INSTITUTE FOR ADVANCED COMPUTING

AND
SOFTWARE DEVELOPMENT

AKURDI, PUNE

Secure building Web Application with CI/CD pipeline using Git, Jenkins and Docker hub

GROUP NO.6

GAURAV R. JADHAV (233412)

KAUSTUBH VEER (233413)

MR. KARTIK AWARI	MR. ROHIT PURANIK
PROJECT GUIDE	CENTRE CO-ORDINATOR

TABLE OF CONTENTS

1.INTRODUCTION

This is introducing a comprehensive DevSecOps project aimed at enhancing security throughout the software development lifecycle. The project's primary objectives are to automate security checks, reduce vulnerabilities, and foster a proactive security culture within the organization. Key elements of the project include the selection and integration of security tools, the implementation of automated security testing, and the creation of security-as-code policies.

The project also addresses the importance of security training and awareness for all stakeholders, ensuring that developers, operations teams, and security experts work collaboratively to identify and mitigate risks. Continuous monitoring and incident response plans are established to promptly detect and respond to security incidents, minimizing potential damage.

Furthermore, the project examines compliance with industry-specific regulations and standards, such as GDPR, HIPAA, and PCI DSS, to guarantee that security practices align with legal requirements. Emphasis is also placed on securing containerized applications and cloud environments, reflecting the modern tech landscape's dynamic nature.

This abstract provides a high-level overview of the DevSecOps project, highlighting its significance in today's cybersecurity landscape. It promotes the integration of security as a fundamental component of the software development process, ultimately enhancing an organization's resilience against cyber threats and ensuring the delivery of secure, reliable, and compliant software products.

2.Tools
2.1 GIT

A) Introduction to Git

Git is a DevOps tool used for source code management. It is a free and open-source version control system used to handle small to very large projects efficiently. Git is used to tracking changes in the source code, enabling multiple developers to work together on non-linear development. Linus Torvalds created Git in 2005 for the development of the Linux kernel.

It is used for:

•Coding collaboration
•Git is used to tracking changes in the source code
•The distributed version control tool is used for source code management
•It allows multiple developers to work together
•It supports non-linear development through its thousands of parallel branches

Features of Git

•Tracks history
•Free and open source
•Supports non-linear development
•Creates backups
•Scalable
•Supports collaboration
•Branching is easier

Git Workflow

The Git workflow is divided into three states:
•Working directory - Modify files in your working directory
•Staging area (Index) - Stage the files and add snapshots of them to your staging area
•Git directory (Repository) - Perform a commit that stores the snapshots permanently to your Git directory. Checkout any existing version, make changes, stage them and commit.

What does Git do?

•Manage projects with Repositories
•Clone a project to work on a local copy
•Control and track changes with Staging and Committing
•Branch and Merge to allow for work on different parts and versions of a project
•Pull the latest version of the project to a local copy
•Push local updates to the main project

B) Git installation on linux

Git is a popular open-source version control system like CVS or SVN. This article is for those, who are not familiar with Git. Here, we are providing you with basic steps of installing Git from source, creating a new project, and Commit changes to the Git repository.

Use the following comman d to install git on Linux –
$ sudo apt-get install git

2.2.GITLEAKS

Gitleaks is a security scanning tool that helps you find potential security vulnerabilities in your git repositories, files, and directories. It does this by scanning your code and looking for sensitive information that may have been accidentally committed.
This information can include passwords, APIs, private keys, and other sensitive data.

Gitleaks has two main commands:

1)Protect

The protect command creates a hook to prevent commits that introduce potential security vulnerabilities.
This command will create a file called. gitleaks.hook in the repository's root directory.

2)Detect

To use the detect command, navigate to the repository's root directory and run the following command:
gitleaks --repo-url=https://github.com/gitleakstest/sample --report=report.json

Why we use gitleaks

Gitleaks is a SAST tool that scans your latest source code and your entire git history to identify any secrets that may have been committed in the past. It is a valuable tool for preventing data breaches, as it can help to uncover hardcoded secrets that attackers could exploit.

2.3 JENKINS

A) Introduction of Jenkins

Jenkins is an open-source automation tool written in Java programming language that allows continuous integration. Jenkins builds and tests our software projects which continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build. The Jenkins project was started in 2004 (originally called Hudson) by Kohsuke Kawaguchi, while he worked for Sun Microsystems. Jenkins is an open-source solution comprising an automation server to enable continuous integration and continuous delivery (CI/CD), automating the various stages of software development such as build, test, and deployment.
In 2011, the Hudson community unanimously accepted a referendum to alter the project name from Hudson to Jenkins, resulting in the creation of the first "Jenkins" project. Hudson was later donated to the Eclipse Foundation and is no longer being worked on. Jenkins development is currently administered as an open-source project under the direction of the CD Foundation, a Linux Foundation initiative.

2)Use of Jenkins

Jenkins is still used in modern software development lifecycles.it enable and automates various processes (such as testing,building,development) as a plateform,it create CI pipelines that define a series of action of server will take for requisite task.

Use CI/CD pipeline: Jenkins offers a simple way to set up a continuous integration or continuous delivery (CI/CD) environment for almost any combination of languages and source code repositories using pipelines, as well as automating other routine development tasks.

