

# **CI/CD automation Jenkins with GitHub, Maven, SonarQube, Nexus and Tomcat with free style pipelines**

## **Introduction**

Welcome to this multi-part project on setting up a basic Jenkins CI pipeline for a Maven project. Using Jenkins to execute SonarQube analysis of Java code on a Maven project is a very useful and popular setup.

## **Objectives**

- A developer push to a GitHub repo
- GitHub sends a JSON/YAML payload to Jenkins
- Jenkins triggers a simple job which fetches the source code on the branch that was just pushed to.
- Jenkins executes static code analysis.
- Jenkins pushes the analysis report to SonarQube.

## **Additionally:**

- The Jenkins job will be able to handle a multi-branch project.
- You will be able to trigger builds manually for whichever branch you choose.
- The SonarQube analysis will look include code coverage.

## **Tools Used:**

- GitHub — Source Code Management
- Maven — Build Tool
- Jenkins — Continuous Integration (CI/CD)
- SonarQube — Code Quality and Code Analysis
- Nexus — Repository Manager
- Tomcat — Application Server

## **Pre-requisites:**

- Tools like Jenkins, SonarQube, Nexus, Tomcat Server is installed and the server is up and running
- Jenkins CI/CD — SonarQube Jenkins Integration
- Jenkins CI/CD — Nexus Jenkins Integration
- Jenkins CI/CD —Tomcat Server Integration
- Make sure the following Jenkins Plugins are installed for this lab setup in the Jenkins Server — **SonarQube Scanner for Jenkins, Nexus artifact Uploader, Deploy to container**

## In This Project, We need to create 4 instances

Launch Instance											Actions
Filter by tags and attributes or search by keyword											
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6	
	nexus	i-0c3f9ce97ee1cc37	t2.medium	ap-south-1a	running	2/2 checks ...	None	ec2-52-66-212-81.ap.s...	52.66.212.61	-	
	sonar	i-0cf27d6dd69d339f	t2.medium	ap-south-1a	running	2/2 checks ...	None	ec2-65-2-181-210.ap.s...	65.2.181.210	-	
	apache tomcat	i-0eb855636183f5750	t2.micro	ap-south-1a	running	2/2 checks ...	None	ec2-43-204-22-109.ap...	43.204.22.109	-	
	JENNY	i-0fe8ed24cce5ffe63	t2.medium	ap-south-1a	running	2/2 checks ...	None	ec2-3-110-77-130.ap.s...	3.110.77.130	-	

Select an instance above

## Jenkins instance details

New EC2 Experience		Launch Instance											
EC2 Dashboard		Filter by tags and attributes or search by keyword											
Instances		Instance: i-0fe8ed24cce5ffe63 (JENNY) Public DNS: ec2-3-110-77-130.ap-south-1.compute.amazonaws.com											
Description													
Instance ID	i-0fe8ed24cce5ffe63	Public DNS (IPv4)	ec2-3-110-77-130.ap-south-1.compute.amazonaws.com	IPv4 Public IP	3.110.77.130	IPv6 IPs	-	Elastic IPs	-	-	-	-	
Instance state	running	Availability zone	ap-south-1a	Security groups	launch-wizard-134	view inbound rules	view outbound rules	Scheduled events	No scheduled events	-	-	-	
Instance type	t2.medium	Finding	Opt-in to AWS Compute Optimizer for recommendations.	Learn more	AMI ID	amzn2-ami-kernel-5.10-hvm-2.0.20220805.0-x86_64-gp2	(ami-0648966022e12a14)	Subnet ID	subnet-0b21818c80b5e653b	Network interfaces	eth0	IAM role	-
Private DNS	ip-172-31-39-82.ap-south-1.compute.internal	Platform	Amazon Linux	Platform details	Linux/UNIX	Platform operation	RunInstances	-	-	-	-	-	-
Private IPs	172.31.39.82	Secondary private IPs	VPC ID	vpc-05554d7f70805c7be	-	-	-	-	-	-	-	-	-

## And Configure security groups for Jenkins Server

sgr-07e31484bd34e9603	HTTP	TCP	80	Custom	0.0.0.0/0	Delete
sgr-02f798706e289c075	Custom TCP	TCP	8080	Custom	0.0.0.0/0	Delete
sgr-06a0ca25159518f7c	Custom TCP	TCP	8080	Custom	::/0	Delete

