

How to Integrate Tomcat Server in CICD Pipeline

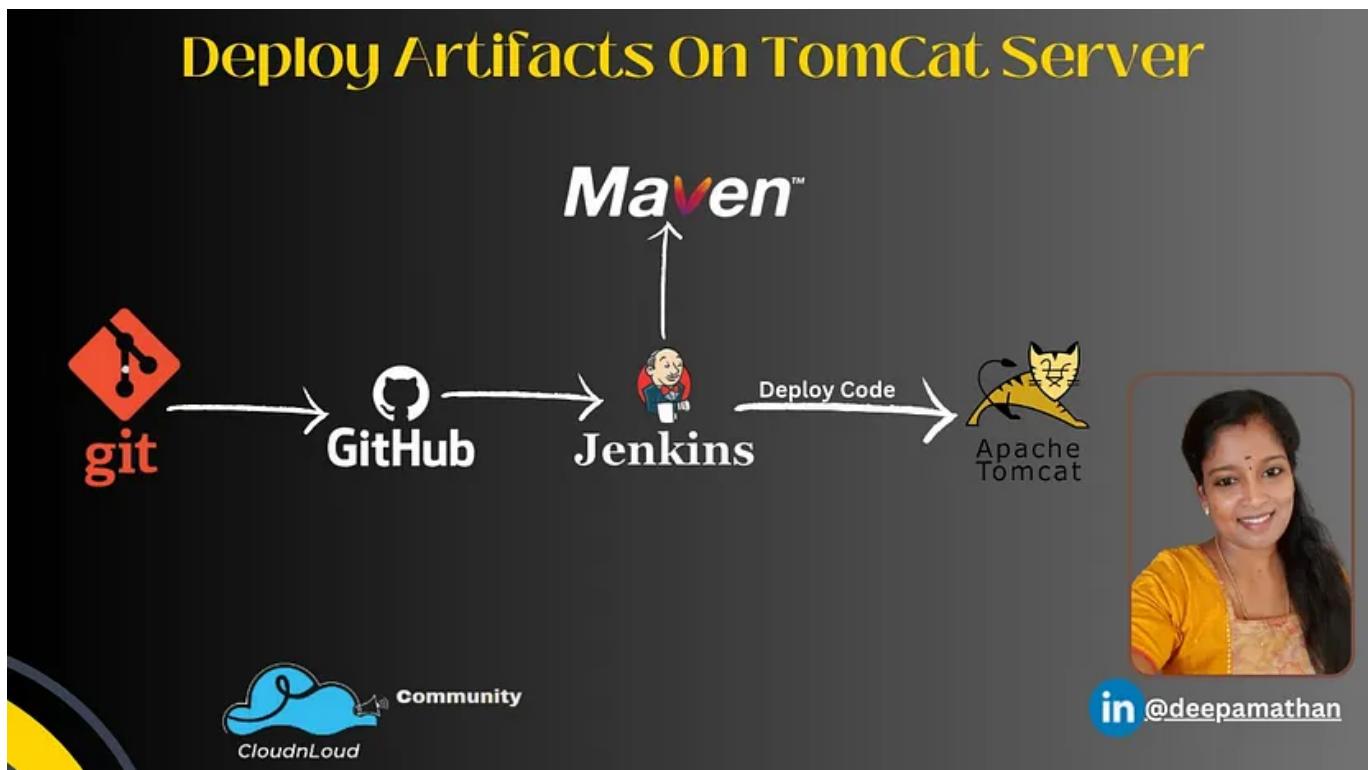


Deepa Mathan · [Follow](#)

6 min read · Sep 21

Listen

Share



In this blog, you can get an idea on how to deploy an artifact on Tomcat server, its a simple job Integrating Apache Tomcat server into a Continuous Integration/Continuous Deployment (CICD) pipeline involves automating the deployment and management of your web application on the Tomcat server.

