

# **CI/CD Pipeline by using Jenkins for Maven Build, Test, Integration Test, Sonarqube (Quality Gates), Nexus (Artifacts Repo Manager), Software Testing and Slack for Notifications**

(Hands on Practice from Udemy)

Date: Feb 28, 2022

## **Scenario**

Consider there is a product development is happening, with Agile SDLC. In this developers building a products and making a regular code changes. All these code changes which gets merged with remote repository, to be built, tested and deploy to server for further. Once the code is packaged into an artifacts, it will be deployed to server. Software testing/ integration testing after deployment. Test report gets evaluated and approval for prod deploy issued.

## **Problems**

- In an Agile SDLC, there will be frequent code changes
- Manual deployment is time consuming
- Involves task assignment/ ticketing/ approvals
- Dependency on Ops, & Build & Release Team

## **Solution**

- Build, Test, Deploy & Test for every commit
- Automated Process
- Notify for every build
- Fix code if bugs, or error found instantly rather than waiting

## **Tools to be used**

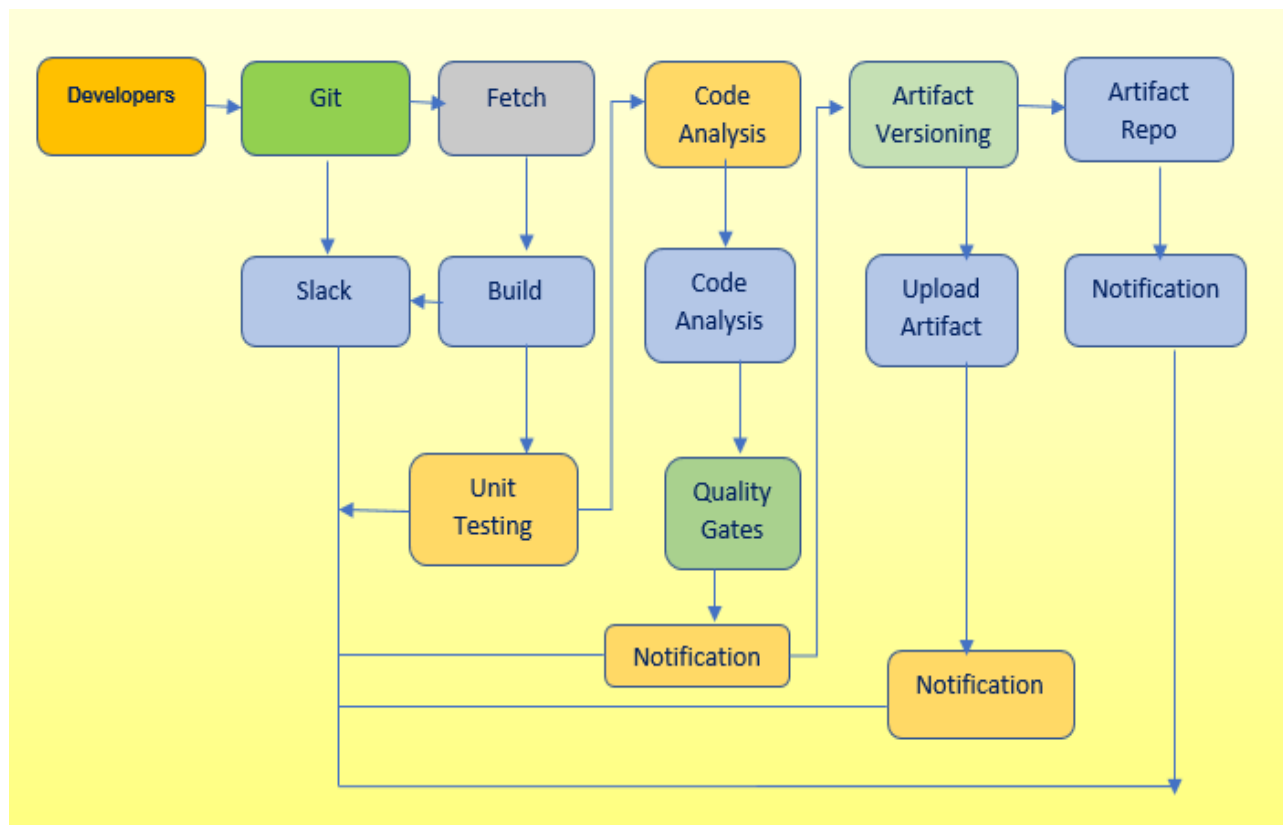
- Jenkins as Continuous Integration Server
- Maven as build tool
- Git as version control
- Check style, Slack
- Nexus as Artifact Repository
- SonarQube as Code Analysis
- Tomcat as Web application Server
- Selenium for Software testing
- Windows server and AWS EC2 for all servers