2.4 SONARQUBE

What is sonarqube: It is a code quality assurance tool that collect and analyses, source code and provide report for the code quality of your projcts.

How sonarqube working:
1.One SonarQube Server starting 3 main processes:
1.Web Server for developers, managers to browse quality snapshots and configure the SonarQube instance
2.Search Server based on Elasticsearch to back searches from the UI
3.Compute Engine Server in charge of processing code analysis reports and saving them in the SonarQube Database
2.One SonarQube Database to store:
1. configuration of the SonarQube instance (security, plugins settings, etc.)
2.The quality snapshots of projects, views, etc.
3.Multiple SonarQube Plugins installed on the server, possibly including language, SCM, integration, authentication, and governance plugins
4.One or more SonarScanners running on your Build / Continuous Integration Servers to analyze projects.

Top Alternatives to SonarQube
•Embold.
•GitHub.
•Coverity.
•Checkmarx.
•Klocwork.
•GitLab.

Benefits of SonarQube
•Sustainability – Reduces complexity, possible vulnerabilities, and code duplications, optimising the life of applications.
•Increase productivity – Reduces the scale, cost of maintenance, and risk of the application; as such, it removes the need to spend more time changing the code.

Sonarqube Installation Process on AWS EC2
1)Launch EC2 instance first,log into your AWS management console and navigate to EC2 dashboard
2) Install Sonarqube
3) Configure Sonarqube
4)Start Sonarqube

2.5 DOCKERHUB

Dockerhub: Docker Hub is a collaboration tool and a marketplace for commuDocker Hub is a collaboration tool and a marketplace for community developers, open source contributors, and independent software vendors (ISVs) to distribute their code publicly.nity developers, open source contributors, and independent software vendors (ISVs) to distribute their code publicly.
Docker Hub is the world's largest library and community for container images.

Containerization
Containerization is a software deployment process that bundles an application's code with all the files and libraries it needs to run on any infrastructure. Traditionally, to run any application on your computer, you had to install the version that matched your machine's operating system. For example, you needed to install the Windows version of a software package on a Windows machine. However, with containerization, you can create a single software package, or container, that runs on all types of devices.

2.6 DOCKER SWARM

Master Slave Concept:
In the master/slave (sometimes called boss/worker) model, a master entity receives one or more requests, then creates slave entities to execute them. Typically, the master controls the number of slaves and what each slave does. A slave runs independently of other slaves.
Example:-
An example of this model is a print job spooler controlling a set of printers. The spooler's role is to ensure that the print requests received are handled in a timely fashion. When the spooler receives a request, the master entity chooses a printer and causes a slave to print the job on the printer.

Difference Between Master And Slave
Master/slave is the historical terminology for a model of asymmetric communication or control where one device or process (the "master") controls one or more other devices or processes (the "slaves") and serves as their communicate on hub.

Master slave Installation steps on Docker

1.Label: Name which will be later used by Jenkins Jobs.
2.Docker image: Docker image name which image needs to be used by Jenkins Slave.
3.Remote File System Root: The home folder for the user we've created in the image. ...
4.Connect Method: SSH.
5.User: Name of the user used to connect, Jenkins.

2.7 LOAD BALANCER

A load balancer is a physical device that acts as a reverse proxy and distributes network or application traffic across a number of servers. Load balancers are used to increase capacity (concurrent users) and reliability of applications.
A load balancer may be: A physical device, a virtualized instance running on specialized hardware, or a software process. Application delivery controllers, which are incorporated into ADCs, are designed to improve the performance and security of applications, no matter where they are hosted.
What is a load balancer and how its works:
Load balancers increase the fault tolerance of your systems by automatically detecting server problems and redirecting client traffic to available servers. You can use load balancing to make these tasks easier: Run application server maintenance or upgrades without application downtime.

TYPES OF LOAD BALANCING

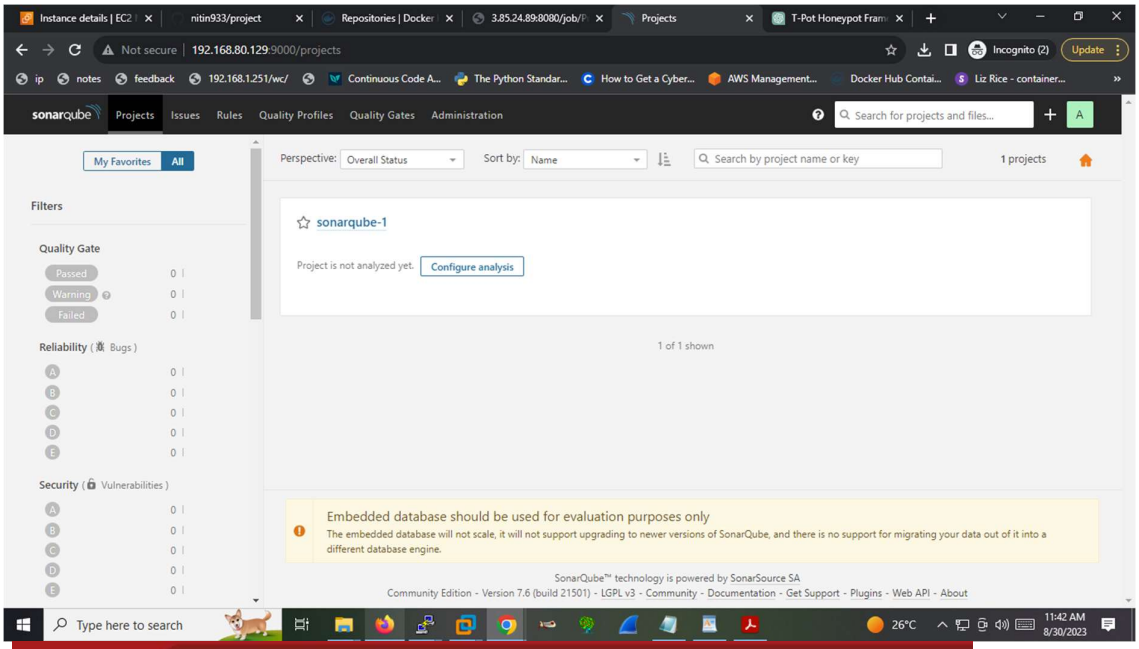
•Application Load Balancer.
•Network Load Balancer.
•Application Load Balancer and Network Load Balancer consideration.

