Project Report on

Web Application Deployment on AWS with Jenkins CI/CD Pipeline, Nagios Monitoring and Snort IDS

Submitted by

Saloni Chambhare	**:**	**240344223007**
Jayesh Mahajan	**:**	**240344223013**
J.V.Tejeswar Reddy	**:**	**240344223014**
Rahul Shelke	**:**	**240344223034**

Under the guidance of

Mr. Sandeep Walvekar

In partial fulfillment of the award of Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS)

Sunbeam Institute of Information Technology,
Pune (Maharashtra)
PG-DITISS -2024

DECLARATION

We declare that this written submission represents our ideas in our own words and where others ideas or words have been included; we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed**.**

Place: Pune

Date:

Saloni Chambhare	**Jayesh Mahajan**	**J.V.Tejeswar Reddy**	**Rahul Shelke**
240344223007	**240344223013**	**240344223014**	**240344223034**

CERTIFICATE

This is to certify that the project report entitled **“Web Application Deployment on AWS with Jenkins CI/CD Pipeline, Nagios Monitoring and Snort IDS”**, submitted by **Saloni Chambhare** is the bonafide work completed under our supervision and guidance in partial fulfillment for the award of Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Mr. Sandeep Walvekar
Guide

Mr. Vishal Salunkhe
Course Coordinator

Mr. Nitin Kudale

CEO

Sunbeam Institute of Information Technology

Pune (M.S.) – 411057

CERTIFICATE

This is to certify that the project report entitled **"Web Application Deployment on AWS with Jenkins CI/CD Pipeline, Nagios Monitoring and Snort IDS"**, submitted by **Jayesh Mahajan** is the bonafide work completed under our supervision and guidance in partial fulfillment for the award of Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Mr. Sandeep Walvekar	**Mr. Vishal Salunkhe**
Guide	Course Coordinator

Mr. Nitin Kudale

CEO

Sunbeam Institute of Information Technology

Pune (M.S.) – 411057

CERTIFICATE

This is to certify that the project report entitled **“Web Application Deployment on AWS with Jenkins CI/CD Pipeline, Nagios Monitoring and Snort IDS”**, submitted by **J.V.Tejeswar Reddy** is the bonafide work completed under our supervision and guidance in partial fulfillment for the award of Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Mr. Sandeep Walvekar	**Mr. Vishal Salunkhe**
Guide	Course Coordinator

Mr. Nitin Kudale

CEO

Sunbeam Institute of Information Technology

Pune (M.S.) – 411057

CERTIFICATE

This is to certify that the project report entitled **“Web Application Deployment on AWS with Jenkins CI/CD Pipeline, Nagios Monitoring and Snort IDS”**, submitted by **Rahul Shelke** is the bonafide work completed under our supervision and guidance in partial fulfillment for the award of Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Mr. Sandeep Walvekar	**Mr. Vishal Salunkhe**
Guide	Course Coordinator

Mr. Nitin Kudale

CEO

Sunbeam Institute of Information Technology

Pune (M.S.) – 411057

APPROVAL CERTIFICATE

This Project II report entitled **“Web Application Deployment on AWS with Jenkins CI/CD Pipeline, Nagios Monitoring and Snort IDS”** by **Saloni Chambhare (240344223007)** is approved for Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Examiner: ____________________

(Signature)

____

(Name)

APPROVAL CERTIFICATE

This Project II report entitled **“Web Application Deployment on AWS with Jenkins CI/CD Pipeline, Nagios Monitoring and Snort IDS”** by **Jayesh Mahajan (240344223013)** is approved for Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Examiner: ____________________

(Signature)

____

(Name)

APPROVAL CERTIFICATE

This Project II report entitled **“Web Application Deployment on AWS with Jenkins CI/CD Pipeline, Nagios Monitoring and Snort IDS”** by **J.V.Tejeswar Reddy (240344223014)** is approved for Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Examiner: ____________________

(Signature)

____

(Name)

APPROVAL CERTIFICATE

This Project II report entitled **“Web Application Deployment on AWS with Jenkins CI/CD Pipeline, Nagios Monitoring and Snort IDS”** by **Rahul Shelke (240344223034)** is approved for Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Examiner: ____________________________

(Signature)

____

(Name)

CONTENTS

ABSTRACT

In today's fast-paced digital landscape, deploying web applications reliably, ensuring continuous integration and delivery (CI/CD), and maintaining robust security measures are paramount. This project presents a comprehensive solution by combining Amazon Web Services (AWS), Elastic Compute Cloud (EC2) for web server deployment, CI/CD pipelines for automated updates, Nagios for continuous monitoring, and an Intrusion Detection System (IDS) for application security.

The deployment process utilizes AWS EC2 instances to create a scalable and adaptable environment for hosting static web application. Through step-by-step guidelines, this project demonstrates the setup and configuration of EC2 instances, selecting appropriate instance types, and deploying static web application effectively.

The integration of CI/CD pipelines streamlines the deployment process and ensures the seamless delivery of application updates. By implementing tools like AWS Pipeline, developers can automate code testing, build processes, and deployment tasks, ultimately reducing manual errors and achieving faster release cycles.

Nagios, a powerful monitoring solution, is employed to oversee the performance and availability of the deployed web server. By configuring Nagios monitoring plugins and defining alert thresholds, administrators can promptly detect anomalies, assess performance metrics, and mitigate potential issues before they impact users.

Security is of paramount concern, and the integration of the Snort IDS adds a layer of protection against potential threats and attacks. The project guides users through the installation and configuration of Snort, showcasing its capability to monitor network traffic, detect intrusion attempts, and generate alerts for timely response.

1. INTRODUCTION

In the ever-evolving landscape of web applications, the process of deploying, managing, and safeguarding online application services has become increasingly intricate and critical. This project introduces a comprehensive approach to web application deployment, employing Amazon Web Services (AWS), Elastic Compute Cloud (EC2) for hosting, integrating continuous integration and delivery (CI/CD) pipeline for efficient updates, implementing Nagios monitoring for proactive oversight, and leveraging the Snort Intrusion Detection System (IDS) for elevated security measures.

Web Server Deployment using AWS EC2:

Amazon EC2 provides a resilient and scalable infrastructure for hosting web applications. This project delves into the fundamentals of deploying web application on AWS EC2 instances. Through a systematic walkthrough, we explore the process of provisioning EC2 instances, configuring networking settings, and optimizing the environment for seamless web application hosting.

Continuous Integration and Delivery (CI/CD):

The adoption of CI/CD practices has revolutionized the software development lifecycle, enhancing code quality and expediting deployment cycles. This project showcases the integration of CI/CD pipelines into the web application deployment process. By employing AWS Pipeline, we illustrate how developers can automate code testing, deployment, and monitoring, leading to consistent and reliable application updates.

