173f4740

Dashboard > Manage Jenkins > Tools

Custom Tool Configuration

☒ Install automatically

Extract \*.zip/\*.tar.gz

Download URL for binary archive

[https://github.com/zaproxy/zaproxy/releases/download/v2.15.0/ZAP\\_2.15.0\\_Linux.tar.gz](https://github.com/zaproxy/zaproxy/releases/download/v2.15.0/ZAP_2.15.0_Linux.tar.gz)

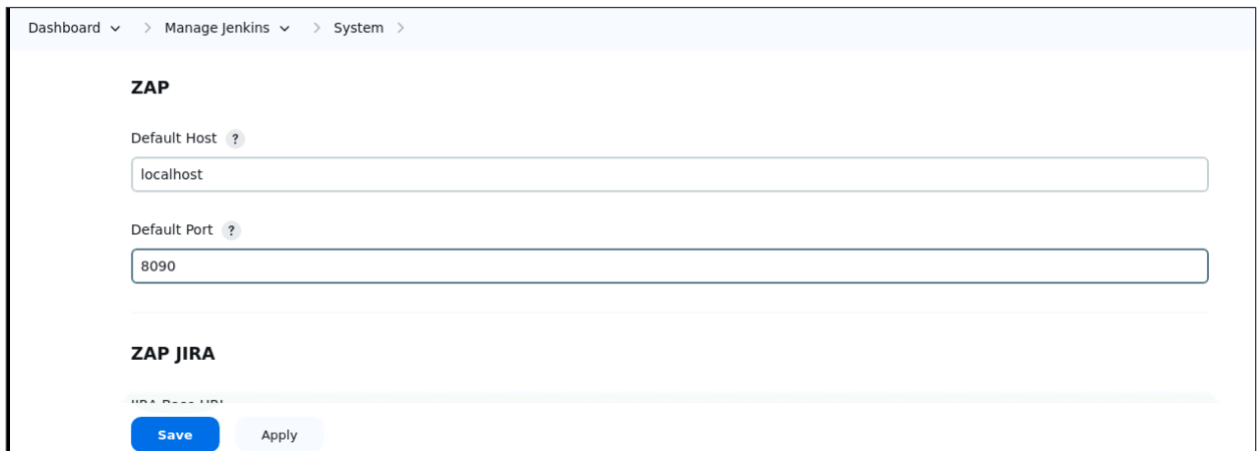
Subdirectory of extracted archive

ZAP\_2\_15

Save

Apply

- 1.8 Navigate back to the **Manage Jenkins** and select **Configure System**, scroll down to **ZAP**, and fill the **Default Host** as **localhost** and **Default Port** as **8090**



Dashboard > Manage Jenkins > System >

**ZAP**

Default Host ?  
localhost

Default Port ?  
8090

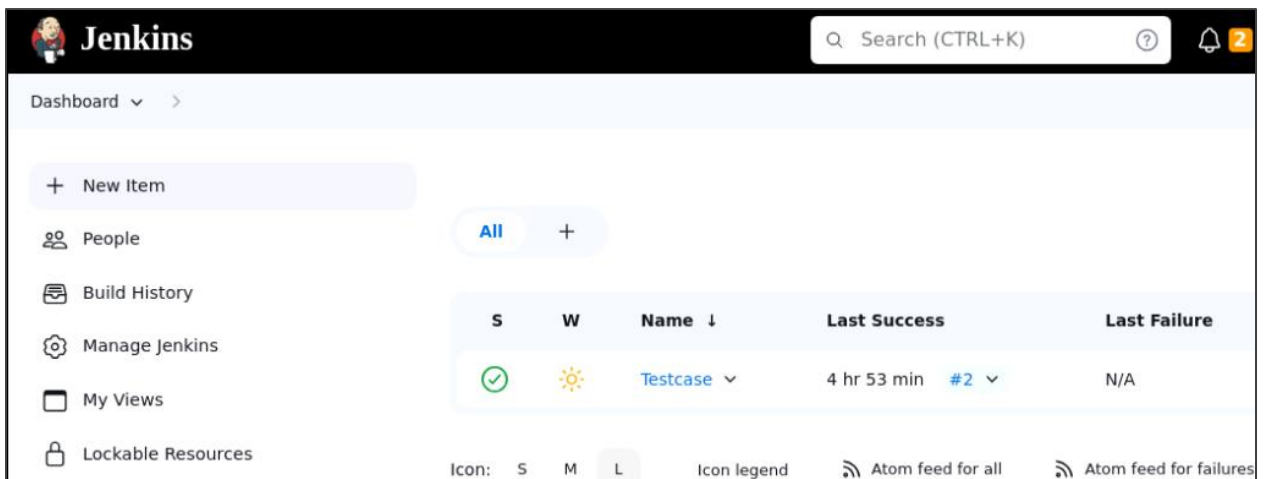
**ZAP JIRA**

JIRA URL

Save Apply

## Step 2: Create a Jenkins pipeline job to integrate the vulnerability scan tool

- 2.1 Navigate the **Jenkins Dashboard** and click on **New Item**



Jenkins

Search (CTRL+K)