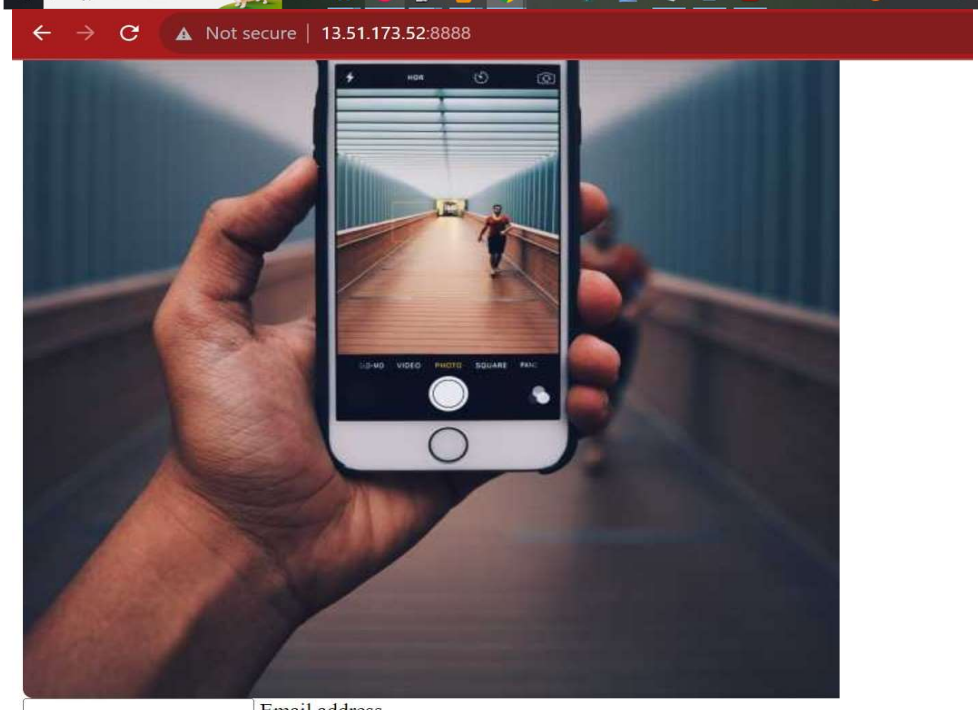
LOAD BALANCING

NGINX As Load Balancer Configuration

To setup Nginx as a load balancer for backend servers, follow these steps:
1.Open the Nginx configuration file with elevated rights
2.Define an upstream element and list each node in your backend cluster
3.Map a URI to the upstream cluster with a proxy_pass location setting
4.Restart the Nginx server to incorporate the config changes
5.Verify successful configuration of the Nginx load balancer setup

SonarQube:


Instance details | EC2
nitin933/project
Repositories | Docker
3.85.24.89:8080/job/P
Projects
T-Pot Honeypot Fram
Not secure | 192.168.80.129:9000/projects
Incognito (2)
Update
ip
notes
feedback
192.168.1.251/wc/
Continuous Code A...
The Python Standar...
How to Get a Cyber...
AWS Management...
Docker Hub Contai...
Liz Rice - container...
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
Search for projects and files...
My Favorites
All
Perspective: Overall Status
Sort by: Name
Search by project name or key
1 projects
Filters
Quality Gate
Passed
Warning
Failed
Reliability (Bugs)
Security (Vulnerabilities)
sonarqube-1
Project is not analyzed yet.
Configure analysis
1 of 1 shown
Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.
SonarQube™ technology is powered by SonarSource SA
Community Edition - Version 7.6 (build 21501) - LGPL v3 - Community - Documentation - Get Support - Plugins - Web API - About
Type here to search
26°C
11:42 AM
8/30/2023


Not secure | 13.51.173.52:8888
Email address
Password
Remember me
Forgot password?
Sign in

3 DATABASE CONNECTION (MariaDB)

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.
Data within the most common types of databases in operation today is typically modeled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data.
Types of databases
There are many different types of databases. The best database for a specific organization depends on how the organization intends to use the data.
1) Relational databases:
•Relational databases became dominant in the 1980s. Items in a relational database are organized as a set of tables with columns and rows. Relational database technology provides the most efficient and flexible way to access structured information.