Nagios Monitoring:

Maintaining the optimal performance and availability of web applications requires vigilant monitoring. Nagios, a renowned monitoring solution, empowers administrators to track various performance metrics and swiftly respond to anomalies. This project demonstrates the implementation of Nagios monitoring, covering the setup of monitoring plugins, configuration of alert thresholds, and real-time notification mechanisms.

Enhancing Security with Snort IDS:

As the digital landscape becomes increasingly susceptible to security threats, integrating robust security measures becomes imperative. The Snort Intrusion Detection System (IDS) serves as a proactive defense mechanism by analyzing network traffic for unauthorized activities. We delve into the deployment and configuration of Snort, showcasing its ability to detect potential intrusion attempts and generating alerts for immediate action.

Through the integration of AWS EC2, CI/CD pipeline, Nagios monitoring, and the Snort IDS, this project aims to provide a comprehensive guide to deploying, managing, and securing web applications effectively. The subsequent sections will delve deeper into each component, offering practical insights, step-by-step instructions, and best practices for orchestrating a successful web server deployment that prioritizes performance, automation, monitoring, and security in a rapidly evolving digital landscape.

1.1 Web Application

A static web application is a web app that delivers fixed content to the user's browser without any server-side processing or dynamic content generation. Unlike dynamic web applications, where content changes based on user interactions or database queries, static web applications serve the same content to every user.

The index.html file is typically the main entry point for a website or web application. It is the default file that a web server will serve when a user accesses the root of a website.

1.2 Project Plan

Table: Activities Details

Sr. No.	ACTIVITY	WEEK			
		1	2	3	4
1	Project group formation	🟨			
2	Project work to be started in respective labs	🟨			
3	First review with PPT presentation		🟨		
4	Design Use-Case view as per project			🟨	
5	Design Block diagram as per project			🟨	
6	Second review with PPT presentation				🟨
7	Selection			🟨	
8	Final review with PPT presentation			🟨	
9	Implementation coding as per project			🟨	
10	Testing, Troubleshooting with different techniques			🟨	🟨
11	Created Soft copy of project and then final hard copy				🟨

2. LITERATURE SURVEY

Paper 1: - A Qualitative Study of DevOps Usage in Practice

Author: Floris Erich, C. Amrit & M. Daneva

Description: Organizations are introducing agile and lean software development techniques in operations to increase the pace of their software development process and to improve the quality of their software. They use the term DevOps, a portmanteau of development and operations, as an umbrella term to describe their efforts. In this paper we describe the ways in which organizations implement DevOps and the outcomes they experience. We first summarize the results of a Systematic Literature Review that we performed to discover what researchers have written about DevOps. We then describe the results of an exploratory interview-based study involving six organizations of various sizes that are active in various industries. As part of our findings, we observed that all organizations were positive about their experiences and only minor problems were encountered while adopting DevOps.

Paper 2: - Devops, A New Approach To Cloud Development & Testing

Author: Dhaya Sindhu Battina

Description: The main purpose of this paper is to explore DevOps and its applications in Cloud development and testing. There's no denying it: DevOps and cloud go hand in hand. This trend will only continue since the bulk of cloud development projects now use DevOps. The advantages of utilizing DevOps with cloud applications are increasingly becoming evident. Competing well in the market necessitates a company's ability to supply services and applications at a rapid rate. To be effective, management procedures and tools need a model that is both swift and dependable. Because of this, we must automate the DevOps processes utilizing cloud and noncloud DevOps automation technologies while designing cloud-native apps. The purpose of this article is to discuss how to migrate DevOps to the cloud and improve software development and operational agility. Likewise, this project will examine ways to expand such DevOps processes and automation to public and/or private clouds. If one is interested in learning more about how the emerging field of DevOps is changing the IT industry, read this paper.

Understanding how DevOps and the Cloud work together to aid organizations in transforming themselves is the ultimate objective.

Paper 3: - Review paper on Snort and reviewing its applications in different fields

Author: Harpreet Sandhu, Manpreet Kaur.

Description: In today's era everyone wants security in data transformation but it is very difficult to protect your system and data from attackers. There are some software's and methods which gives you the surety of security like snort. Snort is a network-centric item. As an intrusion identification system, it could investigate movement inline or offline. Snort basically depends ahead a "known bad" alternately "suspected bad" approach, watching movement for examples that relate with pernicious or suspicious action. At snort detects such activity, it called (passive mode) or square (active mode). The primary may be an IDS; the second an IPS. This is a review paper which includes the information about snort, its working, installation process, components of snort, modes of snort, rules of snort and its uses.

2. SYSTEM DEVELOPMENT AND DESIGN

3.1 Proposed System

We propose a system where we are setting up one Amazon EC2 instance to host the web application, Jenkins and Nagios. Stored our application code in a version-controlled repository (Git Repository). We set up a CI/CD pipeline using Jenkins. On code changes, trigger an automated build and deployment process using Jenkins. Setting up Nagios monitoring server on the same instance. Nagios supports distributed monitoring setups, effectively accommodating growing infrastructure needs.

We Configured Snort as a Web Application Security to control inbound and outbound traffic on another EC2 instances. And also configured Snort rules to detect various types of network traffic anomalies and security threats.