## **Steps to be followed:**

- Login to AWS account.
- Create login key
- Create Security group for Jenkins, Nexux, Sonar
- Create EC2 instances with user data for Jenkins, Nexus and Sonar Qube

- Post Jenkins installation, access from web page and complete initial login setup
- Create repository in Nexus post Nexus installation including Maven repo
- Sonarqube post installation, and create a token which is mapped with Jenkins job
- Jenkins Steps: Build Job, Setup Slack Notification, Check style code analysis job, Set up Sonar integration, Sonar Code Analysis, Artifact upload job,
- Connect all jobs with build pipeline.
- Set automatic build trigger for each pipeline jobs.
- Test with git
- Create Security group for windows server, Tomcat and backend server
- Setup tomcat and backend server on EC2 with user data
- Add windows node as slave to Jenkins
- Create job to run software tests, from windows server
- Deploy artifact to production tomcat server
- Connect all the jobs with build pipeline
- Test it by committing the code to Git Hub

## Workflow of Continuous Integration



## Screenshots of EC2 services of Jenkins, Nexus, and Sonarqube servers

The screenshot displays the AWS Management Console interface. The top navigation bar shows the 'Instances' page. The left sidebar contains navigation options like 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity Reservations', and 'Images'. The main content area shows a list of EC2 instances. Below the list, the details for the 'Nexus server' instance (ID: i-0a5094e7105e5d697) are expanded, showing its state as 'Running' and various network details.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Nexus server	i-0a5094e7105e5d697	Running	t2.medium	2/2 checks passed	No alarms	ap-south-1a
Sonarqube ser...	i-037df67a1115f26fa	Running	t2.medium	2/2 checks passed	No alarms	ap-south-1a
Jenkins-server-ci	i-0ba3eff2240e4facc	Running	t2.small	2/2 checks passed	No alarms	ap-south-1b