2) Object-oriented databases:
•Information in an object-oriented database is represented in the form of objects, as in object-oriented programming.

3) Distributed databases:
•A distributed database consists of two or more files located in different sites. The database may be stored on multiple computers, located in the same physical location, or scattered over different networks.

4) Data warehouses:
•A central repository for data, a data warehouse is a type of database specifically designed for fast query and analysis.

5) NoSQL databases:
•A NoSQL, or nonrelational database, allows unstructured and semistructured data to be stored and manipulated (in contrast to a relational database, which defines how all data inserted into the database must be composed). NoSQL databases grew popular as web applications became more common and more complex.

6) Graph databases:
•A graph database stores data in terms of entities and the relationships between entities.

7) OLTP databases:
• An OLTP database is a speedy, analytic database designed for large numbers of transactions performed by multiple users.
These are only a few of the several dozen types of databases in use today. Other, less common databases are tailored to very specific scientific, financial, or other functions. In

addition to the different database types, changes in technology development approaches and dramatic advances such as the cloud and automation are propelling databases in entirely new directions. Some of the latest databases include.

8) Open-source databases:
•An open-source database system is one whose source code is open source; such databases could be SQL or NoSQL databases.

9) Cloud databases:
•A cloud database is a collection of data, either structured or unstructured, that resides on a private, public, or hybrid cloud computing platform. There are two types of cloud database models: traditional and database as a service (DBaaS). With DBaaS, administrative tasks and maintenance are performed by a service provider.

10) Multimodel database:
•Multimodel databases combine different types of database models into a single, integrated back end. This means they can accommodate various data types.

11) Document/JSON database:
•Designed for storing, retrieving, and managing document-oriented information, document databases are a modern way to store data in JSON format rather than rows and columns.

12) Self-driving databases:
•The newest and most groundbreaking type of database, self-driving database (autonomous databases) are cloud-based and use machine learning to automate database tuning, security, backups, updates, and other routine management tasks traditionally performed by database administrators.

4. IPTABLE

Iptable rules for allow and deny

1) sudo iptables -A INPUT -m conntrack –ctstate ESTABLISHED, RELATED -j ACCEPT

2) sudo iptables -A OUTPUT -m conntrack --ctstate ESTABLISHED -j ACCEPT

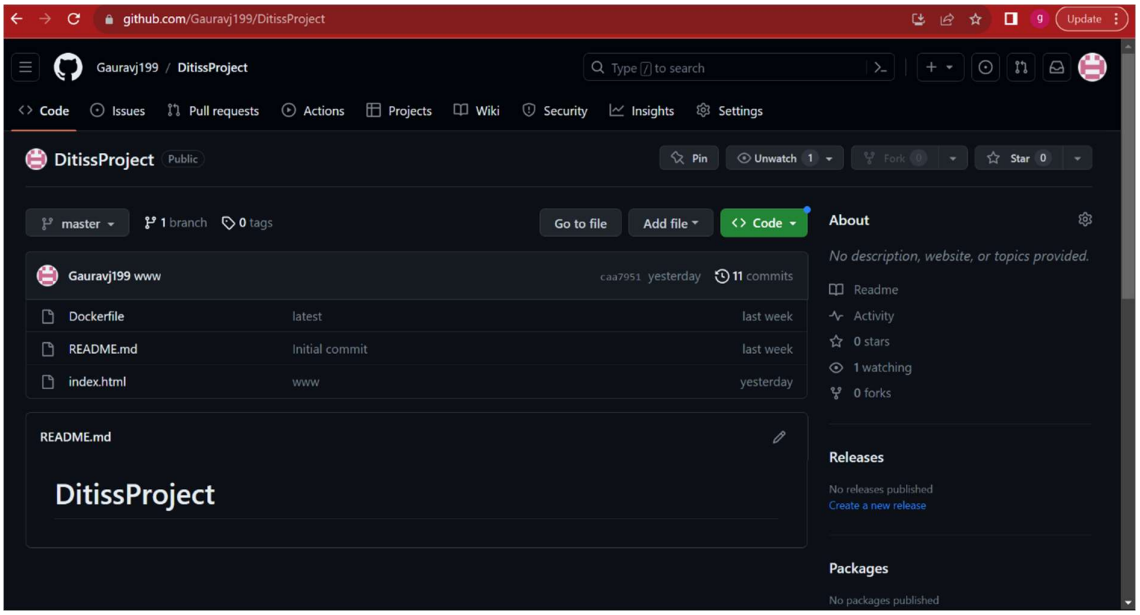
3) sudo iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT

4) sudo iptables -A INPUT -m conntrack --ctstate INVALID -j DROP

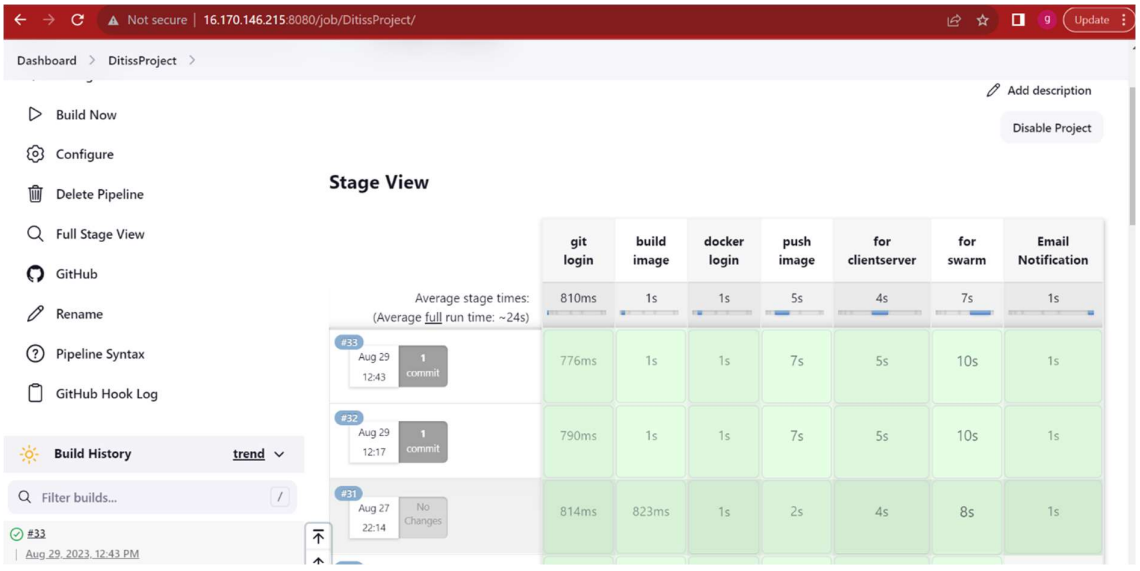
5) sudo iptables -A INPUT -s 203.0.113.51 -j DROP

6) sudo iptables -A INPUT -s 203.0.113.51 -j REJECT

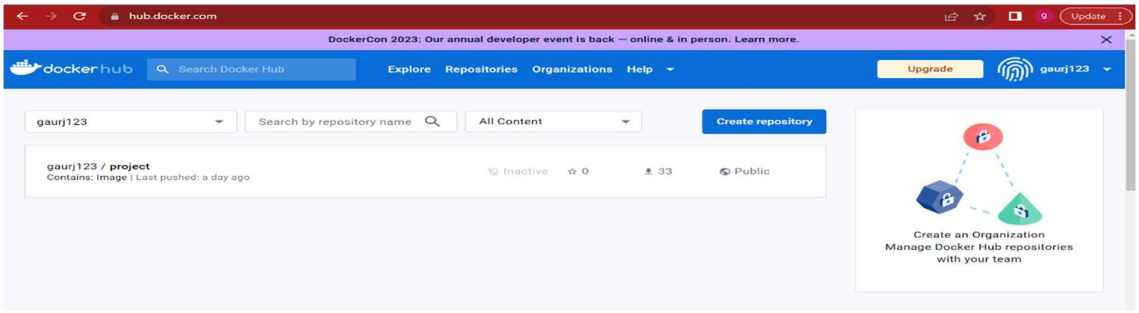
Git


github.com/Gauravj199/DitissProject
Update
Gauravj199 / DitissProject
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
DitissProject
Public
Pin
Unwatch 1
Fork 0
Star 0
master
1 branch
0 tags
Go to file
Add file
Code
About
No description, website, or topics provided.
Gauravj199 www
caa7951 yesterday
11 commits
Dockerfile latest last week
README.md Initial commit last week
index.html www yesterday
Readme
Activity
0 stars
1 watching
0 forks
README.md
DitissProject
Releases
No releases published
Create a new release
Packages
No packages published

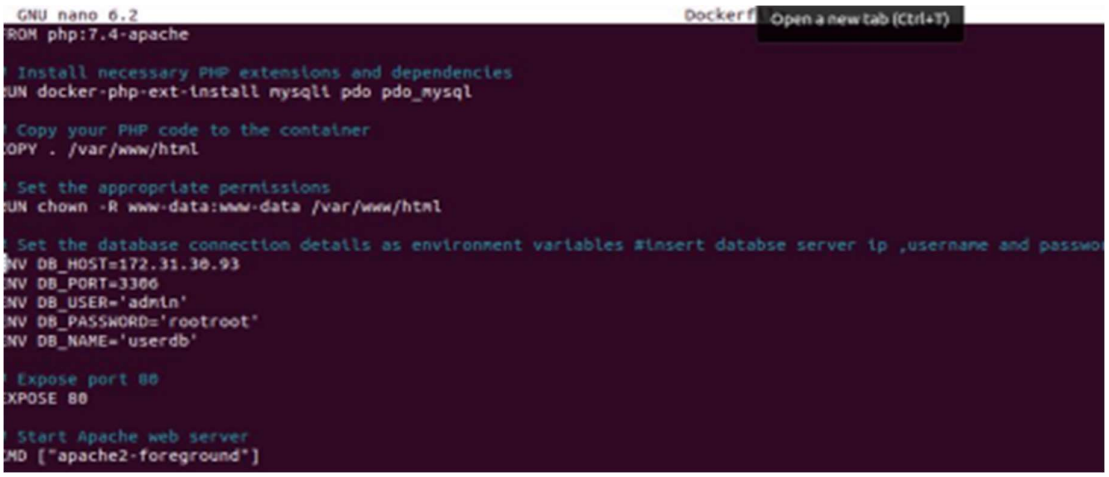
Jenkins:


Not secure | 16.170.146.215:8080/job/DitissProject/
Update
Dashboard
DitissProject
Add description
Disable Project
Build Now
Configure
Delete Pipeline
Full Stage View
GitHub
Rename
Pipeline Syntax
GitHub Hook Log
Stage View
Build History
trend
Filter builds...
#33
Aug 29, 2023, 12:43 PM
git login
build image
docker login
push image
for clientserver
for swarm
Email Notification
Average stage times:
(Average full run time: ~24s)
810ms 1s 1s 5s 4s 7s 1s
#33 Aug 29 12:43 1 commit
776ms 1s 1s 7s 5s 10s 1s
#32 Aug 29 12:17 1 commit
790ms 1s 1s 7s 5s 10s 1s
#31 Aug 27 22:14 No Changes
814ms 823ms 1s 2s 4s 8s 1s

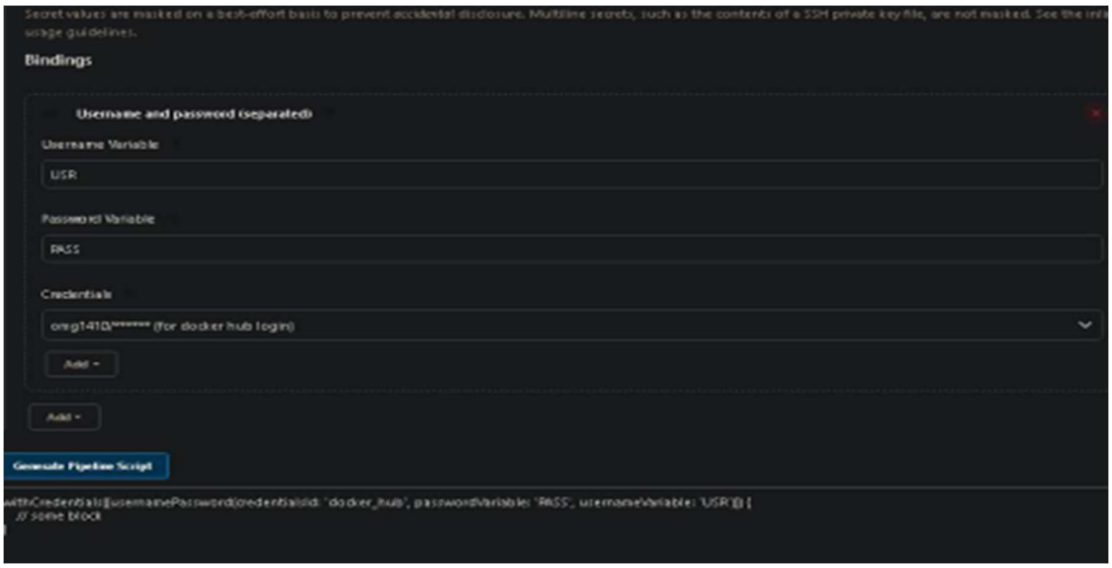
Dockerhub Image:



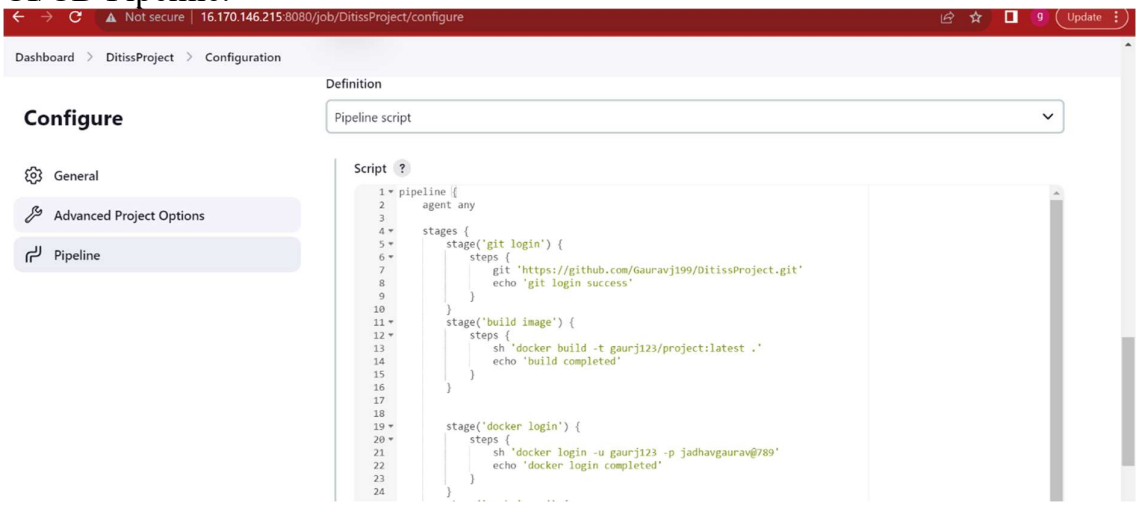
Dockerfile:



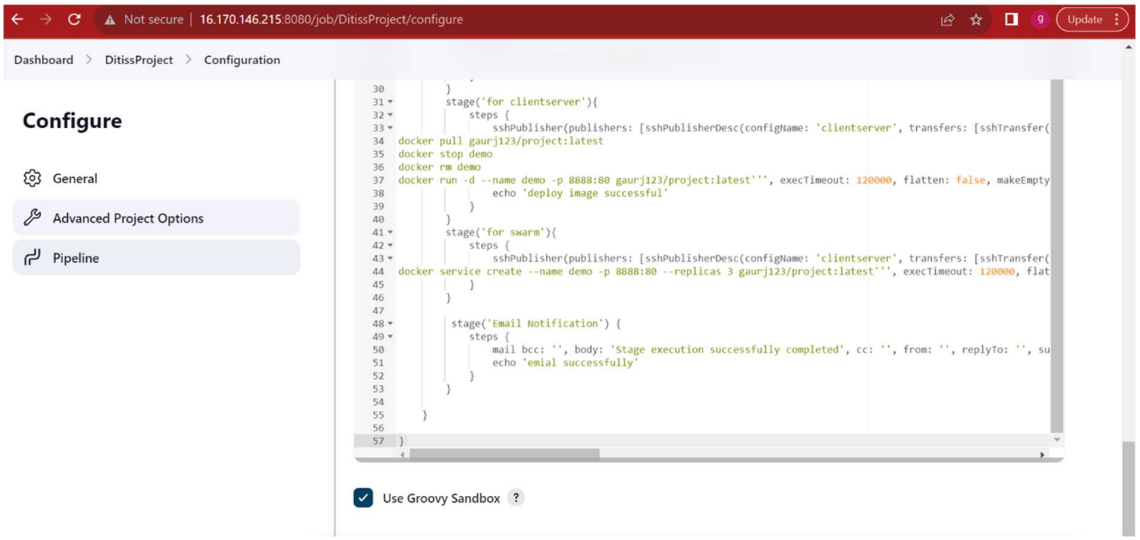
Add Dockerhub credentials:



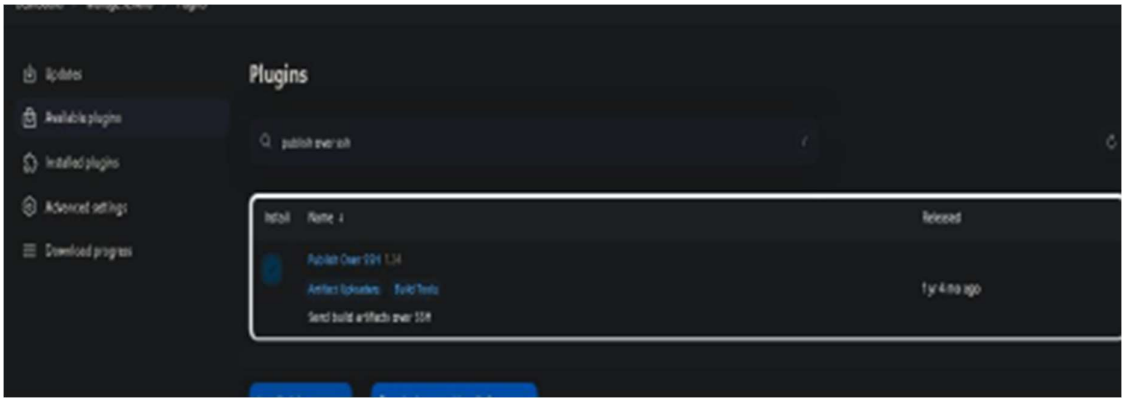
CI/CD Pipeline:



P



Publish over SSH



Manage jenkins -> system -> publish over ssh section-

Creation of load balancer-

- sudo apt-get install squid
 sudo nano /etc/squid/squid.conf

Setting up SQL server-

- sudo apt-get update
- sudo apt-get install mariadb-server
- cd /etc/mysql/mariadb.conf.d
- sudo nano 50-server.conf

Bind-address = ip of databse server I.e own ip

Grant access to nodes and Load balancer-

> CREATE DATABASE userdb;

>CREATE USER ‘admin’@’pvt ip of leader’ IDENTIFIED BY ‘password’; >CREATE USER ‘admin’@’pvt ip of node 1’ IDENTIFIED BY ‘password’; >CREATE USER ‘admin’@’pvt ip of node 2’ IDENTIFIED BY ‘password’;