In this example, Maven is used to build the application and generate a WAR file. The WAR file is then deployed to the Tomcat server. For that have to setup CI/CD pipeline with Github, Jenkins, Maven and Tomcat.

A step-by-step guide to Deploying Artifacts on a Tomcat Server

Deploying artifacts on a Tomcat server typically involves deploying web applications (WAR files) onto the Tomcat container. Here's a step-by-step guide to deploying artifacts on a Tomcat server:

- ✳️ Prepare your artifact
- ✳️ Access Tomcat Manager
- ✳️ Access the Tomcat Manager URL

 Login Tomcat

 Deploy the Artifact

 Monitor Deployment

 Access your application

Lets dive into the detailed view on how to deploy the artifact in target environment which we using here is Tomcat server.

Table of Contents:

 Installation of Java

 Configure Tomcat

 Create EC2 instance

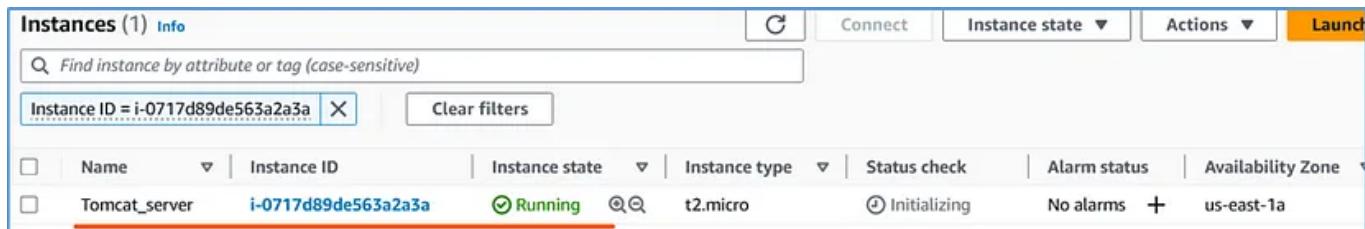
 Start Tomcat Server

 Setup Jenkins Job to deploy code on tomcat server

 Access web page on port 8080

Installation of Java:

Step 1: Create Ec2 instance, here taken linux machine connected via mobaxtrem.



The screenshot shows the AWS CloudWatch Instances console. At the top, it says "Instances (1) Info". Below that is a search bar with placeholder text "Find instance by attribute or tag (case-sensitive)". Underneath is a filter bar with "Instance ID = i-0717d89de563a2a3a" and a "Clear filters" button. The main table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. A single row is selected for "Tomcat_server" with Instance ID "i-0717d89de563a2a3a", which is "Running". The "Status check" shows "Initializing" and "No alarms". The "Availability Zone" is "us-east-1a".

Step 2: To install Tomcat server, primarily we need to install Java using the command “amazon-linux-extras”

```
$ amazon-linux-extras install java-openjdk11
```

```
[root@ip-172-31-88-55 ec2-user]# amazon-linux-extras install java-openjdk11
Installing java-11-openjdk . . .
```

Step 3: Next, in setup of Tomcat Server, have to get the tomcat download link from website [URL](#).

Copy the latest version Tomcat 9 link tar.gz file shown in screenshot below.

← → C 🔒 tomcat.apache.org/download-90.cgi

Tomcat 8.5
Tomcat Connectors
Tomcat Native 2
Tomcat Native 1.2
Wiki
Migration Guide
Presentations
Specifications

Problems?
Security Reports
Find help
FAQ
Mailing Lists

9.0.76

Please see the [README](#) file for packaging information. It explains what every distribution contains.