3.2 Flow chart

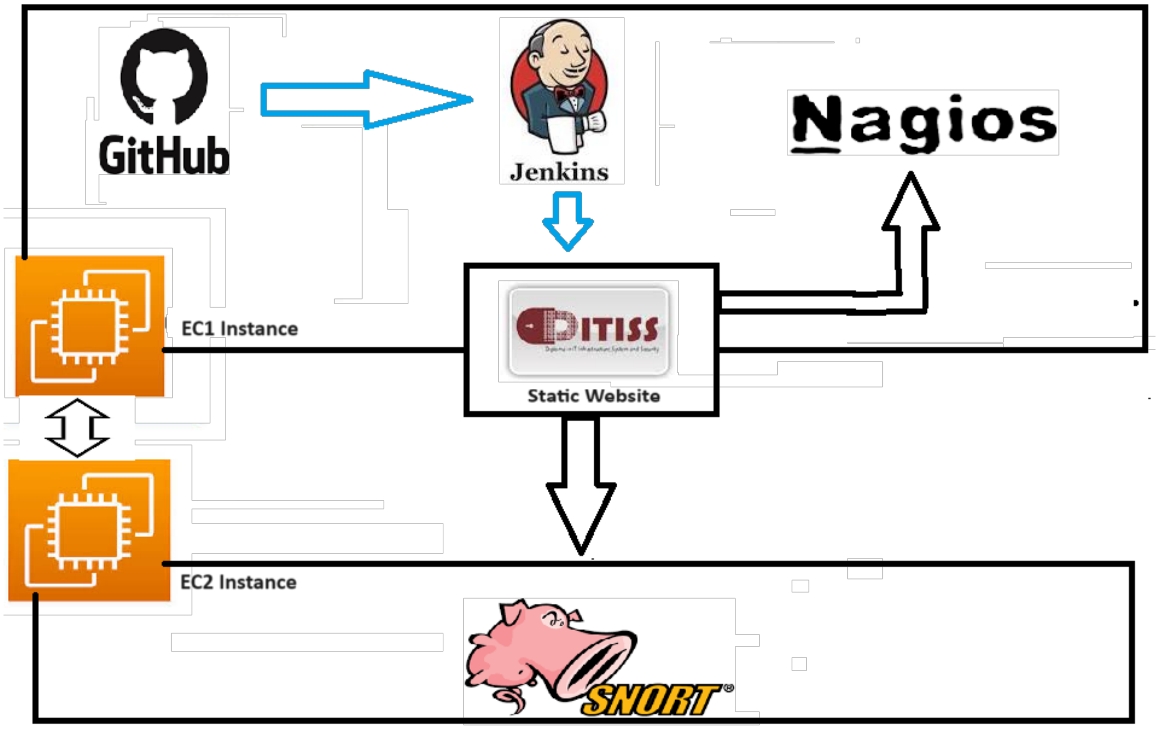


Figure: Flowchart

3.3 Technology used

3.3.1 Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) is a web service provided by Amazon it allows you to rent virtual servers in the cloud, known as instances, to run your applications and workloads. EC2 provides a scalable and flexible infrastructure that enables you to quickly deploy and manage virtual servers without the need to invest in physical hardware.

Key features

Instances: Elastic Compute Cloud (EC2) instances on Amazon Web Services (AWS) are virtual servers that offer cloud computing capability that is scalable. Running programs on virtual computers without having to buy actual hardware is the main purpose of EC2 instances. Depending on their requirements for processing, memory, and storage, users can select from a variety of instance kinds. EC2's goal is to provide users with scalable, adaptable computing capacity so they can simply grow their applications and only pay for the resources they really need.

Elasticity and Scaling: EC2 allows you to easily scale your infrastructure up or down based on your workload's demands. You can create multiple instances or change the instance type to handle varying levels of traffic.

Cost Management: EC2 offers various pricing options, including On-Demand Instances, Reserved Instances, and Spot Instances, which provide flexibility in managing costs based on your usage patterns.

3.3.2 Git

Git is a distributed version control system (VCS) designed to manage source code history and facilitate collaborative software development.

Key features of Git:

Distributed Architecture: Unlike centralized version control systems, Git is distributed. Each developer has a complete copy of the repository, including its entire history. This allows for offline work, faster operations, and improved resilience.

Branching and Merging: Git makes it easy to create branches, which are separate lines of development. Developers can work on features, bug fixes, or experiments in their own branches without affecting the main codebase. Merging branches back together is relatively simple and allows for collaborative development.

Commit History: Git maintains a detailed history of changes to the codebase. Each change is represented by a commit, which includes information about who made the change, when it was made, and what was changed. This commit history provides a clear view of the evolution of the project.

Fast and Efficient: Git is designed for speed and efficiency. Most operations are local, as the repository resides on the developer's machine. This results in rapid commits, branching, and merging.

Collaboration: Git enables effective collaboration among developers. Multiple developers can work on different branches simultaneously, and changes can be shared by pushing them to a remote repository. Pull requests or merge requests facilitate the process of reviewing and integrating changes from different contributors.

3.3.3 Jenkins

Jenkins is an open-source automation server that facilitates the continuous integration and continuous delivery (CI/CD) of software projects. It helps automate various tasks related to building, testing, and deploying applications, making the development and release process more efficient and reliable.

Key features of Jenkins:

Continuous Integration: Jenkins automates the process of integrating code changes from multiple contributors into a shared repository. It triggers builds whenever code is committed, allowing developers to identify and fix integration issues early.

Automated Builds: Jenkins can automatically build projects from source code repositories. It supports various build tools, languages, and platforms, making it versatile for different types of projects.

Extensibility: Jenkins can be extended through a wide range of plugins that provide additional functionalities. Plugins are available for source code management, build tools, testing frameworks, and deployment options.

Pipeline as Code: Jenkins uses a domain-specific language called Groovy to define build pipelines as code. This enables you to define complex workflows that include build, test, and deployment stages in a version-controlled script.

Continuous Delivery: Jenkins supports continuous delivery by automating the deployment process after successful builds. It can deploy applications to different environments, such as development, staging, and production.

Distributed Builds: Jenkins can distribute builds across multiple machines, allowing for parallel builds and improved build performance. This is particularly useful for large and resource-intensive projects.

3.3.4 CI/CD Pipeline

CI/CD pipelines automate the process of software development, from integration (CI) to delivery and deployment (CD). In a CI/CD pipeline, code changes are automatically built, tested, and deployed to production. This ensures that new features, bug fixes, and updates can be released quickly and reliably. The use of CI/CD pipelines reduces manual errors, speeds up development, and allows for frequent and consistent delivery of high-quality software.

3.3.5 Snort

Snort's primary goal is to identify and stop network intrusions. An open-source intrusion detection and prevention system (IDS/IPS) called Snort keeps an eye on network traffic in real time and uses a set of pre-established rules to analyse it in order to spot potentially dangerous or suspicious activity. In order to help defend networks from assaults, Snort can log information about potential threats, notify administrators, or even take automated action to block the malicious traffic.

Key features of Snort include:

Packet Analysis: Snort inspects network packets as they pass through a network interface, analyzing their content, headers, and metadata to identify potential security threats.

Rule-Based Detection: Snort uses a rule-based system to detect specific patterns or signatures associated with known attack methods. Rules define the conditions under which an alert is generated.

Customizable Rules: Snort allows you to create custom rules based on your network environment and the threats you want to detect. This flexibility enables you to tailor the IDS to your specific needs.

Anomaly Detection: In addition to signature-based detection, Snort can also detect anomalies in network traffic behavior. This is useful for identifying new or previously unknown attacks.

Logging and Alerting: Snort generates alerts when it identifies suspicious or malicious activity. These alerts can include information about the type of attack, source and destination IP addresses, and other relevant details.

Rule Actions: Snort rules specify actions to take when a match is found. Actions can include logging the event, generating an alert, or even blocking or dropping packets.