**Instance: i-0a5094e7105e5d697 (Nexus server)**

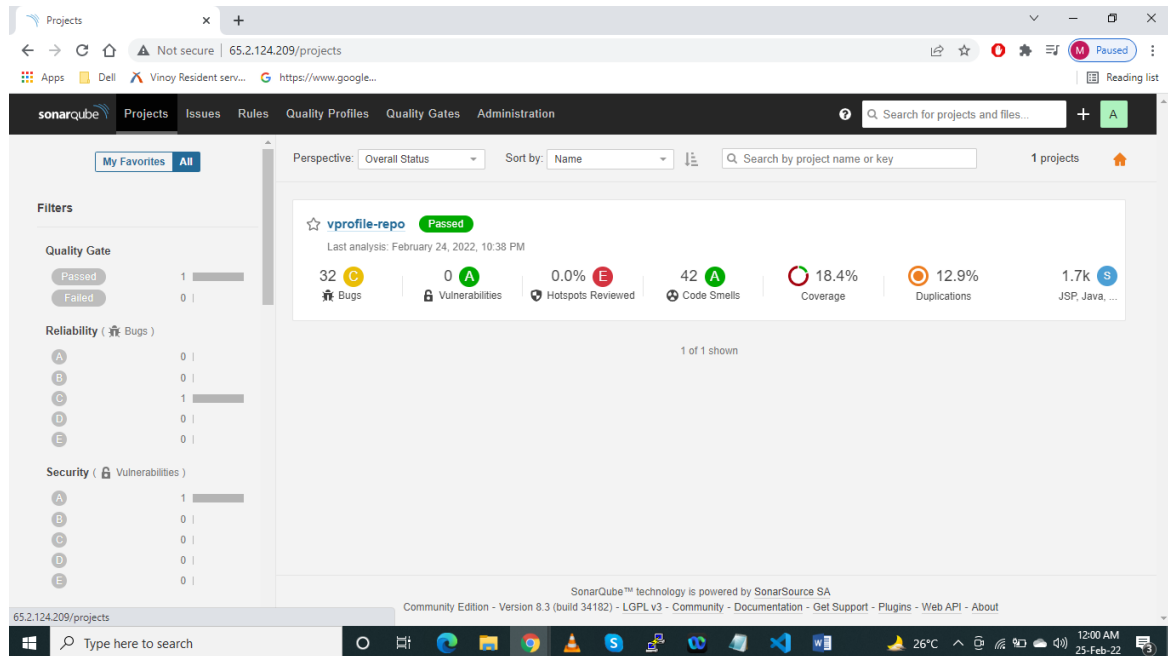
Instance summary	
Instance ID	Public IPv4 address
i-0a5094e7105e5d697 (Nexus server)	3.108.228.164   <a href="#">open address</a>
Private IPv4 addresses	172.31.45.118
IPv6 address	Instance state
-	Running
	Public IPv4 DNS
	ec2-3-108-228-164.ap-south-

## Screenshot of Nexus Repository Manager

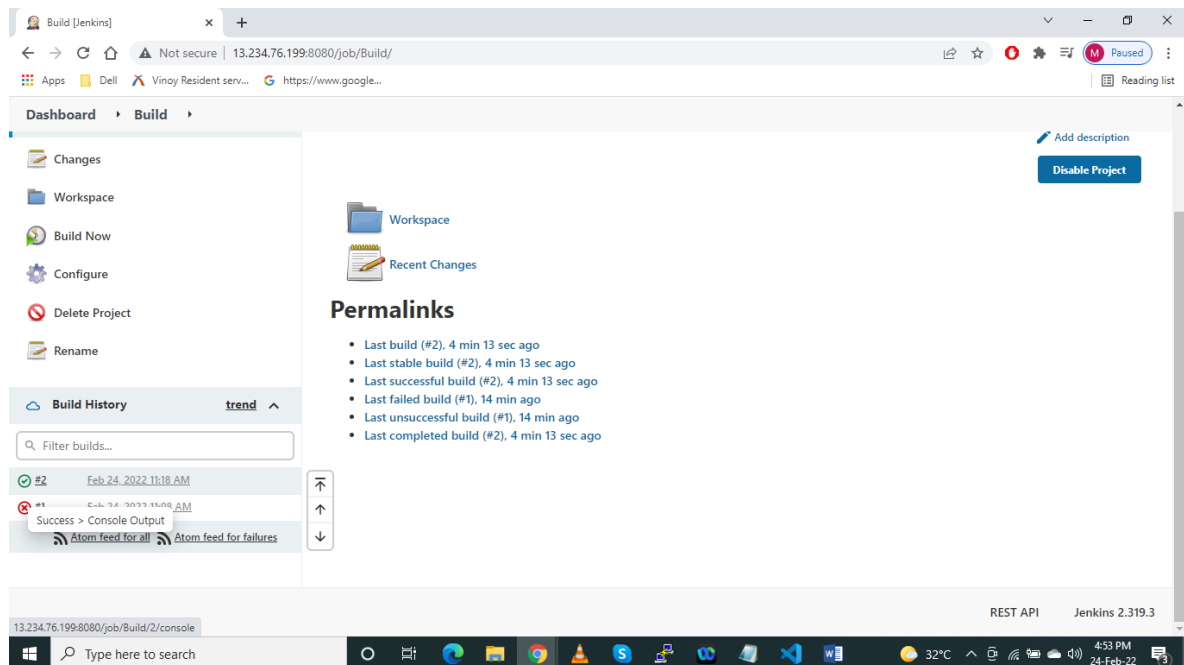
The screenshot shows the Sonatype Nexus Repository Manager web interface. The top navigation bar includes the Sonatype logo, version 'OSS 3.37.3-02', a search bar, and user information 'admin'. The left sidebar contains navigation options like 'Administration', 'Repository', 'Repositories', 'Blob Stores', 'Cleanup Policies', 'Content Selectors', 'Proprietary Repositories', 'Routing Rules', 'Security', 'Privileges', 'Roles', 'Users', and 'Anonymous Access'. The main content area displays a table of repositories.