Binary Distributions

- Core:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
 - [32-bit Windows zip \(pgp, sha512\)](#)
 - [64-bit Windows zip \(pgp, sha512\)](#)
 - [32-bit/64-bit Windows Service Installer \(pgp, sha512\)](#)
- Full documentation:

Step 4: Using the command wget<tomcat download link>

```
wget<tomcat download link>
wget https://downloads.apache.org/tomcat/tomcat-9/v9.0.80/bin/apache-tomcat-9.0.80.tar.gz.sha512
```

```
[root@ip-172-31-88-55 /]# cd /opt
[root@ip-172-31-88-55 opt]# wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.76/bin/apache-tomcat-9.0.76.tar.gz
--2023-07-06 14:12:26-- https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.76/bin/apache-tomcat-9.0.76.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org) ... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11677655 (11M) [application/x-gzip]
Saving to: 'apache-tomcat-9.0.76.tar.gz'

100%[=====] 2023-07-06 14:12:26 (153 MB/s) - 'apache-tomcat-9.0.76.tar.gz' saved [11677655/11677655]
```

Step 5: Unzip/extract the .tar file using the command “tar -xvzf <file name>”

```
tar -xvzf <file name>
tar -xvzf apache-tomcat-9.0.76.tar.gz
```

```
2023-07-06 14:12:26 (153 MB/s) - 'apache-tomcat-9.0.76.tar.gz' saved [11677655/11677655]
[root@ip-172-31-88-55 opt]# tar -xvzf apache-tomcat-9.0.76.tar.gz
```

Step 6: After extracting, This is optional, if required you can change the name using the command mv <oldname> <newname>

```
mv <oldname> <newname>
mv apache-tomcat-9.0.76 tomcat
```

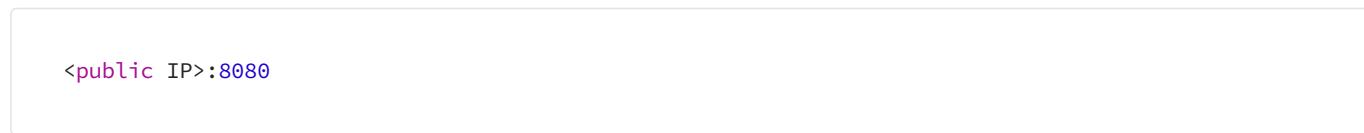
```
[root@ip-172-31-88-55 opt]# ll
total 11404
drwxr-xr-x 9 root root      220 Jul  6 14:15 apache-tomcat-9.0.76
-rw-r--r-- 1 root root 11677655 Jun  5 07:32 apache-tomcat-9.0.76.tar.gz
drwxr-xr-x 4 root root      33 Jun 28 19:10 aws
drwxr-xr-x 2 root root      6 Aug 16 2018 rh
[root@ip-172-31-88-55 opt]# mv apache-tomcat-9.0.76
mv: missing destination file operand after 'apache-tomcat-9.0.76'
Try 'mv --help' for more information.
[root@ip-172-31-88-55 opt]# mv apache-tomcat-9.0.76 tomcat
[root@ip-172-31-88-55 opt]# ll
total 11404
-rw-r--r-- 1 root root 11677655 Jun  5 07:32 apache-tomcat-9.0.76.tar.gz
drwxr-xr-x 4 root root      33 Jun 28 19:10 aws
drwxr-xr-x 2 root root      6 Aug 16 2018 rh
drwxr-xr-x 9 root root    220 Jul  6 14:15 tomcat
[root@ip-172-31-88-55 opt]#
```

Step 7: Now go inside tomcat and within that we have a bin directory to start the tomcat services.