3.3.6 Nagios

Nagios offers comprehensive monitoring for IT infrastructure, including servers, networks, applications, and services.Its robust alerting system notifies administrators of issues promptly, enabling quick response to critical issues.Nagios proactively identifies potential problems, minimizing downtime and enhancing system reliability.Centralized management simplifies administration tasks, improving efficiency across multiple systems and services.Highly scalable, Nagios supports distributed monitoring setups, effectively accommodating growing infrastructure needs.

Key features of Nagios include:

Monitoring Hosts and Services: Nagios can monitor various types of hosts (servers, devices) and services (applications, network services) by regularly checking their availability and responsiveness.

Alerting: Nagios generates alerts when it detects that a monitored host or service has a problem. Alerts can be sent via email, SMS, or other notification methods to ensure timely response and issue resolution.

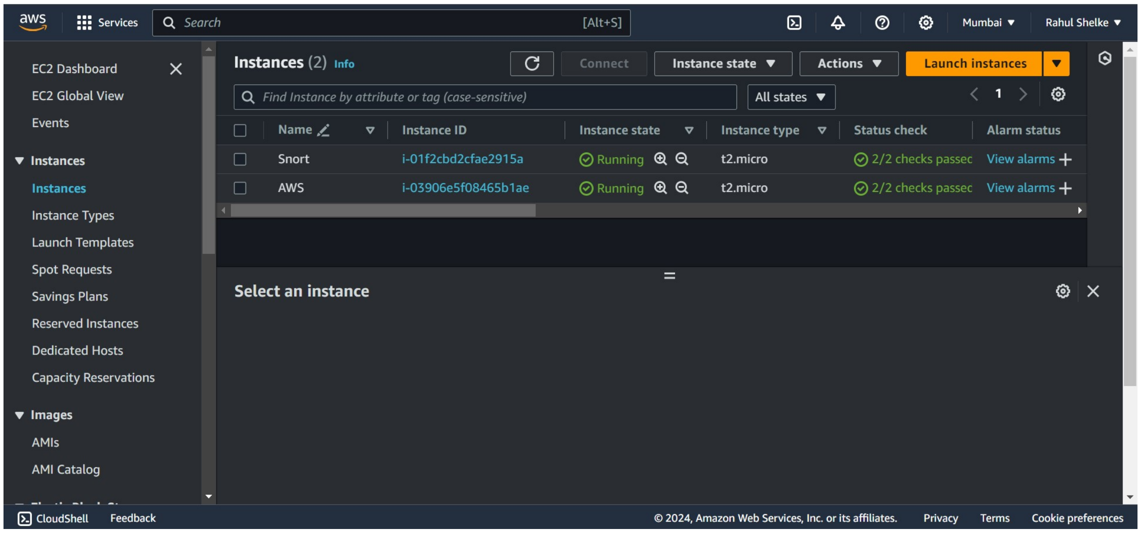
Threshold Monitoring: Nagios enables you to define thresholds for various metrics (CPU usage, memory usage, response time) and generate alerts when those thresholds are exceeded.

Flexible Notification: Nagios supports flexible notification configurations, allowing you to define who should be notified based on the time of day, the severity of the issue, and other criteria.

Plugins: Nagios uses plugins to perform monitoring checks. There are a wide variety of pre-built plugins available, and you can also create custom plugins to monitor specific aspects of your environment.

4. Project Output

4.1 AWS EC2 Instances



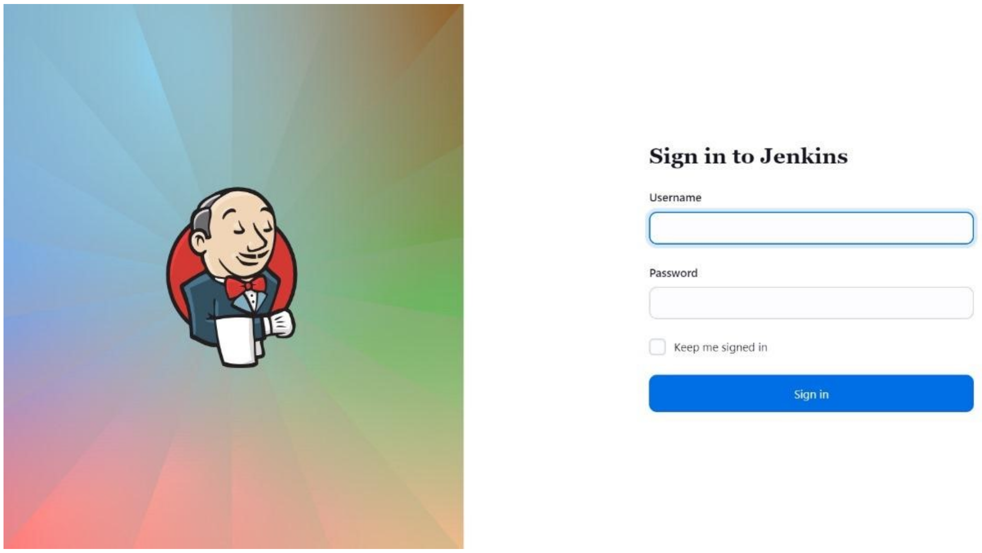
Instances (2) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
□	Snort	i-01f2cbd2cfae2915a	Running	t2.micro	2/2 checks passec	View alarms +
□	AWS	i-03906e5f08465b1ae	Running	t2.micro	2/2 checks passec	View alarms +

Select an instance

© 2024, Amazon Web Services, Inc. or its affiliates.