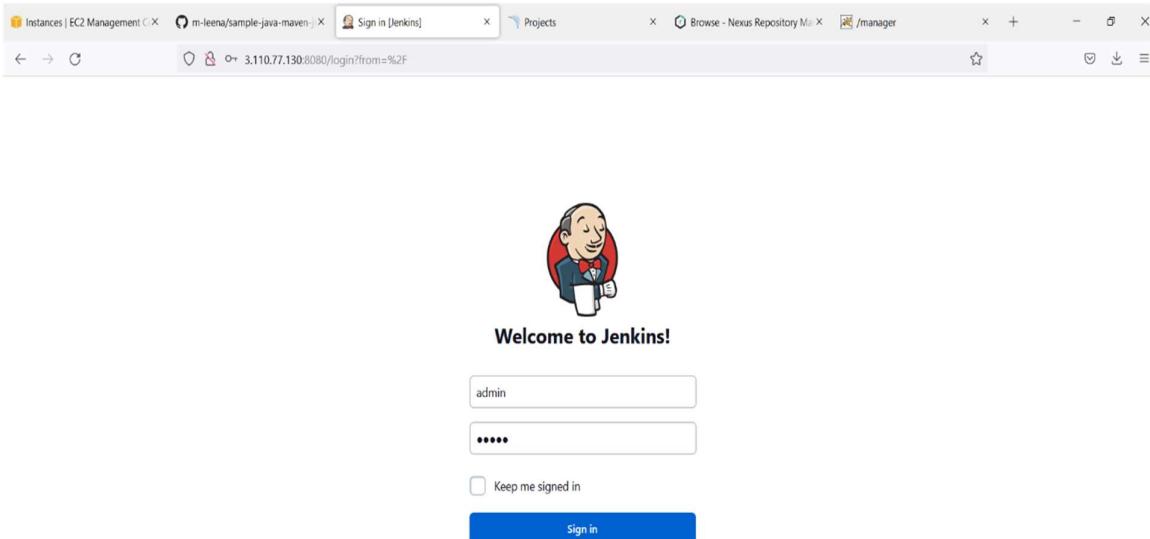
## Install Jenkins and check the status

```

root@ip-172-31-39-82:/home/ec2-user
DELL DESKTOP-BCEHACM MINGW64 ~
$ cd downloads
$ ssh -i "jenkins.pem" ec2-user@ec2-13-234-122-209.ap-south-1.compute.amazonaws.com
Last login: Sun Sep 18 11:23:45 2022
[ec2-user@ip-172-31-39-82 ~]# Amazon Linux 2 AMI
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-39-82 ~]# curl su
[ec2-user@ip-172-31-39-82 ~]# systemctl start jenkins
[jenkins.service - Jenkins Continuous Integration Server
 Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; vendor preset: disabled)
 Active: active (running) since Sun 2022-09-18 09:59:53 UTC; 3h 25min ago
 Main PID: 3006 (java)
 CGroup: /system.slice/jenkins.service
           └─3006 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080
Sep 18 13:25:20 ip-172-31-39-82.ap-south-1.compute.internal jenkins[3006]: at org.eclipse.jetty.io.FillInterest.fillable(FillInterest.java:100)
Sep 18 13:25:20 ip-172-31-39-82.ap-south-1.compute.internal jenkins[3006]: at org.eclipse.jetty.io.SelectableChannelEndPoint$1.run(SelectableChannelEndPoint.java:53)
Sep 18 13:25:20 ip-172-31-39-82.ap-south-1.compute.internal jenkins[3006]: at org.eclipse.jetty.util.thread.strategy.AdaptiveExecutionStrategy$RunnableTask.consumeTask(AdaptiveExecutionStrategy.java:412)
Sep 18 13:25:20 ip-172-31-39-82.ap-south-1.compute.internal jenkins[3006]: at org.eclipse.jetty.util.thread.strategy.AdaptiveExecutionStrategy$tryProduce(AdaptiveExecutionStrategy.java:381)
Sep 18 13:25:20 ip-172-31-39-82.ap-south-1.compute.internal jenkins[3006]: at org.eclipse.jetty.util.thread.strategy.AdaptiveExecutionStrategy$lambda$new$0(AdaptiveExecutionStrategy.java:138)
Sep 18 13:25:20 ip-172-31-39-82.ap-south-1.compute.internal jenkins[3006]: at org.eclipse.jetty.util.thread.strategy.AdaptiveExecutionStrategy$lambda$run$1(AdaptiveExecutionStrategy.java:407)
Sep 18 13:25:20 ip-172-31-39-82.ap-south-1.compute.internal jenkins[3006]: at org.eclipse.jetty.util.thread.QueuedThreadPool$Worker.run(QueuedThreadPool.java:894)
Sep 18 13:25:20 ip-172-31-39-82.ap-south-1.compute.internal jenkins[3006]: at java.base/java.lang.Thread.run(Thread.java:1038)
[ec2-user@ip-172-31-39-82 ~]# curl https://jenkins.io/doc/book/install/linux/
cat: /vmlinuz: No such file or directory
sudo wget -O /etc/yum.repos.d/jenkins.repo
http://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
sudo yum upgrade
sudo yum install jenkins
sudo systemctl daemon-reload
[ec2-user@ip-172-31-39-82 ~]#

```

## After installing Jenkins log into server



## SonarQube instance details

A screenshot of the AWS EC2 Instances page. The left sidebar shows navigation options like 'New EC2 Experience', 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', 'Instances', 'Images', 'AMIs', 'AMI Catalog', and 'Elastic Block Store'. The main area displays a table of instances. One row for 'sonar' is selected, showing its details: Instance ID i-00cf27d6dd9d339f, Public DNS ec2-65-2-181-210.ap-south-1.compute.amazonaws.com, Instance state running, Instance type t2 medium, and so on. A detailed view for the 'sonar' instance is open, showing fields like Description, Status Checks, Monitoring, Tags, and various configuration parameters such as Public DNS, Instance ID, and Security groups.