In bin directory, if you listed you get startup.sh script which we need to run “./startup.sh” to start the tomcat services.

```
[root@ip-172-31-88-55 tomcat]# cd bin
[root@ip-172-31-88-55 bin]# ll
total 904
-rw-r----- 1 root root  35212 Jun  5 07:17 bootstrap.jar
-rw-r----- 1 root root  16852 Jun  5 07:17 catalina.bat
-rwxr-x--- 1 root root  25323 Jun  5 07:17 catalina.sh
-rw-r----- 1 root root   1664 Jun  5 07:17 catalina-tasks.xml
-rw-r----- 1 root root   2123 Jun  5 07:17 ciphers.bat
-rwxr-x--- 1 root root   1997 Jun  5 07:17 ciphers.sh
-rw-r----- 1 root root  25661 Jun  5 07:17 commons-daemon.jar
-rw-r----- 1 root root 214214 Jun  5 07:17 commons-daemon-native.tar.gz
-rw-r----- 1 root root   2040 Jun  5 07:17 configtest.bat
-rwxr-x--- 1 root root  1922 Jun  5 07:17 configtest.sh
-rwxr-x--- 1 root root   9100 Jun  5 07:17 daemon.sh
-rw-r----- 1 root root   2091 Jun  5 07:17 digest.bat
-rwxr-x--- 1 root root   1965 Jun  5 07:17 digest.sh
-rw-r----- 1 root root   3606 Jun  5 07:17 makebase.bat
-rwxr-x--- 1 root root  3382 Jun  5 07:17 makebase.sh
-rw-r----- 1 root root  3814 Jun  5 07:17 setclasspath.bat
-rwxr-x--- 1 root root  4317 Jun  5 07:17 setclasspath.sh
-rw-r----- 1 root root  2020 Jun  5 07:17 shutdown.bat
-rwxr-x--- 1 root root  1902 Jun  5 07:17 shutdown.sh
-rw-r----- 1 root root  2022 Jun  5 07:17 startup.bat
-rwxr-x--- 1 root root  1904 Jun  5 07:17 startup.sh
-rw-r----- 1 root root 48970 Jun  5 07:17 tomcat-juli.jar
-rw-r----- 1 root root 438373 Jun  5 07:17 tomcat-native.tar.gz
-rw-r----- 1 root root   4574 Jun  5 07:17 tool-wrapper.bat
-rwxr-x--- 1 root root  5540 Jun  5 07:17 tool-wrapper.sh
-rw-r----- 1 root root  2026 Jun  5 07:17 version.bat
-rwxr-x--- 1 root root  1908 Jun  5 07:17 version.sh
[root@ip-172-31-88-55 bin]# ./startup.sh
Using CATALINA_BASE: /opt/tomcat
Using CATALINA_HOME: /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME: /
Using CLASSPATH: /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[root@ip-172-31-88-55 bin]#
```

Tomcat started successfully. We can check it in browser via port 8080



Step 8: In Tomcat web page, click on Manager APP and login with your login username and password which we created.

Now we have configured our tomcat server successfully.

Step 9: Next have to integrate tomcat with Jenkins to deploy the code in tomcat server. First need to install plugins and then configure tomcat.

Step 10: Install Deploy to Container plugins. To install plugins go to “manage Jenkins” search for the available plugins “deploy to container” and install it without restart.

The screenshot shows the Jenkins Plugins management interface. The left sidebar has tabs for 'Updates', 'Available plugins' (which is selected), 'Installed plugins', and 'Advanced settings'. A search bar at the top right contains the text 'deploy'. The main area is titled 'Plugins' and lists the 'Deploy to container' plugin by 'Artifact Uploader'. The plugin version is 1.16, released 2 years and 8 months ago. It is described as allowing deployment of a war to a container after a successful build, specifically for Glassfish 3.x remote deployment. Below this, the 'Docker Pipeline' plugin is listed, released 7 months and 7 days ago, with a note that it is up for adoption. At the bottom of the list are buttons for 'Install without restart' (highlighted in blue) and 'Download now and install after restart', along with status information: 'Update information obtained: 25 min ago' and 'Check now'.

Step 11: Next to Configure tomcat server with credentials, again go to manage Jenkins then click on credentials in that click on “Global Credentials” here you can add new credentials.