Dashboard >

+ New Item

People

Build History

Manage Jenkins

My Views

Lockable Resources

S	W	Name ↓	Last Success	Last Failure
✓	⚠	Testcase	4 hr 53 min #2	N/A

Icon: S M L Icon legend Atom feed for all Atom feed for failures

2.2 Click on **Freestyle project** and put **ZAPDAST** under **Enter an item name**, then click on **OK**

The screenshot shows the Jenkins 'Enter an item name' dialog box. At the top, there is a breadcrumb trail: 'Dashboard > All >'. Below this, the title 'Enter an item name' is displayed. A text input field contains the name 'ZAPDAST', with a small note below it stating '» Required field'. Below the input field, there is a section titled 'Freestyle project' with a box icon and a description: 'This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system and can be even used for something other than software build.' At the bottom of the dialog, there is a blue 'OK' button, which is highlighted with a red rectangular box.

2.3 Navigate to the **Build Steps**, click on **Add build step**, and select **Execute ZAP**

The screenshot shows the Jenkins 'Configure' page for a project named 'ZAPDAST'. The breadcrumb trail is 'Dashboard > ZAPDAST > Configuration'. On the left, there is a sidebar with various configuration sections: 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps' (which is selected and highlighted), and 'Post-build Actions'. The main content area is titled 'Build Steps' and features an 'Add build step' button. A dropdown menu is open, showing a list of build steps. The 'Execute ZAP' option is highlighted with a red rectangular box. Other options in the list include 'Execute Windows batch command', 'Execute shell', 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets', 'Run with timeout', and 'Set build status to "pending" on GitHub commit'.

2.4 Scroll down to the **Installation Method** and select **ZAP** as the **Name**

The screenshot shows the Jenkins 'Configure' page for 'ZAPDAST', specifically the 'Installation Method' section. The breadcrumb trail is 'Dashboard > ZAPDAST > Configuration'. The left sidebar is the same as in the previous screenshot. The 'Installation Method' section has a radio button selected for 'Custom Tools Installation'. Below this, there is a 'Name' field with a question mark icon. A dropdown menu is open, showing three options: 'ZAP', 'ZAP', and 'Default'. The first 'ZAP' option is highlighted with a blue background. Below the dropdown, there is a radio button for 'System Installed: ZAP Installation Directory'.

2.5 Scroll down to the **ZAP Home Directory** and provide the path `/var/lib/jenkins/za_proxy`

The screenshot shows the 'Configure' page for ZAPDAST. On the left, there is a sidebar with navigation links: General, Source Code Management, Build Triggers, Build Environment, and Build Steps (which is highlighted). The main content area is titled 'ZAP Home Directory' and contains a 'Path' field with the value '/var/lib/jenkins/za\_proxy'. Below this, there is a 'Session Management' section with two radio buttons: 'Load Session' and 'Persist Session' (which is selected).

2.6 In the **Session Management** section, select **Persist Session** and write **Filename** as `zap_demo`

The screenshot shows the 'Configure' page for ZAPDAST. The 'Session Management' section is visible, with the 'Persist Session' radio button selected. Below this, there is a 'Filename' field with the value 'zap\_demo'. The 'ZAP Home Directory' field above it now shows the path '/var/lib/jenkins/workspace/ZAPDAST1'. At the bottom, there are 'Save' and 'Apply' buttons.

2.7 Under **Session Properties**, enter `demo_testfile` as the **Context Name**, provide `https://demo.testfire.net/` for the **Include in Context** field, and enter `^(?!https://demo.testfire.net/).*` for the **Exclude from Context** field

The screenshot shows the 'Configure' page for ZAPDAST. The 'Session Properties' section is visible. It contains three fields: 'Context Name' with the value 'demo\_testfile', 'Include in Context' with the value 'https://demo.testfire.net/', and 'Exclude from Context' which is currently empty. The 'Build Steps' sidebar item is highlighted on the left.



Dashboard > ZAPDAST > Configuration

### Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps**

Include in Context ?

https://demo.testfire.net/

Exclude from Context ?

`^(?:(!https://demo.testfire.net/).*)$`

**Save** Apply

2.8 Now, scroll down to **Attack Mode** and enter the **Starting Point** as **https://demo.testfire.net/**, and then select **Spider Scan**, **Recurse**, **AJAX Spider**, and **Active Scan**

Dashboard > ZAPDAST > Configuration

### Configure

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

#### Attack Mode

Starting Point ?

https://demo.testfire.net/

☒ Spider Scan ?