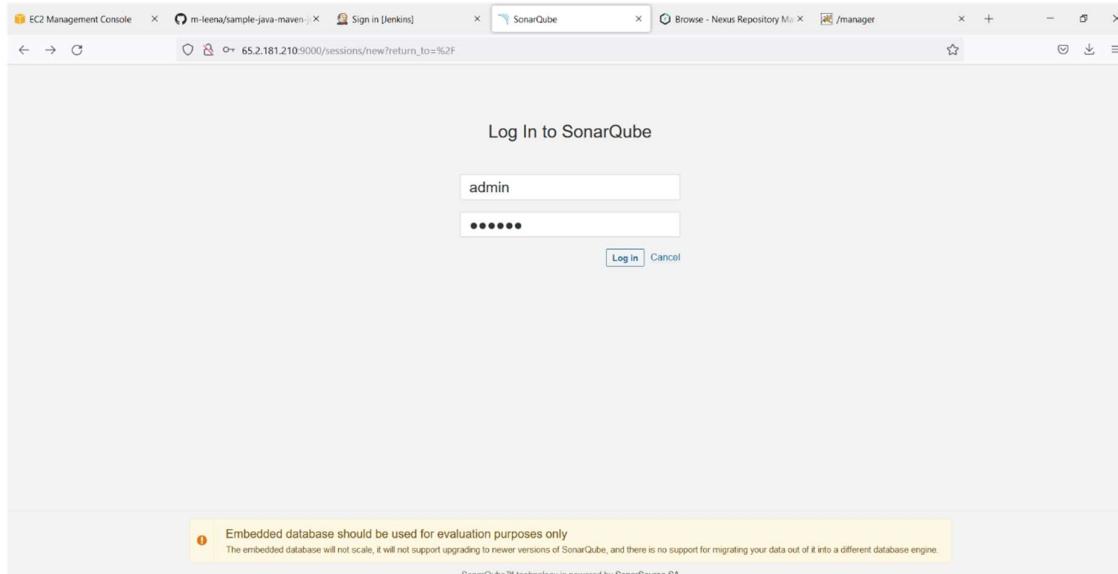
## Configure security groups for SonarQube Server

A screenshot of the AWS Security Groups 'Edit inbound rules' page. The top navigation shows 'EC2 &gt; Security Groups &gt; sg-0d0f1af80a53f5829 - launch-wizard-131 &gt; Edit inbound rules'. The main content area is titled 'Inbound rules' and contains three rules listed in a table. The first rule is for 'Custom TCP' on port 9000 from 'Custom' source. The second rule is for 'SSH' on port 22 from 'Custom' source. The third rule is for 'Custom TCP' on port 9000 from 'Custom' source. At the bottom are buttons for 'Add rule', 'Cancel', 'Preview changes', and 'Save rules'.

## Install SonarQube and check status

```
sonaradmin@ip-172-31-41-213:/opt/sonar/bin/linux-x86-64
$ cd downloads/
DELLDESKTOP-BECEHAC MINGW64 ~
$ ssh -i "jenkins.pem" ec2-user@ec2-65-2-181-210.ap-south-1.compute.amazonaws.com
The authenticity of this host can't be established.
ED25519 key fingerprint is SHA256:chtau9HkZL5qB2jVn+ke9/8d0t7r0NDeX000MEu7UE.
This host key is known to your local host, but the key for this host has changed.
It is recommended that you double-check the key for this host.
Warning: Permanently added 'ec2-65-2-181-210.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Are you sure you want to continue connecting (yes/no/[Fingerprint])? yes
Warning: Permanently added 'ec2-65-2-181-210.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Last login: Sat Sep 17 07:03:47 2022 from 124.123.189.169
[ -| _-|_ ) Amazon Linux 2 AMI
[ -| \_|-|_
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-41-213 ~]$ sudo su
[root@ip-172-31-41-213 ec2-user]# su sonaradmin
Last login: Sat Sep 17 07:03:47 2022 on pts/0
[sonaradmin@ip-172-31-41-213 ~]$ cd /opt
[sonaradmin@ip-172-31-41-213 opt]$ ls
sonar      sonar-5.6.12.1-bin.zip
[sonaradmin@ip-172-31-41-213 opt]$ ls sonar
[sonaradmin@ip-172-31-41-213 sonar]$ ls
bin        conf COPING data dependency-license.json elasticsearch extensions lib logs temp web
bin/sonar.sh license macosx-universal-64 windows-x86-64
[sonaradmin@ip-172-31-41-213 bin]$ rm bin/*
[sonaradmin@ip-172-31-41-213 bin]$ ls
lib  sonar.sh wrapper
[sonaradmin@ip-172-31-41-213 lib]$ sonar.sh
Usage: sonar.sh {start | stop | force-stop | restart | status | dump}
[sonaradmin@ip-172-31-41-213 lib]$ ./sonar.sh start
starting SonarQube...
Starting SonarQube
[sonaradmin@ip-172-31-41-213 lib]$ ./sonar.sh status
SonarQube is running (7569).
[sonaradmin@ip-172-31-41-213 lib]$
```