4.2 Jenkins



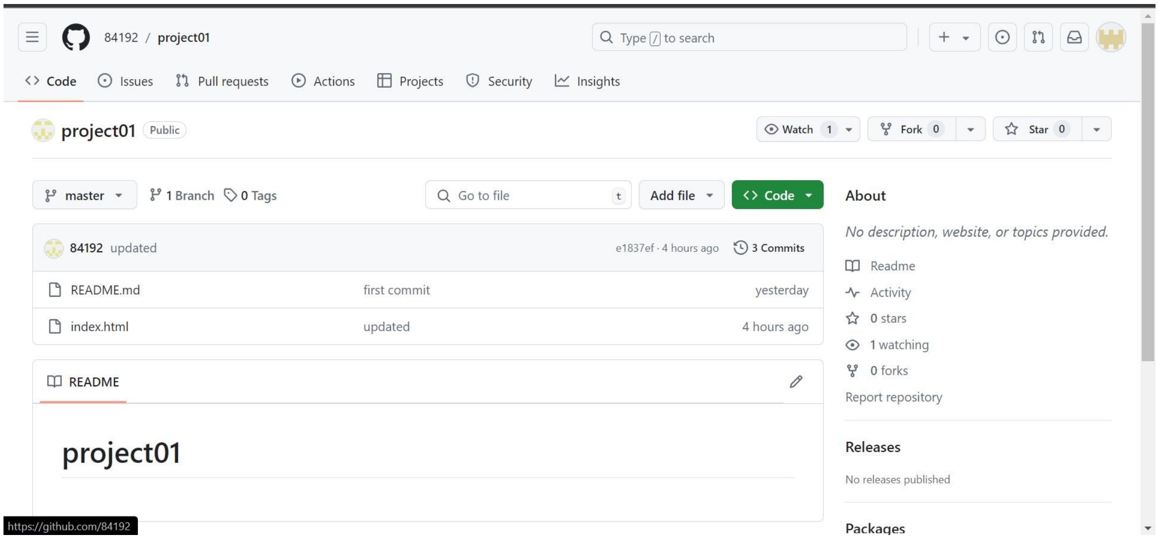
Sign in to Jenkins

Username

Password

□ Keep me signed in

4.3 GitHub



84192 updated e1837ef · 4 hours ago 3 Commits

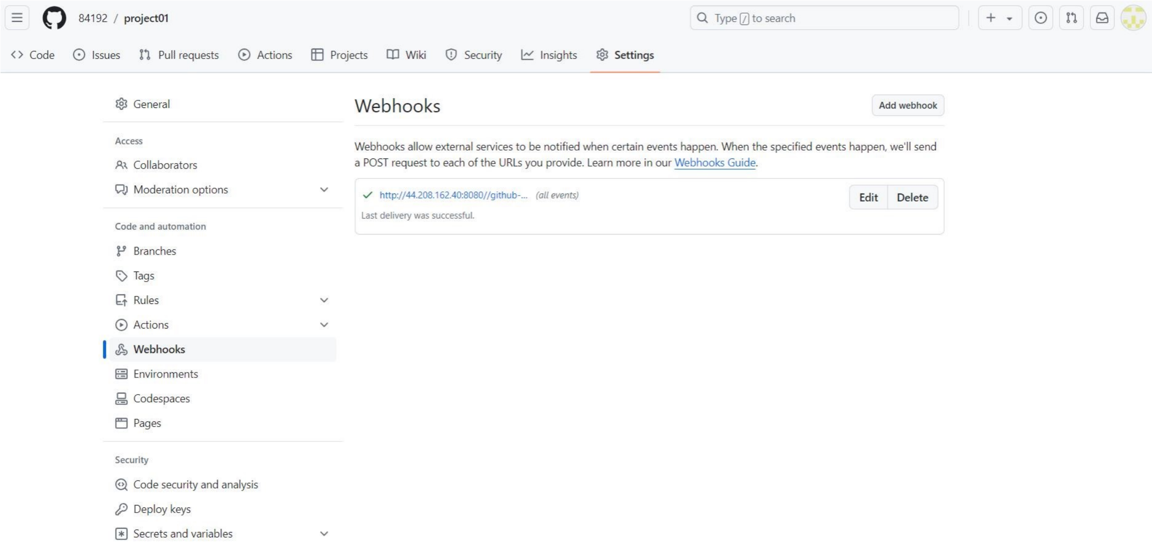
README.md	first commit	yesterday
index.html	updated	4 hours ago

No description, website, or topics provided.

No releases published

project01

4.4 Webhook



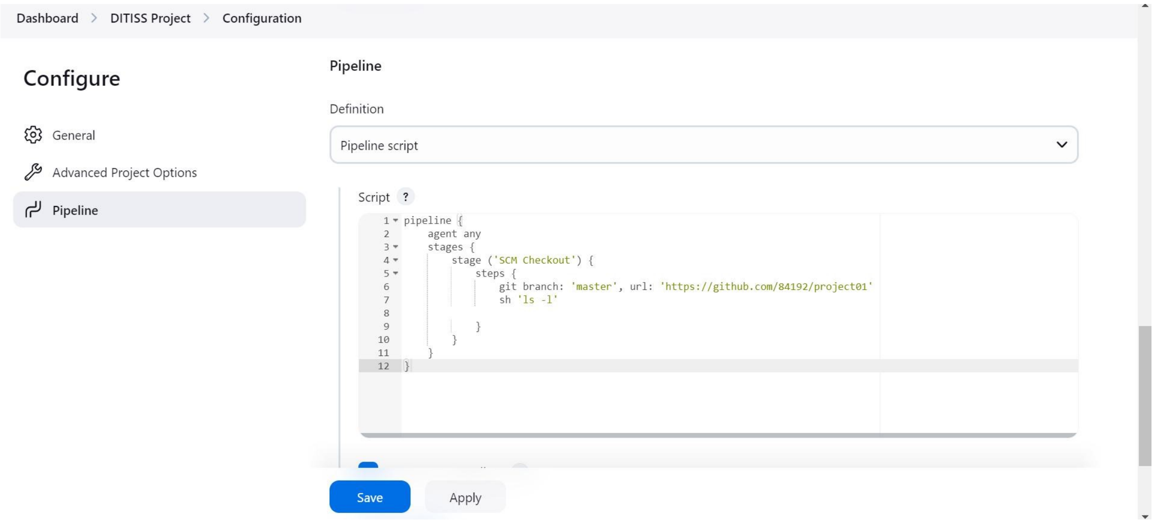
Webhooks

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

✓ http://44.208.162.40:8080//github-... (all events)

Last delivery was successful.

4.5 Pipeline Creation



Pipeline

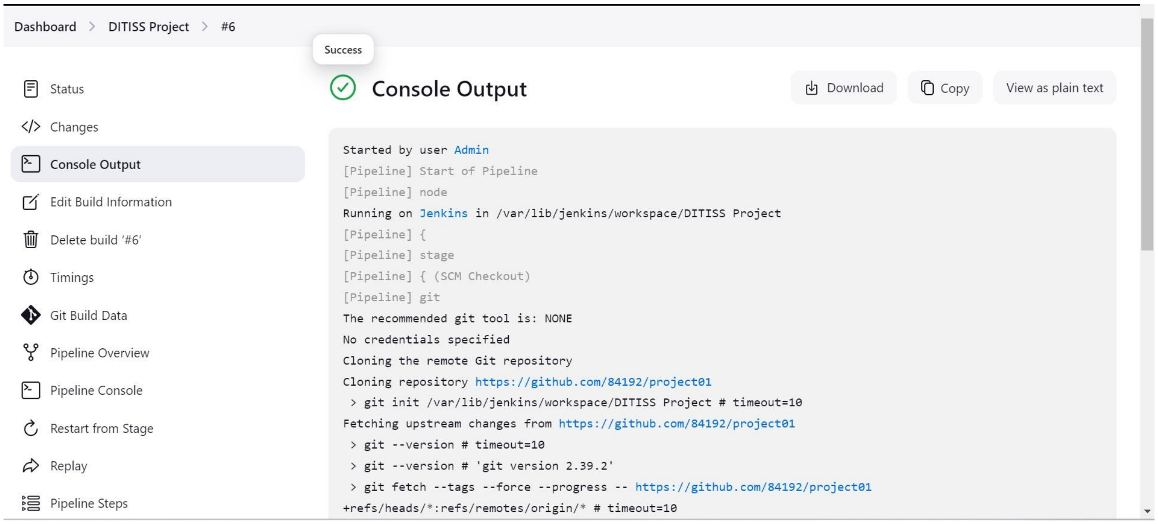
Definition

Pipeline script

Script ?