Name	Type	Format	Status	URL	Health check	IQ Policy Vi...
maven-central	proxy	maven2	Online - Ready to Connect	<a href="#">copy</a>	<a href="#">Analyze</a>	<a href="#">...</a>
maven-public	group	maven2	Online	<a href="#">copy</a>	<a href="#">...</a>	<a href="#">...</a>
maven-releases	hosted	maven2	Online	<a href="#">copy</a>	<a href="#">...</a>	<a href="#">...</a>
maven-snapshots	hosted	maven2	Online	<a href="#">copy</a>	<a href="#">...</a>	<a href="#">...</a>
nuget-group	group	nuget	Online	<a href="#">copy</a>	<a href="#">...</a>	<a href="#">...</a>
nuget-hosted	hosted	nuget	Online	<a href="#">copy</a>	<a href="#">...</a>	<a href="#">...</a>
nuget.org-proxy	proxy	nuget	Online - Ready to Connect	<a href="#">copy</a>	<a href="#">Analyze</a>	<a href="#">...</a>
vprofile-maven-central	proxy	maven2	Online - Remote Available	<a href="#">copy</a>	<a href="#">Analyze</a>	<a href="#">...</a>
vprofile-maven-group	group	maven2	Online	<a href="#">copy</a>	<a href="#">...</a>	<a href="#">...</a>
vprofile-maven-snapshot	hosted	maven2	Online	<a href="#">copy</a>	<a href="#">...</a>	<a href="#">...</a>
vprofile-release	hosted	maven2	Online	<a href="#">copy</a>	<a href="#">...</a>	<a href="#">...</a>

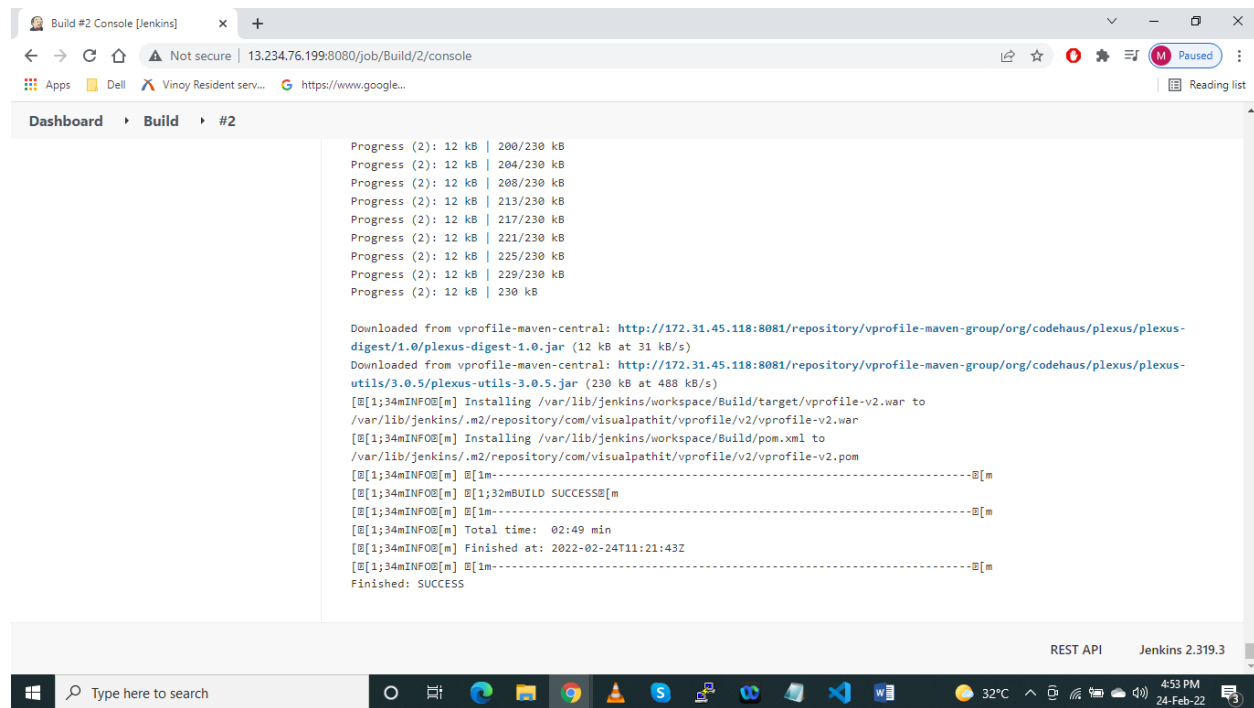
## Screenshot of Sonarqube Quality gates