## After installing sonarqube login to server



## Nexus instance details

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPs
nexus	i-00c3f9ce97ee1cc37	I2 medium	ap-south-1a	running	2/2 checks ...	None	ec2-52-66-212-61.ap...	52.66.212.61	-
sonar	i-00c27d9d9d9d339f	I2 medium	ap-south-1a	running	2/2 checks ...	None	ec2-65-2-181-210.ap...	65.2.181.210	-
apache tomcat	i-0eb8556361835f750	I2 micro	ap-south-1a	running	2/2 checks ...	None	ec2-43-204-22-109.ap...	43.204.22.109	-
JENNY	i-0fe8ed24cc0e9ff63	I2 medium	ap-south-1a	running	2/2 checks ...	None	ec2-3-110-77-130.ap...	3.110.77.130	-

Instance: i-00c3f9ce97ee1cc37 (nexus) Public DNS: ec2-52-66-212-61.ap-south-1.compute.amazonaws.com

Description	Status Checks	Monitoring	Tags
Instance ID: i-00c3f9ce97ee1cc37	running		
Private DNS: ip-172-31-46-219.ap-south-1.compute.internal			
Private IPs: 172.31.46.219			
Secondary private IPs:			
VPC ID: vpc-05554db7005c7be			
Platform: Amazon Linux			
Platform details: Linux/UNIX			
User connection: Runinstances			

Public DNS (IPv4): ec2-52-66-212-61.ap-south-1.compute.amazonaws.com  
IPv4 Public IP: 52.66.212.61  
Elastic IPs: -  
Availability zone: ap-south-1a  
Security groups: launch-wizard-132, view inbound rules, view outbound rules  
Scheduled events: No scheduled events  
AMI ID: amzn2.ami-kernel.5.10-hvm-2.0.20220805-x06\_64-gp2  
(ami-06489666022e12a14)  
Subnet ID: subnet-0621819c80b5e83b  
Network interfaces: eth0  
IAM role: -

### **And Configure security groups for nexus Server**

EC2 > Security Groups > sg-0d2fb76cf22313b72 - launch-wizard-132 > Edit inbound rules

## Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>
sgr-0007d29e7891ab87a	SSH	TCP	22	Custom	<input type="text" value="0.0.0.0"/> <a href="#">X</a>
sgr-0cde6d5b4ad978ca9	Custom TCP	TCP	8081	Custom	<input type="text" value="0.0.0.0"/> <a href="#">X</a>

## Install nexus and check status

```

root@ip-172-31-46-219:/opt/nexus3/bin

[ELB02SKTOP-BEACH MINGW64 ~] cd downloads/
[ELB02SKTOP-BEACH MINGW64 ~/downloads (main)
$ ssh -i "jenkins.pem" ec2-user@ec2-52-66-212-61.ap-south-1.compute.amazonaws.co
The authenticity of host 'ec2-52-66-212-61.ap-south-1.compute.amazonaws.com (52.
66.212.61)' can't be established.
ED25519 key fingerprint is SHA256:00fffc7c2yKm+71D4UTFGtHn+8q0vwxDG45j1kM.
This host key is known by the following other names:addresses
  ->ssh/known_hosts:96: ec2-43-205-195-45.ap-south-1.compute.amazonaws.com
  ->ssh/known_hosts:106: ec2-13-235-114-248.ap-south-1.compute.amazonaws.com
  ->ssh/known_hosts:110: ec2-13-235-114-248.ap-south-1.compute.amazonaws.com
  ->ssh/known_hosts:113: ec2-35-154-110-88.ap-south-1.compute.amazonaws.com
  ->ssh/known_hosts:120: ec2-43-205-140-196.ap-south-1.compute.amazonaws.com
  ->ssh/known_hosts:132: ec2-13-235-114-248.ap-south-1.compute.amazonaws.com
  ->ssh/known_hosts:133: ec2-13-235-114-248.ap-south-1.compute.amazonaws.com
  ->ssh/known_hosts:148: ec2-3-110-150-114.ap-south-1.compute.amazonaws.com
(1 additional address omitted)
Are you sure you want to connect (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-66-212-61.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Last login: Sun Sep 18 09:59:44 2022
[ec2-user@ip-172-31-46-219 ~]$ ls
[ec2-user@ip-172-31-46-219 ~]$ ls
[ec2-user@ip-172-31-46-219 ~]$ sudo su
[root@ip-172-31-46-219 ec2-user]$ ls
nexus-3.41.1-01-unix.tar.gz
[root@ip-172-31-46-219 ~]$ cd /opt
[root@ip-172-31-46-219 opt]$ ls
aws nexus3 nexus-3.41.1-01-unix.tar.gz rh sonatype-work
[root@ip-172-31-46-219 opt]$ ls
aws nexus3 nexus-3.41.1-01-unix.tar.gz rh sonatype-work
[root@ip-172-31-46-219 opt]$ ls
bin deploy etc lib NOTICE.txt OSS-LICENSE.txt PRO-LICENSE.txt public replicator system
[root@ip-172-31-46-219 nexus3]$ cd bin
[root@ip-172-31-46-219 bin]$ ./nexus
[nexus] Service command summary: only basic LS8 actions (start, stop, restart, try-restart, reload, force-reload, status). For other actions, please try to use systemctl.
[root@ip-172-31-46-219 bin]$ service nexus start
Last login: Sun Sep 18 13:14:57 UTC 2022
[root@ip-172-31-46-219 bin]$ service nexus status
nexus is running.
[root@ip-172-31-46-219 bin]$
```

## **After installing nexus login to server**

A screenshot of a web browser window showing the Nexus Repository Manager interface. The address bar displays the URL "52.66.212.61:8081#/browse/browse". The page features a dark background with a glowing orange fractal pattern. In the lower-left corner, there is a green hexagonal logo with a white 'r' inside, followed by the text "Nexus Repository Manager". Below this, a progress bar icon with the text "Initializing ..." indicates the system is starting up.