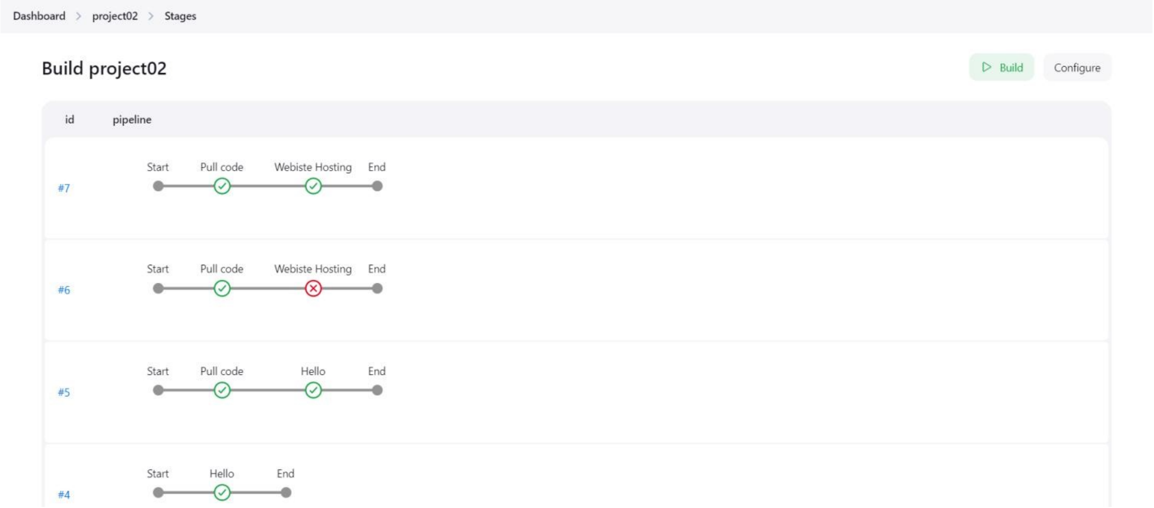
```
pipeline {
    agent any
    stages {
        stage ('SCM Checkout') {
            steps {
                git branch: 'master', url: 'https://github.com/84192/project01'
                sh 'ls -l'

            }
        }
    }
}
```

4.6 Console Output



```
Started by user Admin
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/DITISS Project
[Pipeline] {
[Pipeline] stage
[Pipeline] { (SCM Checkout)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/84192/project01
 > git init /var/lib/jenkins/workspace/DITISS Project # timeout=10
Fetching upstream changes from https://github.com/84192/project01
 > git --version # timeout=10
 > git --version # 'git version 2.39.2'
 > git fetch --tags --force --progress -- https://github.com/84192/project01
+refs/heads/*:refs/remotes/origin/* # timeout=10
```

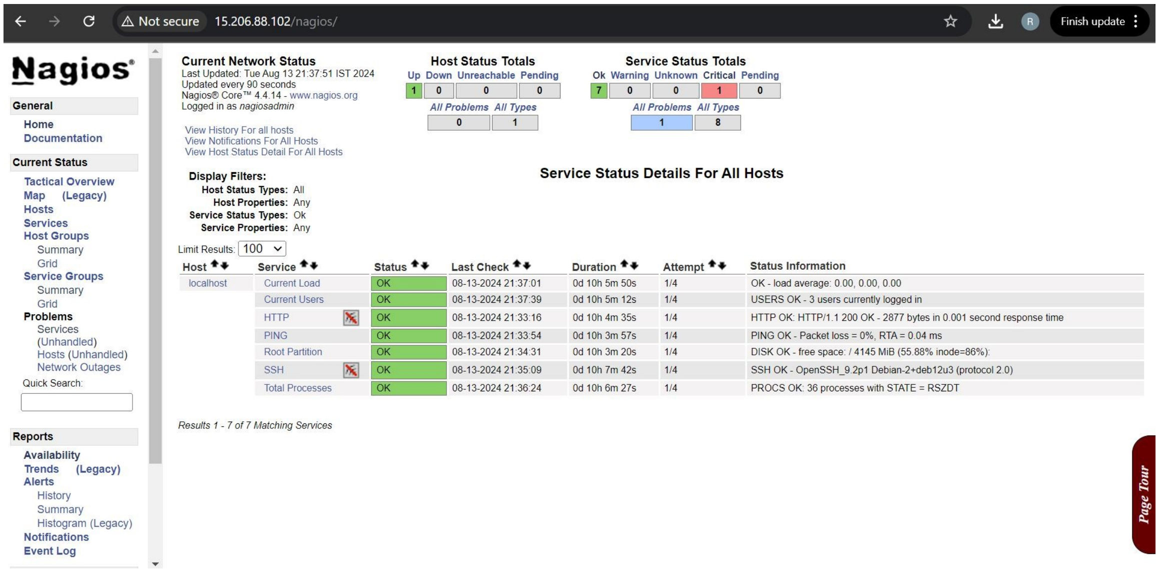
4.7 Build Stages



Build project02

id	pipeline
#7	Start → Pull code ✓ → Webiste Hosting ✓ → End
#6	Start → Pull code ✓ → Webiste Hosting ✗ → End
#5	Start → Pull code ✓ → Hello ✓ → End
#4	Start → Hello ✓ → End

4.8 Nagios



Current Network Status
Last Updated: Tue Aug 13 21:37:51 IST 2024
Updated every 90 seconds
Nagios® Core™ 4.4.14 - www.nagios.org
Logged in as *nagiosadmin*

View History For all hosts
View Notifications For All Hosts
View Host Status Detail For All Hosts

Host Status Totals

Up	Down	Unreachable	Pending
1	0	0	0

All Problems	*All Types*
0	1

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
7	0	0	1	0

All Problems	*All Types*
1	8

Display Filters:
Host Status Types: All
Host Properties: Any
Service Status Types: Ok
Service Properties: Any