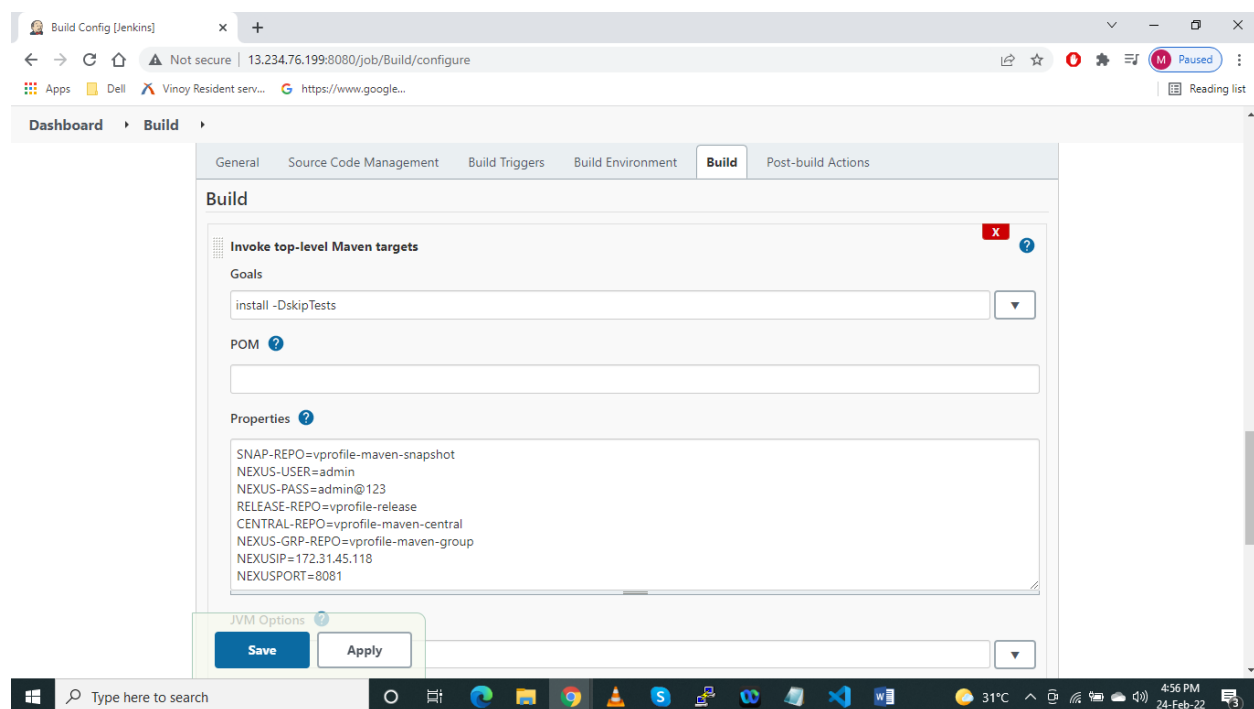
## Post installation of Jenkins, ready to configure to build a pipeline