Screenshot of the Sonatype Nexus Repository Manager (OSS 3.41.1-01) Welcome page. The page displays a message about enabling Log4j Visualizer capability in response to the CVE-2021-44228 vulnerability. It also highlights the new subdomain routing feature for Docker repositories in Nexus Repository 3.40 Pro.

**Welcome** Learn about Sonatype Nexus Repository Manager

In response to the log4j vulnerability identified in [CVE-2021-44228](#) (also known as "log4shell") impacting organizations world-wide, we are providing an experimental Log4j Visualizer capability to help our users identify log4j downloads impacted by CVE-2021-44228 so that they can mitigate the impact. Note that enabling this capability may impact Nexus Repository performance. Also note that the visualizer does not currently identify or track other log4j vulnerabilities.

[Enable Capability](#)

**What's new in Nexus Repository 3.40 Pro?**

**New repository connector type!**

In this version, we introduce subdomain routing for Docker repositories.

With subdomain routing, you no longer need to use port connectors or remember a lengthy list of port numbers. Create more easily memorable subdomains with logically assigned names instead.

You'll also experience the added benefit of avoiding the performance limitations that come with port connectors. Learn about these changes in our [help documentation](#) and check out the "[Have you heard](#)" video.



## Tomcat instance details

Screenshot of the AWS Management Console EC2 Instances page. The instance details for an Apache Tomcat instance (i-0eb855636183f5750) are shown. The instance is running on an Amazon Linux 2 micro instance type in the ap-south-1a availability zone, with a public DNS of ec2-43-204-22-109.ap-south-1.compute.amazonaws.com and a private IP of 172.31.34.190.

Description	Value
Instance ID	i-0eb855636183f5750
Instance state	running
Instance type	t2.micro
Finding	Opt-in to AWS Compute Optimizer for recommendations.
Private DNS	ip-172-31-34-190.ap-south-1.compute.internal
Private IPs	172.31.34.190
Public DNS	ec2-43-204-22-109.ap-south-1.compute.amazonaws.com
Public IPv4	43.204.22.109
IPv6 IPs	-
Elastic IPs	-
Availability zone	ap-south-1a
Security groups	launch-wizard-133, view inbound rules, view outbound rules
Scheduled events	No scheduled events
AMI ID	amzn2.ami.kernel-5.10-hvm-2.0.20220805.0-x86_64-gp2 (ami-0e4986602e12a14)
Subnet ID	subnet-0b21818c80b5e803b
Network interfaces	eth0
IAM role	-

## Security group of Tomcat

Screenshot of the AWS Management Console Security Groups page, specifically for the Tomcat security group. The inbound rules table shows three rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-011ad64a9da7d77fe	Custom TCP	TCP	81	Custom	0.0.0.0/0
sgr-079ed5ec5c6c0fa1d	Custom TCP	TCP	8080	Custom	0.0.0.0/0
sgr-035764675945718d	SSH	TCP	22	Custom	0.0.0.0/0

## Install Tomcat and check

```
root@ip-172-31-39-82:/home/ec2-user
$ cd downloads/
$ ssh -i "jenkins.pem" ec2-user@ec2-43-204-22-109.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-43-204-22-109.ap-south-1.compute.amazonaws.com (43.204.22.109)' can't be established.
ED25519 key fingerprint is SHA256:0lb5/nq2UePtm1ZxJuxFc9r5/ef6gQuInmPGfve4U.
This host key is known by the following other names/addresses:
  -./ssh/known_hosts:115: ec2-13-127-139-196.ap-south-1.compute.amazonaws.com
  -./ssh/known_hosts:119: ec2-43-205-145-124.ap-south-1.compute.amazonaws.com
  -./ssh/known_hosts:145: ec2-13-232-54-15.ap-south-1.compute.amazonaws.com
  -./ssh/known_hosts:152: ec2-65-0-92-42.ap-south-1.compute.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-43-204-22-109.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Last login: Sat Sep 17 08:08:57 2022 From 124.123.189.169
-| _ |_
-| (   / Amazon Linux 2 AMI
-| \|_|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-34-190 ~]$ sudo su
[root@ip-172-31-34-190 ec2-user]# ls
apache-tomcat-9.0.65 apache-tomcat-9.0.65.tar.gz
[root@ip-172-31-34-190 ec2-user]# cd apache-tomcat-9.0.65
[root@ip-172-31-34-190 apache-tomcat-9.0.65]# ls
bin BUILDING.txt conf CONTRIBUTING.md lib LICENSE logs NOTICE README.md RELEASE-NOTES RUNNING.txt temp webapps work
[root@ip-172-31-34-190 apache-tomcat-9.0.65]# cd bin
[root@ip-172-31-34-190 bin]# ls
bootstrap.jar catalina-tasks.xml commons-daemon.jar configtest.sh digest.sh setclasspath.bat shutdown.sh tomcat-juli.jar tool-wrapper.sh
catalina.bat ciphers.bat commons-logging-native.tar.gz daemon.sh makebase.bat setclasspath.sh startup.bat tomcat-native.tar.gz version.bat
catalina.sh catalina.sh configtest.bat digest.bat makebase.sh shutdown.bat startup.sh tool-wrapper.bat version.sh
[root@ip-172-31-34-190 bin]# ./startup.sh start
Using CATALINA_BASE: /home/ec2-user/apache-tomcat-9.0.65
Using CATALINA_HOME: /home/ec2-user/apache-tomcat-9.0.65
Using CATALINA_TMPDIR: /home/ec2-user/apache-tomcat-9.0.65/temp
Using JRE_HOME: /
Using CLASSPATH: /home/ec2-user/apache-tomcat-9.0.65/bin/bootstrap.jar:/home/ec2-user/apache-tomcat-9.0.65/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[root@ip-172-31-34-190 bin]# client_loop: send disconnect: Connection reset by peer
[ec2-user@ip-172-31-34-190 ~]$
```