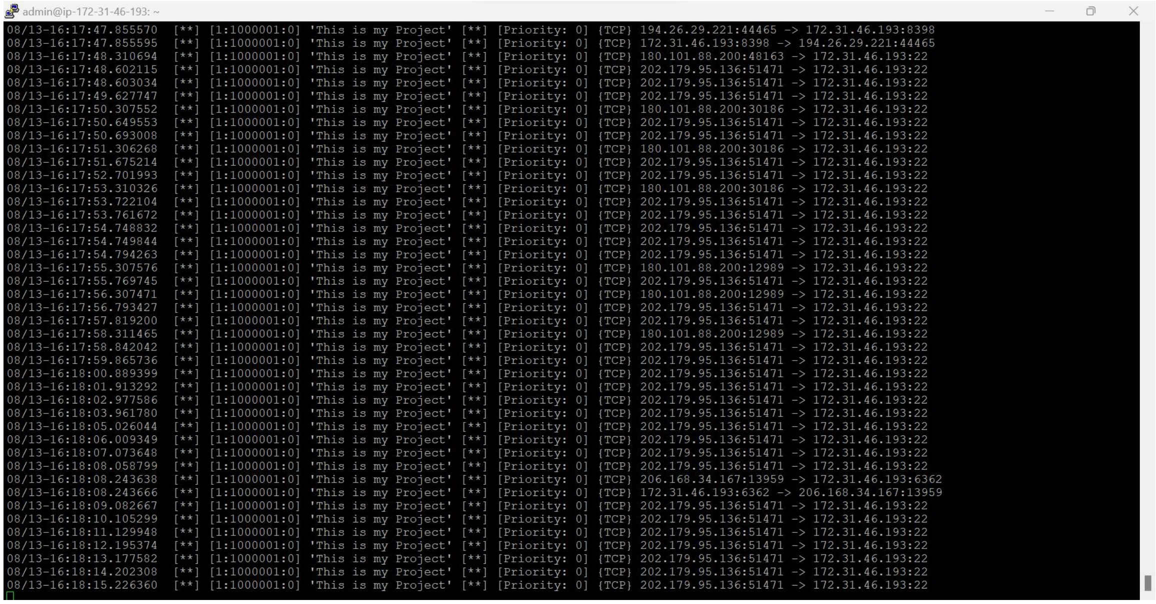
Service Status Details For All Hosts

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	08-13-2024 21:37:01	0d 10h 5m 50s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	08-13-2024 21:37:39	0d 10h 5m 12s	1/4	USERS OK - 3 users currently logged in
	HTTP	OK	08-13-2024 21:33:16	0d 10h 4m 35s	1/4	HTTP OK: HTTP/1.1 200 OK - 2877 bytes in 0.001 second response time
	PING	OK	08-13-2024 21:33:54	0d 10h 3m 57s	1/4	PING OK - Packet loss = 0%, RTA = 0.04 ms
	Root Partition	OK	08-13-2024 21:34:31	0d 10h 3m 20s	1/4	DISK OK - free space: / 4145 MiB (55.88% inode=86%):
	SSH	OK	08-13-2024 21:35:09	0d 10h 7m 42s	1/4	SSH OK - OpenSSH_9.2p1 Debian-2+deb12u3 (protocol 2.0)
	Total Processes	OK	08-13-2024 21:36:24	0d 10h 6m 27s	1/4	PROCS OK: 36 processes with STATE = RSZDT

Results 1 - 7 of 7 Matching Services

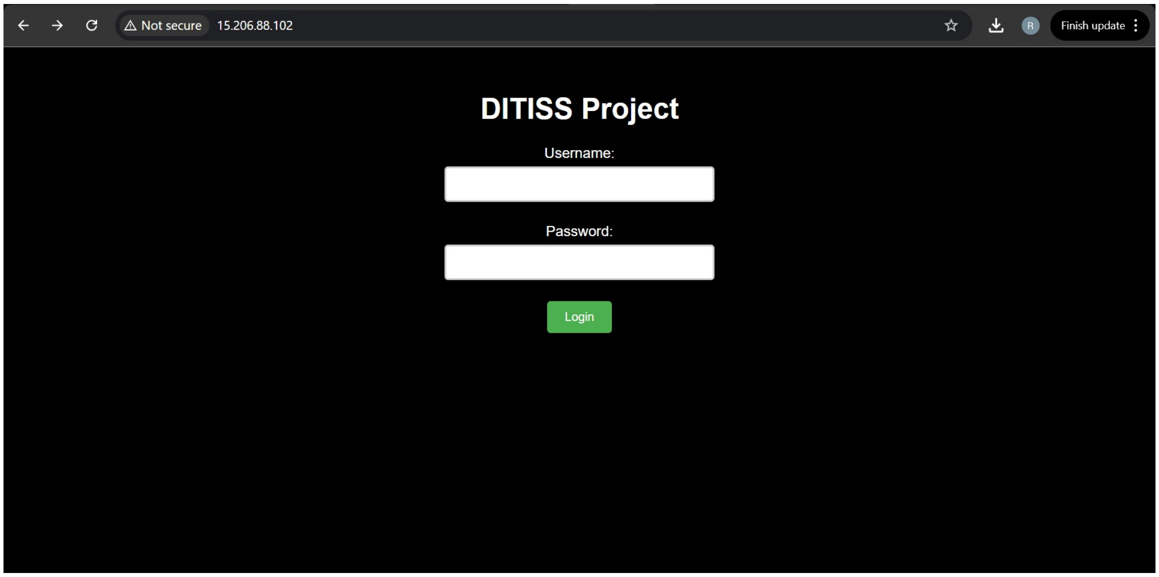
4.9 Snort



```
08/13-16:17:47.855570  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 194.26.29.221:44465 -> 172.31.46.193:8398
08/13-16:17:47.855595  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 172.31.46.193:8398 -> 194.26.29.221:44465
08/13-16:17:48.310694  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 180.101.88.200:48163 -> 172.31.46.193:22
08/13-16:17:48.602115  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:17:48.603034  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:17:49.627747  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:17:50.307552  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 180.101.88.200:30186 -> 172.31.46.193:22
08/13-16:17:50.649553  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:17:50.693008  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:17:51.306268  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 180.101.88.200:30186 -> 172.31.46.193:22
08/13-16:17:51.675214  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:17:52.701993  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:17:53.310326  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 180.101.88.200:30186 -> 172.31.46.193:22
08/13-16:17:53.722104  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:17:53.761672  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:17:54.748832  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:17:54.749844  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:17:54.794263  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:17:55.307576  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 180.101.88.200:12989 -> 172.31.46.193:22
08/13-16:17:55.769745  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:17:56.307471  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 180.101.88.200:12989 -> 172.31.46.193:22
08/13-16:17:56.793427  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:17:57.819200  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:17:58.311465  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 180.101.88.200:12989 -> 172.31.46.193:22
08/13-16:17:58.842042  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:17:59.865736  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:18:00.889399  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:18:01.913292  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:18:02.977586  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:18:03.961780  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:18:05.026044  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:18:06.009349  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:18:07.073648  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:18:08.058799  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:18:08.243638  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 206.168.34.167:13959 -> 172.31.46.193:6362
08/13-16:18:08.243666  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 172.31.46.193:6362 -> 206.168.34.167:13959
08/13-16:18:09.082667  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:18:10.105299  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:18:11.129948  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:18:12.195374  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:18:13.177582  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:18:14.202308  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
08/13-16:18:15.226360  [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 202.179.95.136:51471 -> 172.31.46.193:22
```