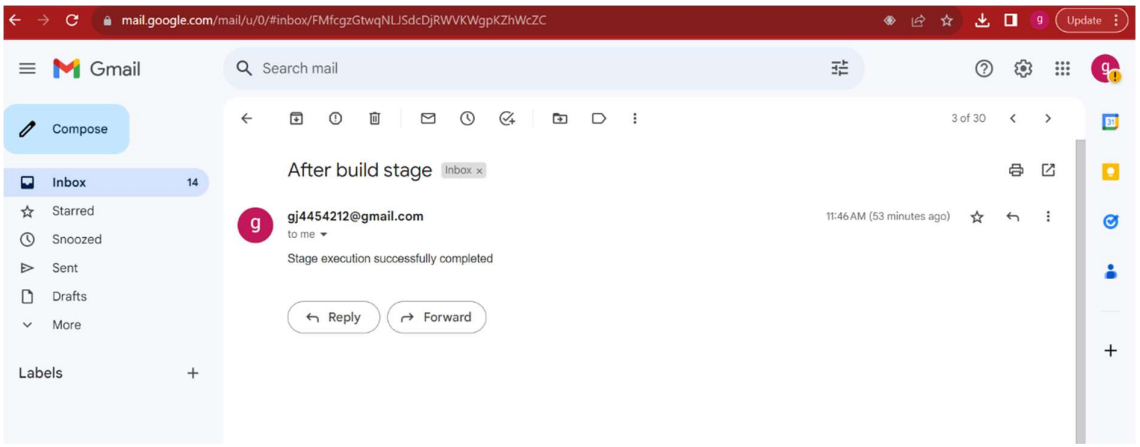
>SHOW GRANTS;
>GRANT ALL ON *.* TO ‘admin’@’pvt ip of leader’;
>GRANT ALL ON *.* TO ‘admin’@’pvt ip node 1’;
>GRANT ALL ON *.* TO ‘admin’@’pvt ip of node 2’;

>SHOW tables;
>DESC user_table;

5.Email alert configuration-

Enable 2 step verification on your google account -> manage google account -> security->passowrd -> custom -> generate app based password

Jenkins : install plugin : email alert
Manage jenkins -> syste -> extended email notification



IPtables firewall configuration-

Go to website : https://lite.ip2location.com/china-ip-address-ranges?lang=en_US Select Ip2location firwall list -> Download -> extract the file

6. ADVANTAGES & DISADVANTAGES

Advantages of Multilayered Security in a CI/CD Pipeline:

1. Comprehensive Protection: Multilayered security provides defense against a wide rangeof threats, reducing the risk of vulnerabilities slipping through the cracks.
2. Early Detection: Security measures at different stages of the pipeline catchvulnerabilities and issues early in the development process, saving time and resources.
3. Minimized Attack Surface: Multiple layers reduce the attack surface by addressingvulnerabilities across various aspects of the pipeline.
4. Adaptability: If one layer is compromised, other layers can provide backup protection, making it harder for attackers to exploit weaknesses.
5. Regulatory Compliance: Multilayered security aids in meeting regulatory requirementsby implementing various security controls and best practices.

Disadvantages of Multilayered Security in a CI/CD Pipeline:

1. Complexity: Managing multiple security layers can lead to increased complexity, potentially affecting ease of use and maintenance.
2. Resource Intensive: Implementing and maintaining multiple security measures requiresadditional resources, including time, personnel, and tools.
3. False Positives: Different security layers may generate false positives, leading totimespent investigating non-issues.
4. Configuration Challenges: Coordinating and configuring multiple security solutions canbe challenging, leading to misconfigurations or conflicts.
5. Integration Difficulties: Ensuring smooth integration between various security tools andlayers may require additional effort.

7. CONCLUSION

In conclusion, the adoption of a multilayered security approach in a CI/CD pipelineis a strategic imperative for organizations aiming to deliver secure and reliable softwareproducts efficiently. This approach acknowledges the evolving threat landscape andtheneed to fortify every stage of the development and deployment process. By implementingsecurity measures across multiple layers, from source code to runtime environments, organizations can significantly enhance their ability to detect, prevent, and respondtosecurity threats effectively. The advantages of multilayered security are evident. It offers aholistic defense strategy that minimizes the risk of vulnerabilities going undetected, providing early threat identification and reducing the attack surface. This approachsupports regulatory compliance, strengthens incident response capabilities, and fosters aculture of security awareness throughout the development lifecycle. However, it's essential to approach multilayered security with careful consideration. Balancing security measureswith operational efficiency is key to ensuring that the CI/CD pipeline remains agile andproductive. Proper configuration, seamless integration of security tools, and ongoingtraining for development teams are critical components of successful implementation.

8. REFERENCES

1) Snort Manual : http://manual-snort-org.s3-website-us-east-1.amazonaws.com/

2) Git tutorials Geeks for geeks : https://www.geeksforgeeks.org/git-tutorial/

3) Jenkins_Documentation:https://www.jenkins.io/doc/ , https://www.jenkins.io/user-handbook.pdf

4) Docker Documentation : https://docs.docker.com/

5) Linux Tutorial Geek for geek : https://www.geeksforgeeks.org/linux-tutorial/

6) SQL Manual : https://dev.mysql.com/doc/refman/8.0/en/

7) AWS tutorials : https://aws.amazon.com/getting-started/hands-on/

8) AWS Documentation: https://docs.aws.amazon.com/

9) Iptables Manual : https://linux.die.net/man/8/iptables

10) HTML Tutorial W3 School : https://www.w3schools.com/