## Integrate sonarqube with Jenkins

The screenshot shows the Jenkins 'Configure System' page under 'SonarQube servers'. A checkbox for enabling SonarQube server configuration as build environment variables is checked. A 'SonarQube installations' section lists one entry named 'sonarqube' with a 'Server URL' of 'http://65.2.181.210:9000/' and a 'Server authentication token' of 'sonar'. Buttons for 'Save' and 'Apply' are at the bottom.

## Integrate nexus with Jenkins

The screenshot shows the Jenkins 'Plugin Manager' page with the search bar set to 'nexus artifact uploader'. It lists the 'Nexus Artifact Uploader' plugin version 2.13, which is installed and enabled. The Jenkins version is 2.361.1.

The screenshot shows the Jenkins 'Configure System' page under 'Sonatype Nexus'. It configures a 'Nexus Repository Manager 3.x Server' with a 'Display Name' of 'jenkins-nexus', 'Server ID' of 'nexus', and 'Server URL' of 'http://52.66.212.61:8081/'. The 'Credentials' dropdown contains 'admin/\*\*\*\*\* (nexus)'. Buttons for 'Save' and 'Apply' are at the bottom.

## Integrate tomcat with Jenkins

The screenshot shows the Jenkins Plugin Manager interface. At the top, there are tabs for 'Updates', 'Available', 'Installed' (which is selected), and 'Advanced'. A search bar contains the text 'deploy'. Below the tabs, a card for the 'Deploy to container Plugin 1.16' is displayed. The card includes a description: 'This plugin allows you to deploy a war to a container after a successful build. Glassfish 3.x remote deployment'. It has a status indicator showing it is 'Enabled' with a green checkmark icon. There is also a red 'x' icon and a link to 'Report an issue with this plugin'. At the bottom right of the card, it says 'Jenkins 2.361'.

## In Jenkins first we create job for sonar qube

The screenshot shows the Jenkins job creation dialog. The title is 'Enter an item name'. A text input field contains the value 'SONAR'. Below the input field, a note says '» Required field'. There are four options listed: 'Freestyle project' (selected), 'Maven project', 'Pipeline', and 'Multi-configuration project'. Each option has a brief description and a corresponding icon. At the bottom of the dialog is a blue 'OK' button.

The screenshot shows a GitHub repository page for 'm-leena / sample-java-maven-jenkins-nexus-sonar'. The repository is public and forked from 'AzagraMac/sample-java-maven-jenkins-nexus-sonar'. The page includes sections for 'Code', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. On the right side, there is an 'About' section with details: 'Ejemplo de app en Java de integración continua', 'Readme', 'GPL-3.0 license', '0 stars', '0 watching', and '20 forks'. Below that is a 'Releases' section stating 'No releases published' and 'Create a new release'. At the bottom is a 'Packages' section stating 'No packages published' and 'Publish your first package'. A context menu is open over the repository name, showing options like 'Clone', 'HTTPS', 'SSH', 'GitHub CLI', 'Open with GitHub Desktop', and 'Download ZIP'.

[Instances | EC2 Management](#) [m-leena/sample-java-maven](#) [sonarqube Config \[Jenkins\]](#) [Projects](#) [Welcome - Nexus Repository](#) [/manager](#)

Dashboard > sonarqube >

Configuration

Discard old builds ?

GitHub project

Project url ?

Disable Automated Maven Repository Cleanup

This project is parameterized ?

Throttle builds ?

Execute concurrent builds if necessary ?

Source Code Management

None

File System

---

[Instances | EC2 Management](#) [m-leena/sample-java-maven](#) [sonarqube Config \[Jenkins\]](#) [Projects](#) [Welcome - Nexus Repository](#) [/manager](#)

Dashboard > sonarqube >

Configuration

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Repository URL ?