## Console output of Maven build when build comes success



## Giving application properties with variables in build job configuration.



*Screenshot of pipeline jobs created such as Build, Test, Integration Test, Code Analysis, Sonar Scanner Analysis, and Deploy to Nexus*

Screenshot of the Jenkins web interface showing a Build Pipeline. The pipeline consists of the following steps:

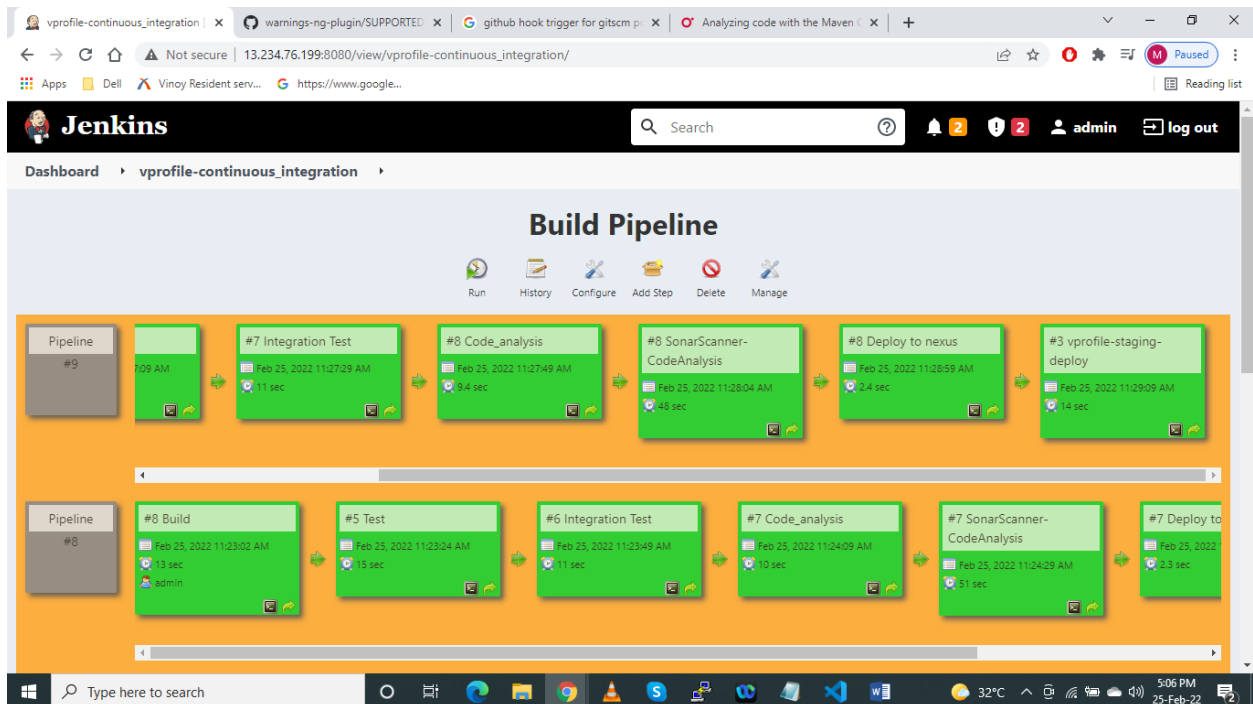
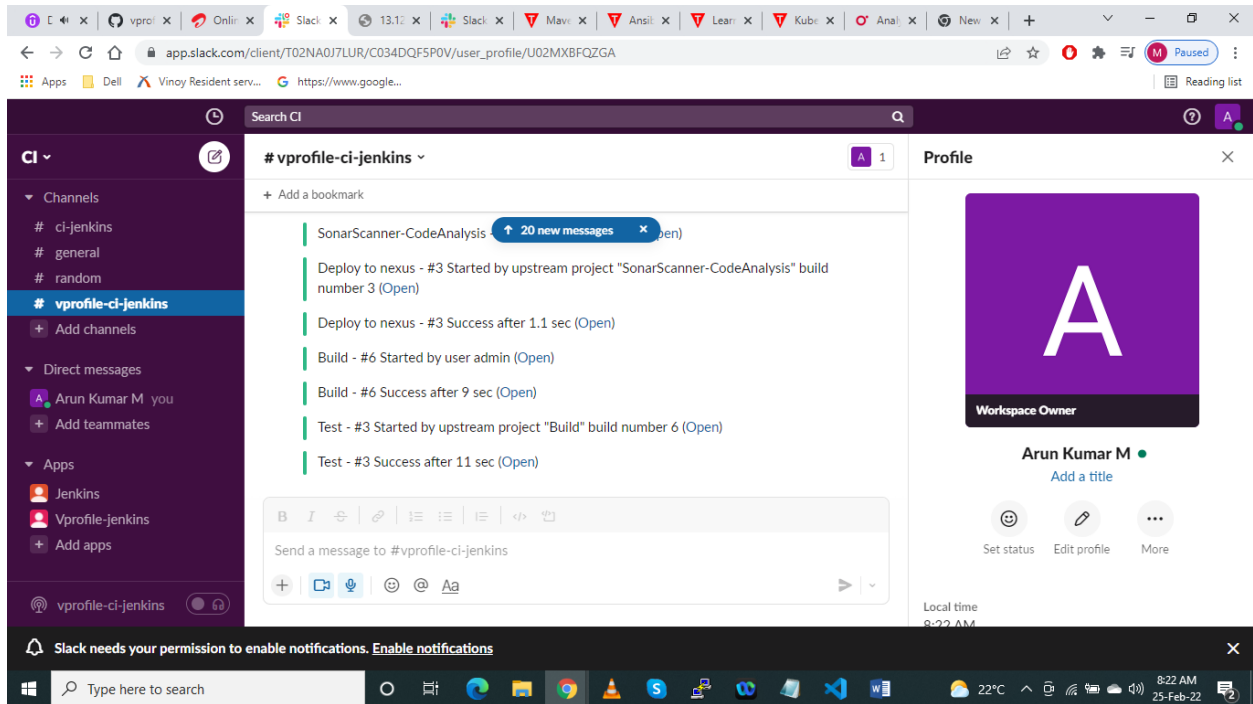
- Pipeline #5
- #5 Build (Feb 25, 2022 2:47:54 AM, 8.3 sec)
- #2 Test (Feb 25, 2022 2:48:13 AM, 11 sec)
- #2 Integration Test (Feb 25, 2022 2:48:33 AM, 8.8 sec)
- #3 Code\_analysis (Feb 25, 2022 2:48:46 AM, 4.6 sec and counting)
- SonarScanner-CodeAnalysis (N/A)
- Deploy to nexus (N/A)