☒ Recurse ?

☐ Subtree Only ?

**Save** Apply

Dashboard > ZAPDAST > Configuration

### Configure

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

0

☒ AJAX Spider ?

☐ In Scope Only ?

☒ Active Scan ?

Policy ?

**Save** Apply

2.9 Under **Finalize Run**, select **Generate Reports**, **Clean Workspace Reports**, **Generate Reports** and select **HTML** as format

Dashboard > ZAPDAST > Configuration

**Configure**

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

**Finalize Run**

☒ Generate Reports ?

☒ Clean Workspace Reports ?

Filename ?

JENKINS\_ZAP\_VULNERABILITY\_REPORT

Save Apply

Dashboard > ZAPDAST > Configuration

**Configure**

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

JENKINS\_ZAP\_VULNERABILITY\_REPORT

☒ Generate Report

Format ?

xml  
html

☐ Export Report

Save Apply

2.10 In the **Post-build-Actions**, click on **Add post-build-action** and select **Archive the artifacts** and write **reports/\*** under **Files to archive**, then click on **Save**

Dashboard > ZAPDAST > Configuration

**Post-build Actions**

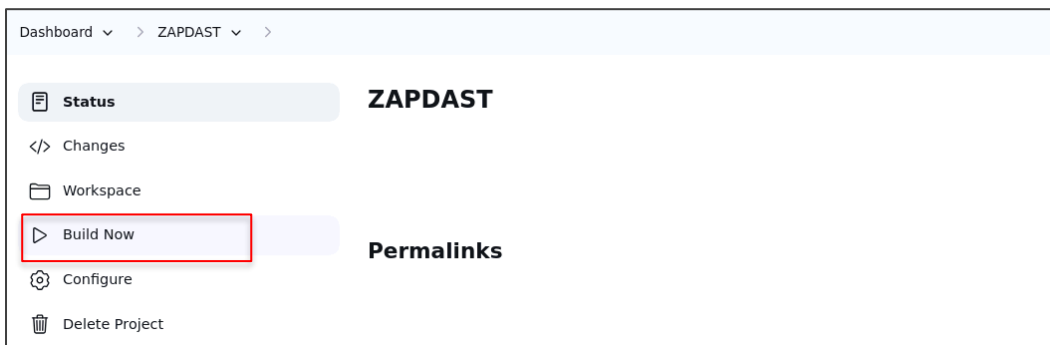
Add post-build action ^

Filter

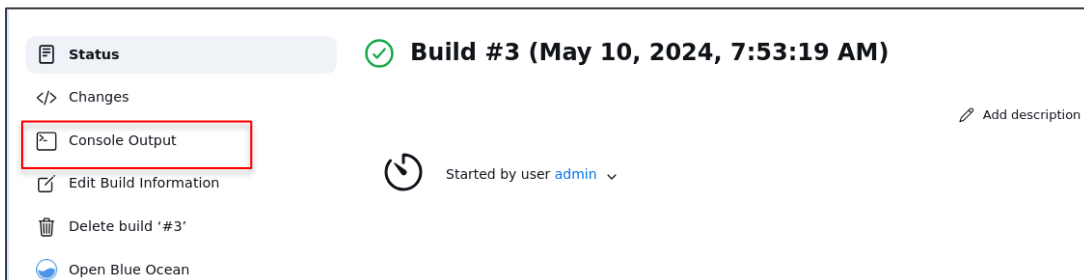
- Aggregate downstream test results
- Archive the artifacts**
- Build other projects
- Publish HTML reports
- Publish JUnit test result report
- Record fingerprints of files to track usage
- Git Publisher



2.11 Now, click on **Build Now** to execute the build



2.12 Click on **Console Output** to see the output



> Console Output

 **Console Output**

```
Started by user admin
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/ZAPDAST

[ZAP Jenkins Plugin] START PRE-BUILD ENVIRONMENT VARIABLE REPLACEMENT
HOST = [ localhost ]
PORT = [ 8090 ]

SESSION FILENAME = [  ]
INTERNAL SITES = [  ]

CONTEXT NAME = [ demo testfile ]

INCLUDE IN CONTEXT = [ https://demo.testfire.net/ ]

EXCLUDE FROM CONTEXT = [ ^(?:?!https:\\\\demo.testfire.net/).*)$. $ ]

STARTING POINT (URL) = [ https://demo.testfire.net/ ]
```

You can see that the build is configured successfully.

By following these steps, you have successfully installed and configured the OWASP ZAP plugin on Jenkins to automate the security testing of web applications during the build process.