Credentials ?

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

---

[Instances | EC2 Management](#) [m-leena/sample-java-maven](#) [sonarqube Config \[Jenkins\]](#) [Projects](#) [Welcome - Nexus Repository](#) [/manager](#)

Dashboard > sonarqube >

Configuration

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Inspect build log for published Gradle build scans

Prepare SonarQube Scanner environment ?

Server authentication token  
SonarQube authentication token. Mandatory when anonymous access is disabled. Will default to the one defined in the SonarQube installation.

SSH Agent

Terminate a build if it's stuck

With Ant ?

Build Steps

Invoke top-level Maven targets ?

Maven Version

Goals

SonarQube Configuration Jenkins Job

**Configuration**

**Build Steps**

**Invoke top-level Maven targets**

Maven Version: (Default)

Goals:

```
clean
sonar:sonar
```

**Save** **Apply**

Jenkins Console Output for SonarQube #9 Build

Started by user Leena M  
Running as SYSTEM  
Building in workspace /var/lib/jenkins/workspace/sonarqube  
The recommended git tool is: NONE  
No credentials specified

```
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/sonarqube/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/m-leena/sample-java-maven-jenkins-nexus-sonar.git # timeout=10
Fetching upstream changes from https://github.com/m-leena/sample-java-maven-jenkins-nexus-sonar.git
> git --version # timeout=10
> git -v # git version 2.37.1'
> git fetch --tags --force +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 344df1b09e33935091098442c57d0bbb4fe9db74 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 344df1b09e33935091098442c57d0bbb4fe9db74 # timeout=10
Commit message: "Update pom.xml"
> git rev-list --no-walk 158287931493f629b5cdaff3bf1fd0456ce2df03 # timeout=10
Injecting SonarQube environment variables using the configuration: sonarqube
[sonarqube] $ mvn clean sonar:sonar
[INFO] Scanning for projects...
[INFO]
```

Jenkins Console Output for SonarQube #9 Build

[INFO] Sensor Zero Coverage Sensor  
[INFO] Sensor Zero Coverage Sensor (done) | time=6ms  
[INFO] Sensor Java CPD Block Indexer  
[INFO] Sensor Java CPD Block Indexer (done) | time=14ms  
[INFO] SCM Publisher SCM provider for this project is: git  
[INFO] SCM Publisher 2 source files to be analyzed  
[INFO] SCM Publisher 2/2 source files have been analyzed (done) | time=103ms  
[INFO] CPD Executor 1 file had no CPD blocks  
[INFO] CPD Executor Calculating CPD for 0 files  
[INFO] CPD Executor CPD calculation finished (done) | time=0ms  
[INFO] Analysis report generated in 84ms, dir size=100 KB  
[INFO] Analysis report compressed in 26ms, zip size=16 KB  
[INFO] Analysis report uploaded in 9ms  
[INFO] ANALYSIS SUCCESSFUL, you can browse http://65.2.181.210:9000/dashboard?id=com.web.ca1X3AWebAppCal  
[INFO] Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report  
[INFO] More about the report processing at http://65.2.181.210:9000/api/ce/task?id=AyNRE0\_X0MmAi6or7L  
[INFO] Analysis total time: 7.292 s  
[INFO] -----
[INFO] BUILD SUCCESS  
[INFO] -----
[INFO] Total time: 9.284s  
[INFO] Finished at: Sun Sep 18 14:49:23 UTC 2022  
[INFO] Final Memory: 14M/50M  
[INFO] -----
Finished: SUCCESS

**sonarcube** Projects Issues Quality Profiles Quality Gates Administration

My Favorites All

Filters

Quality Gate

- Passed: 3
- Failed: 0

Reliability (Bug)

- 0: 2
- 1: 0
- 2: 0
- 3: 1
- 4: 0
- 5: 0

Security (Vulnerabilities)

- 0: 3
- 1: 0
- 2: 0
- 3: 0
- 4: 0
- 5: 0

Security Review (Security Hotspots)

- > 80%: 1
- 70% - 80%: 0
- 50% - 70%: 0
- 30% - 50%: 0

3 projects

Perspective: Overall Status Sort by: Name

Last analysis: yesterday

Bugs: 0 A Vulnerabilities: 0 A Hotspots Reviewed: - A Code Smells: 1 A Coverage: 0.0% Duplications: 0.0% Lines: 65 XML

Last analysis: yesterday

Bugs: 0 A Vulnerabilities: 0 A Hotspots Reviewed: - A Code Smells: 0 A Coverage: 0.0% Duplications: 0.0% Lines: 29 XML

Last analysis: 19 minutes ago

Bugs: 1 D Vulnerabilities: 0 A Hotspots Reviewed: - A Code Smells: 3 A Coverage: 0.0% Duplications: 0.0% Lines: 41 XML, Java

3 of 3 shown

## Now, we create job for nexus

Dashboard > All >

Enter an item name

nex  
nexus

**Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

**Maven project**  
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

**Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

**Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

**OK**

Instances | EC2 Management < x m-leena/sample-java-maven < sonarqube #9 Console [Jenkins] < Projects < Problem loading page < /manager

Dashboard > nexus >

**Configuration**

**General**

Discard old builds ?  
 Github project  
 Project url ?  
 https://github.com/m-leena/sample-java-maven-jenkins-nexus-sonar.git/

Disable Automated Maven Repository Cleanup  
 This project is parameterized ?  
 Throttle builds ?  
 Execute concurrent builds if necessary ?