The screenshot shows the Jenkins Global credentials management interface. The left sidebar shows a path: Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted). The main title is 'Global credentials (unrestricted)'. There is a button '+ Add Credentials' in the top right. Below the title, a note says 'Credentials that should be available irrespective of domain specification to requirements matching.' A table lists one credential entry:

ID	Name	Kind	Description
tomcat_deployer	deployer/******** (tomcat_deployer)	Username with password	tomcat_deployer

New credentials

Kind

Username with password

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Username ?

deployer

Treat username as secret ?

Password ?

Step 12: Next have to create the new job that to deploy the code on tomcat server. I have created the job in the name of deplytomcatjob. In creating a job have selected the maven project, then updated the git repository URL in SCM.

Enter an item name

buildndeployjob

» 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

OK

Update the github url where the code is available to deploy in tomcat server.

Source Code Management

None

Git [?](#)

Repositories [?](#)

Repository URL [?](#)

```
https://github.com/deepa-mathan/job1.git
```

Credentials [?](#)

```
- none -
```

Step 13: Set the goals as clean and install, to clean the previous build cache information and install.

Build

Root POM [?](#)

```
pom.xml
```

Goals and options [?](#)

```
clean install
```

Advanced [▼](#)

Step 14: Next click on post build actions, where we can deploy our code on tomcat server. Under post-build actions choose Deploy war/ear to a container. This option to be enabled because we have already installed deploy to container.

Post-build Actions

☰ Deploy war/ear to a container X

WAR/EAR files ?

webapp/target/webapp.war

War/ear files to deploy. Relative to the workspace root. You can also specify Ant-style GLOBs, like “**/*.war”

(from [Deploy to container Plugin](#))

Context path ?

We can give the full path of the .war file or else can mention like War/ear files to deploy. Relative to the workspace root. You can also specify Ant-style GLOBs, like “**/*.war” so that it will can wherever the file available.

Step 15: Next add the containers choose Tomcat 8.x Remote then add credential which we already added to deploy container and tomcat url.

Containers

☰ Tomcat 8.x Remote X

Credentials

deployer/******** (tomcat_deployer)

Add ▾

Tomcat URL ?

http://54.175.219.199:8080

Advanced ▾

Add Container ▾

Step 16: Finally click on Apply and save the details then click on Build now to build the created job.

Maven project deploytomcatjob

Status: Green

To deploy in tomcat server

Changes: None

Workspace: None

Build Now: None

Configure: None

Delete Maven project: None

Modules: None

Rename: None

Build History trend ▾

Filter builds... /

#2 Sep 21, 2023, 4:26 PM

#1 Sep 21, 2023, 4:25 PM

Latest Test Result (no failures)

Test Result Tree

- Passed: 2
- Skipped: 0
- Failed: 0

Permalinks

- Last build (#2), 2 min 21 sec ago
- Last stable build (#2), 2 min 21 sec ago
- Last successful build (#2), 2 min 21 sec ago
- Last completed build (#2), 2 min 21 sec ago

```
[INFO] Maven Project ..... SUCCESS [ 1.342 s]
[INFO] Server ..... SUCCESS [ 5.497 s]
[INFO] Webapp ..... SUCCESS [ 1.960 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 11.193 s
[INFO] Finished at: 2023-09-21T16:26:35Z
[INFO] -----
[JENKINS] Archiving /var/lib/jenkins/workspace/deploytomcatjob/webapp/pom.xml to com.example.maven-project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/deploytomcatjob/webapp/target/webapp.war to com.example.maven-project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.war
[JENKINS] Archiving /var/lib/jenkins/workspace/deploytomcatjob/server/pom.xml to com.example.maven-project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/deploytomcatjob/server/target/server.jar to com.example.maven-project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.jar
[JENKINS] Archiving /var/lib/jenkins/workspace/deploytomcatjob/pom.xml to com.example.maven-project/maven-project/1.0-SNAPSHOT/maven-project-1.0-SNAPSHOT.pom
channel stopped
[DeployPublisher][INFO] Attempting to deploy 1 war file(s)
[DeployPublisher][INFO] Deploying /var/lib/jenkins/workspace/deploytomcatjob/webapp/target/webapp.war to container Tomcat 8.x Remote with context null
  Redeploying [/var/lib/jenkins/workspace/deploytomcatjob/webapp/target/webapp.war]
  Undeploying [/var/lib/jenkins/workspace/deploytomcatjob/webapp/target/webapp.war]
  Deploying [/var/lib/jenkins/workspace/deploytomcatjob/webapp/target/webapp.war]
Finished: SUCCESS
```