The screenshot also shows the Windows taskbar and the Jenkins REST API endpoint (Jenkins 2.319.3).

REST API Jenkins 2.319.3

REST API Jenkins 2.319.3

Screenshot of Slack notifications while committing each code changes, which triggers the automated build and rest of pipeline jobs



## Placing WAR file on tomcat application, and validating

The screenshot shows the Tomcat Web Application Manager in a web browser. The address bar shows the URL `13.127.166.38:8080/manager/html`. The page title is "Tomcat Web Application Manager". Below the title, there is a "Message" box with "OK". The main content area is divided into sections: "Manager" (with links for List Applications, HTML Manager Help, Manager Help, and Server Status), "Applications" (a table of running applications), and "Deploy" (a section for deploying directories or WAR files).

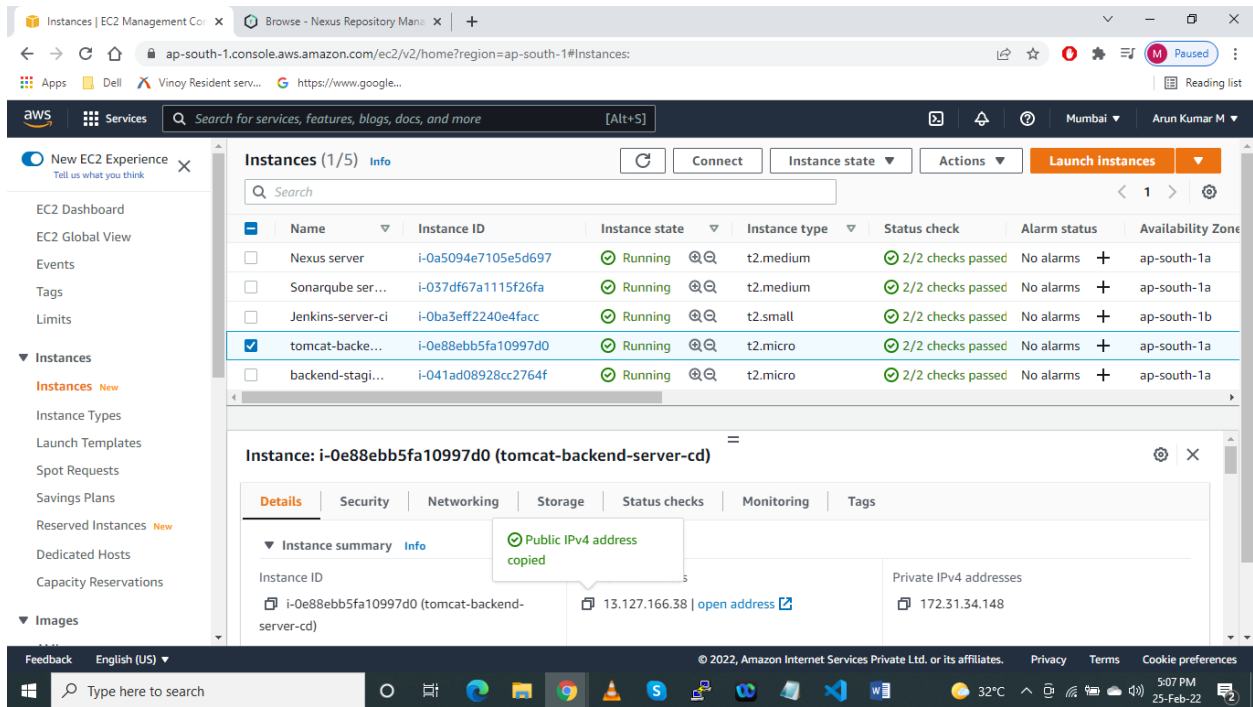
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet 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
/vprofile-v2	None specified	Account Registration Web Application	false	0	Start Stop Reload Undeploy

## Screenshot of web application home page, which is running from the tomcat server installed from continuous delivery.

The screenshot shows the login page of the Visual Path web application. The page has a dark header with "VISUAL PATH" and navigation links for "TECHNOLOGIES", "ABOUT", "BLOG", "LOGIN", and "SIGN UP". The main content area features a "LOGIN" form with fields for "Username" and "Password", a "LOGIN" button, and a link to "Create an account".

The screenshot shows the user profile page for "admin\_vp" (admin\_vp@visualpath.co.in). The page includes a profile picture, a bio, and a post titled "The Key to DevOps Success". The bio states: "DevOps For Product Management and Strategy of Application Delivery at VisualPath Technologies. Responsible of providing customers with counsel on their DevOps strategies to help them deliver higher quality software and services to market faster." The post content reads: "The Key to DevOps Success" Collaboration. Collaboration is essential to DevOps, yet how to do it is often unclear with many teams falling back on ineffective conference calls, instant messaging, documents, and SharePoint sites. In this keynote we will share a vision for a next generation DevOps where collaboration, continuous documentation, and knowledge capture are combined with automation toolchains to enable rapid innovation and deployment.





Screenshot of software testing job performed in windows server, which will take screen shot of login page function test.