**Source Code Management**

None  
 File System

**Save** **Apply**

Instances | EC2 Management

m-leena/sample-java-maven -> nexus Config [Jenkins]

Projects

Problem loading page

/manager

Dashboard > nexus >

## Configuration

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

Invoke top-level Maven targets

Maven Version: (Default)

Goals: install -DskipTest

POM:

```
SNAP-REPO=Sample-snapshot  
NEXUS-USER=admin  
NEXUS-PASS=admin7  
RELEASE-REPO=sample-release  
NEXUSIP=172.31.46.219  
NEXUSPORT=8081
```

JVM Options:

Save Apply

Instances | EC2 Management

m-leena/sample-java-maven -> nexus Config [Jenkins]

Projects

Problem loading page

/manager

Dashboard > nexus >

## Configuration

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

inject ouro vanadis

Use private Maven repository

Settings file: Settings file in filesystem

File path: /etc/maven/settings.xml

Global Settings file: Global settings file on filesystem

File path:

Add build step

Save Apply

Instances | EC2 Management

m-leena/sample-java-maven -> New Item [Jenkins]

Projects

Welcome - Nexus Repository

/manager

Sign In

Sign In

admin

Sign In Cancel

What's new in Nexus Repository 3.40 Pro?

New repository connector type!

In this version, we introduce subdomain routing for Docker repositories.

With subdomain routing, you no longer need to remember a lengthy list of port numbers. Create logically assigned names instead.

You'll also experience the added benefit of port connectors. Learn more about them and check out the "Have you heard" video.

Open Source Attacks on the Rise: Top 8 Malicious Packages Found in npm

Read More...

Help us understand your needs!

Let us get to know you!

English

Name	Type	Format	Status	URL	Health check	IQ Policy Vl...
maven-central	proxy	maven2	Online - Ready to Connect		Analyze	
maven-public	group	maven2	Online			
maven-releases	hosted	maven2	Online			
maven-snapshots	hosted	maven2	Online			
nuget-group	group	nuget	Online			
nuget-hosted	hosted	nuget	Online			
nuget.org-proxy	proxy	nuget	Online - Ready to Connect		Analyze	
sample-release	hosted	maven2	Online			
sample-snapshot	hosted	maven2	Online			

## Now, we create a job for tomcat

Enter an item name

tomcat > Required field

**Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

**Maven project**  
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

**Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

**Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

**OK**

Configuration

**General**

Source Code Management

Discard old builds ?

GitHub project

Project url ?  
https://github.com/m-leena/sample-java-maven-jenkins-nexus-sonar.git

Advanced...

Disable Automated Maven Repository Cleanup

This project is parameterized ?

Throttle builds ?

Execute concurrent builds if necessary ?

Advanced...

Source Code Management

**Save** **Apply**

Instances | EC2 Management | m-leena/sample-java-maven | tomcat Config [Jenkins] | Projects | Browse - Nexus Repository Ma | /manager

Dashboard > tomcat >

## Configuration

General

**Source Code Management**

Build Triggers

Build Environment

Build Steps

Post-build Actions

Git

Repositories

Repository URL: https://github.com/m-leena/sample-java-maven-jenkins-nexus-sonar.git

Credentials: none

+ Add

Advanced...

Add Repository

Branches to build

Branch Specifier (blank for 'any'): ...

Save Apply

Instances | EC2 Management | m-leena/sample-java-maven | tomcat Config [Jenkins] | Projects | Browse - Nexus Repository Ma | /manager

Dashboard > tomcat >

## Configuration

General

Source Code Management

Build Triggers

Build Environment

**Build Steps**

Post-build Actions

### Build Steps

Invoke top-level Maven targets

Maven Version: (Default)

Goals: clean package

Advanced...

Add build step ▾

### Post-build Actions

Deploy war/ear to a container

WAR/EAR files

Save Apply

Instances | EC2 Management | m-leena/sample-java-maven | tomcat Config [Jenkins] | Projects | Browse - Nexus Repository Ma | /manager

Dashboard > tomcat >

## Configuration

General

Source Code Management

Build Triggers

Build Environment

Build Steps

**Post-build Actions**

### Deploy war/ear to a container

WAR/EAR files: target/\*.war

Context path: tomcat

Containers

#### Tomcat 9.x Remote

Credentials: admin/\*\*\*\*\*\*\*\* (tomcat9.0)

+ Add

Tomcat URL: http://43.204.22.109:8080/

Save Apply

The screenshot shows a browser window with multiple tabs open. The active tab is '43.204.22.109:8080/manager/html'. The page title is 'Tomcat Web Application Manager'. The main content area is titled 'Applications' and lists the following 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
/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	0	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

A 'Deploy' section at the bottom allows for deploying a directory or WAR file located on the server.

If we refresh the tomcat server this is the required output

Hello World Deploying Java Project in Tomcat Webserver