Step 17: Job sucessfully created the same can be seen in Build history. The same you can check in tomcat server where the artifact webapp.war file is available.

```
[root@ip-172-31-88-55 tomcat]# cd webapps
[root@ip-172-31-88-55 webapps]# ll
total 8
drwxr-x--- 16 root root 4096 Jul  6 14:15 docs
drwxr-x---  7 root root   99 Jul  6 14:15 examples
drwxr-x---  6 root root   79 Jul  6 14:15 host-manager
drwxr-x---  6 root root  114 Jul  6 14:15 manager
drwxr-x---  3 root root  223 Jul  6 14:15 ROOT
drwxr-x---  4 root root   54 Sep 21 16:26 webapp
-rw-r-----  1 root root 2255 Sep 21 16:26 webapp.war
[root@ip-172-31-88-55 webapps]#
```

After successful deployment, you can check the same in tomcat webpage, where one more webapp directory got created.

Tomcat Web Application Manager

Message:		OK			
Manager					
List Applications	HTML Manager Help			Manager Help	Server Status
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
/webapp	None specified	Webapp	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy
Deploy directory or WAR file located on server

Step 18: If you click on the “webapp” directory it redirects to the deployed app webpage output is given below.

54.175.219.199:8080/webapp/

Welcome to Cloudnloud

Please fill this form to register

Enter Name Enter Full Name
Enter mobile Enter mobile number
Enter Email Enter Email
Skillset Enter Skills
Career Goal goal

[Register](#)

Thankyou, Happy Learning

Hope you got a complete process on how to deploy a simple html code using CICD Pipeline integrated on Tomcat server.

In the next blog will share on how to automate the job whenever there is a change in code commit, how can we schedule and the various options in build triggers to make the job automated

Let's keep the conversation going! Your engagement through claps  and comments  helps build a vibrant community of like-minded individuals .



Follow



Written by Deepa Mathan

86 Followers

I am ambitious in learning and adapting new skills, passionate in Cloud & DevOps, always wanted to stay up to date with the latest industry trends .

More from Deepa Mathan



TERRAFORM A BEGINNER GUIDE



Community



[deepamathan](#)

Deepa Mathan

Terraform—Infrastructure as Code Tool

A Beginner's Guide to Infrastructure Automation with Terraform

15 min read · Oct 23

20 2



>>>>

ANSIBLE AUTOMATION

GUIDE FOR BASIC CONCEPTS



DEVOPS CONSULTANT



Deepa Mathan

Ansible Basics and its Playbook Configuration

This blog completely talks about the basic concepts of Ansible and playbook configurations with hands-on .

8 min read · Sep 6

47 1

+

>>>>

APACHE MAVEN

POWER OF BUILD AUTOMATION



DEVOPS CONSULTANT



Deepa Mathan

Apache Maven—Build Tools in DevOps

In this blog let's explore on one of the best and widely used build tool in devops .

5 min read · Aug 29



Deepa Mathan

AWS Account Creation Made Easy: A Detailed Walkthrough

Amazon Web Services (AWS) is Amazon.com's widely used and extensive cloud computing platform. It offers a wide range of cloud services to...