4.10 Web Application Outputs

4.10.1



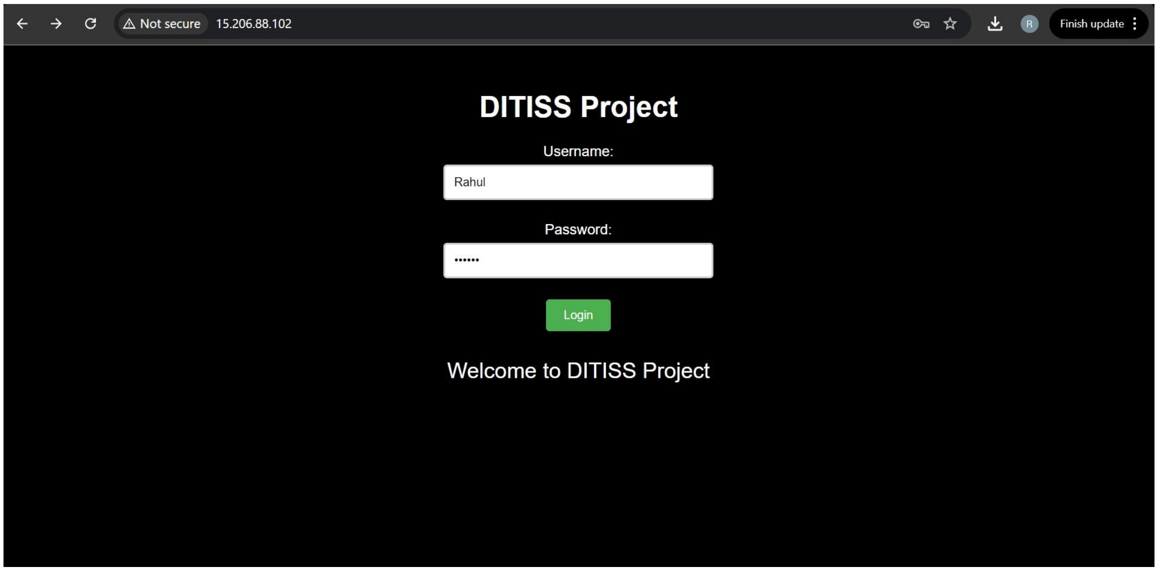
DITISS Project

Username:

Password:

Login

4.10.2



DITISS Project

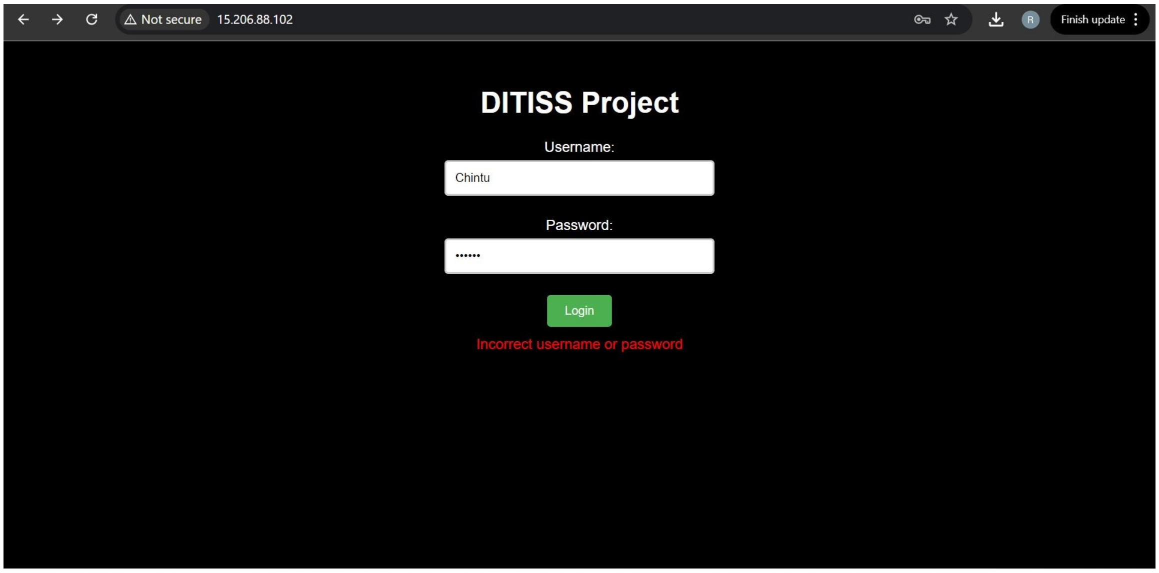
Username: Rahul

Password: ••••••

Login

Welcome to DITISS Project

4.10.3



DITISS Project

Username: Chintu

Password: ••••••

Login

Incorrect username or password

5. CONCLUSION

5.1 Conclusion

Hence, we have successfully deployed a highly available and secure web server environment on Amazon Web Services (AWS). And ensured the reliability, performance, and security of the web application while maintaining efficient development and operational processes.

5.2 Future Scope

The future scope for Secure Web Application Deployment on AWS with Jenkins CI/CD, Nagios monitoring, and Snort IDS involves advancing automation, security, and scalability. Enhanced integration of Jenkins with Infrastructure-as-Code (IaC) and serverless technologies will streamline deployments and improve efficiency. Advanced threat detection and response can be achieved by integrating Snort with machine learning and other security tools, while comprehensive monitoring solutions combining Nagios

with real-time analytics will offer better visibility and proactive management. Additionally, adopting containerization, improving disaster recovery strategies, and ensuring compliance with evolving regulations will further bolster the resilience and security of web applications

REFERENCES

Paper 1: - A Qualitative Study of DevOps Usage in Practice

Author: Floris Erich, C. Amrit & M. Daneva

Paper 2: - Devops, A New Approach To Cloud Development & Testing

Author: Dhaya Sindhu Battina

Paper 3: - Review paper on Snort and reviewing its applications in different fields

Author: Harpreet Sandhu, Manpreet Kaur.