5 min read · Oct 17

See all from Deepa Mathan

Recommended from Medium



Sushant Kapare

🚀 Explaining DevOps Projects in Interviews 🔧

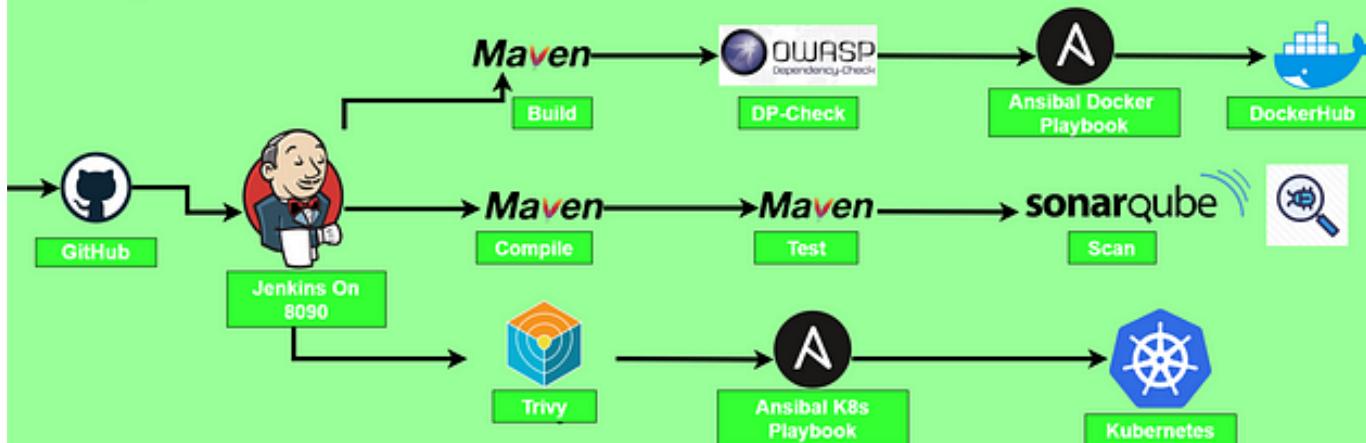
Introduction:-

3 min read · 4 days ago

37 1

+

/Secops



Dhananjay Tupe

Ci-Cd With Ansible—DevSecOps Petshop project

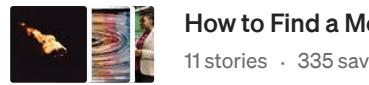
You can find GitHub Repository from here :

13 min read · Dec 10

Lists

**Self-Improvement 101**

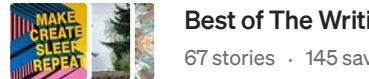
20 stories · 1064 saves

**How to Find a Mentor**

11 stories · 335 saves

**Good Product Thinking**

11 stories · 386 saves

**Best of The Writing Cooperative**

67 stories · 145 saves

```
outputs.tf
variables.tf
IAM
main.tf
outputs.tf
variables.tf
KEY
main.tf
variables.tf
NAT-GW
main.tf
variables.tf
NodeGroup
main.tf
variables.tf
vpc
main.tf
outputs.tf
variables.tf
README.md
```

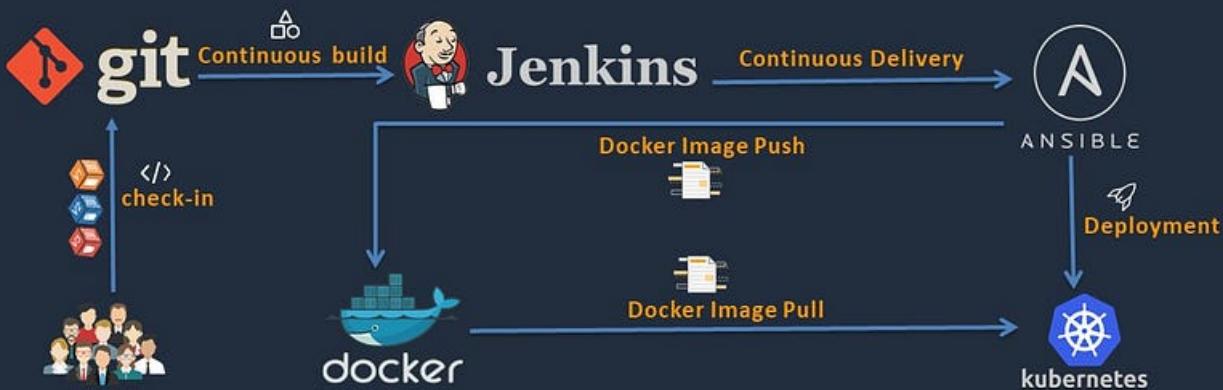
 Ernest Buadu Kwashie

Provisioning Amazon Elastic Kubernetes Service (EKS) Cluster using Terraform

Hello Everyone, I am excited to share with you my latest article in which I dockerized an application, deployed it to Amazon Elastic...

4 min read · Dec 6

Simple DevOps Project



Dibaal Akari

Building a CI/CD Pipeline for a Retail Company

Edureka Industry Grade Project...

17 min read · Sep 1

83 2

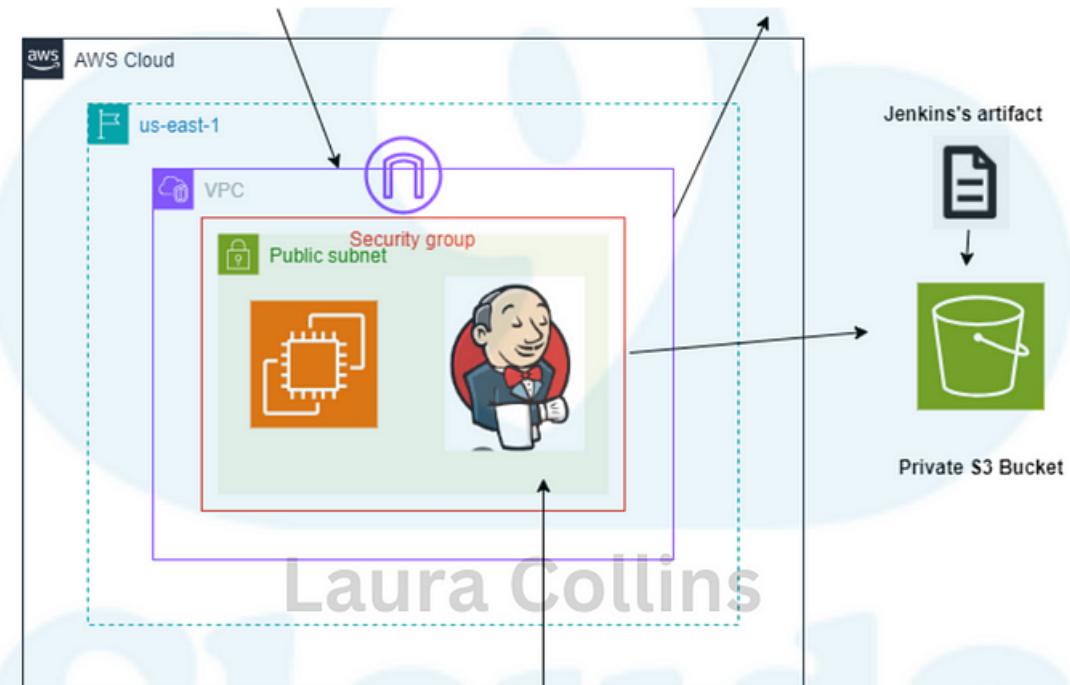


Samuel Colon in Towards AWS

DevOps (Lab-5)-Detecting code vulnerabilities with OWASP

About OWASP:

4 min read · Nov 27



 Laura Collins in DevOps.dev

Deploying EC2 Instance and S3 Bucket with Jenkins using Terraform

Scenario: Your team would like to use Jenkins as their CI/CD tool to create pipelines for DevOps projects. They need you to create the...

13 min read · Dec 7

See more recommendations