

Project Report on

**Web Server Deployment on AWS with Nagios Monitoring, and IDS Security tool**

**Submitted by**

**Priyanka Pawar (240344223026)**

**Sanghamitra Gawai (240344223033)**

**Astha Bhende (240344223042)**

**Nidhi Patil (240344223022)**

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 -2023**

# 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:

**Nidhi Patil Priyanka Pawar Sanghamitra Gawai Astha Bhende**

**(240344223022) (240344223026) (240344223033) (240344223042)**

# CERTIFICATE

This is to certify that the project report entitled **“Web Server Deployment on AWS with Nagios Monitoring, and IDS Security ”**, submitted by **Nidhi Patil** 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 Server Deployment on AWS with Nagios Monitoring, and IDS Security”**, submitted by **Priyanka Pawar** 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 Server Deployment on AWS with Nagios Monitoring, and IDS Security ”**, submitted by **Sanghamitra Gawai** 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 Server Deployment on AWS with Nagios Monitoring and IDS Security”**, submitted by **Astha Bhende** 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 Server Deployment on AWS with Nagios Monitoring and IDS Security”** by **Nidhi Patil (240344223022)** 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 Server Deployment on AWS with Nagios Monitoring and IDS Security”** by **Priyanka Pawar (240344223026)** 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 Server Deployment on AWS with Nagios Monitoring and IDS Security”** by **Sanghamitra Gawai (240344223033)** 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 Server Deployment on AWS with Nagios Monitoring and IDS Security”** by **Astha Bhende** **(240344223042)** 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

|  |  |
| --- | --- |
| **TITLE** | **PAGE NO** |
| **Declaration** |  |
| **Certificate** |  |
| **Approval Certificate** |  |
| **Abstract** | i |
| **1.INTRODUCION** | 1 |
| 1.1 Applications | 2 |
| 1.2 Organization and Project Plan | 3 |
| **2. LITERATURE SURVEY** | 4 |
| Paper 1 | 4 |
| Paper 2 | 4 |
| Paper 3 | 5 |
| **3. SYSTEM DEVELOPMENT AND DESIGN** | 6 |
| 3.1 Proposed System | 6 |
| 3.2 Flow Chart | 7 |
| 3.3 Technology used | 8 |
| 3.3.1 AWS EC2 | 8 |
| 3.3.2 Git | 9 |
| 3.3.3 Docker | 10 |
| 3.3.4 Jenkins | 11 |
| 3.3.5 Nessus | 12 |
| 3.3.6 Nagios | 14 |
| 3.3.7 Snort | 15 |
| **4. PROJECT OUTPUT** | 17 |
| **5. CONCLUSION** | 21 |
| 5.1 Conclusion | 21 |
| 5.2 Future Scope | 21 |
| **REFERENCES** | 22 |

# 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 vigilant monitoring, and an Intrusion Detection System (IDS) for heightened security.

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

The integration of CI/CD pipelines streamlines the deployment process and ensures the seamless delivery of application updates. By employing tools such as AWS Code Pipeline and Code Deploy, 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.

# INTRODUCTION

In the ever-evolving landscape of web applications, the process of deploying, managing, and safeguarding online services has become increasingly intricate and critical. This project introduces a comprehensive approach to web server deployment, employing Amazon Web Services (AWS) Elastic Compute Cloud (EC2) for hosting, integrating continuous integration and delivery (CI/CD) pipelines 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 servers 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 server deployment process. By employing AWS Code Pipeline and Code Deploy, we illustrate how developers can automate code testing, deployment, and monitoring, leading to consistent and reliable application updates.

Nagios Monitoring for Proactive Oversight:

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 pipelines, 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.

## Applications

* E-Commerce Platforms: Online stores and e-commerce platforms require reliable web server deployments to handle high traffic loads, ensure seamless user experiences, and protect sensitive customer data.
* Finance and Banking Systems: In the finance sector, security and performance are critical. Web server deployment with CI/CD ensures that updates to financial applications are rigorously tested and deployed smoothly.
* Gaming and Entertainment Platforms: Online gaming and entertainment platforms need scalable web server deployments to handle sudden surges in user activity. CI/CD enables game updates and feature releases without disrupting user experiences. Nagios monitoring guarantees the availability of gaming services, while the Snort IDS safeguards against cheating and hacking attempts.

## 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.

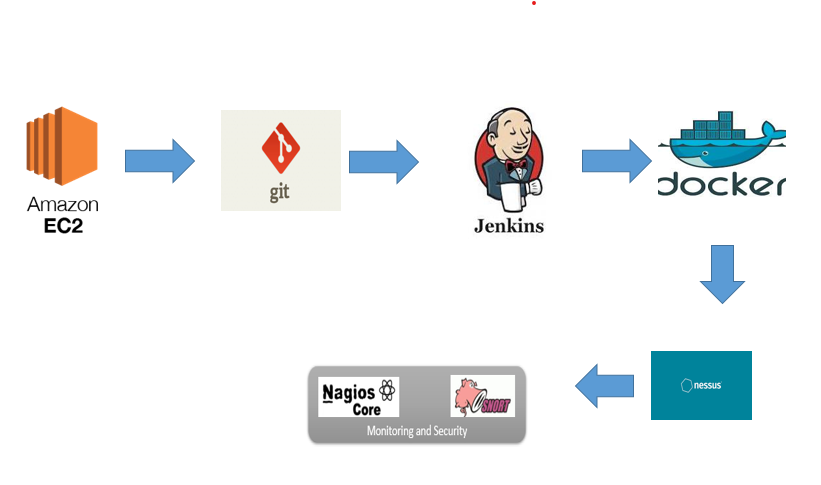
# SYSTEM DEVELOPMENT AND DESIGN

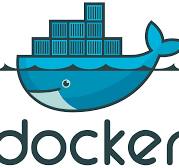
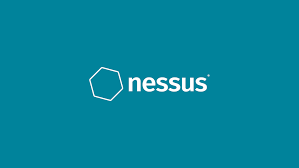
## 3.1 Proposed System

We propose a system where we are setting up two Amazon EC2 instances to host your web application. Storing our application code in a version-controlled repository (Git-Hub).

Set up a CI/CD pipeline using Jenkins. On code changes, trigger an automated build and deployment process using Jenkins. Setting up another EC2 instance to host the Nagios monitoring server. Install Nagios plugins to monitor various aspects of your infrastructure, such as server health, resource usage, and application responsiveness. Configuring AWS Security Groups to control inbound and outbound traffic to our EC2 instances. And also configuring Snort rules to detect various types of network traffic anomalies and security threats.

**Architecture:**

****

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: These are virtual servers that you can launch and manage in the cloud. You can choose from a wide range of instance types optimized for different use cases, such as general-purpose computing, memory-intensive tasks, high-performance computing, and more.

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.

Amazon Machine Images (AMIs): AMIs are pre-configured templates that contain an operating system and often additional software needed to launch an instance. You can use AWS-provided AMIs or create your own custom AMIs.

Security Groups: Security groups act as virtual firewalls that control inbound and outbound traffic to your instances. You can define rules to allow or deny specific types 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 code base. 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 Docker

Docker is an open-source platform that allows you to automate the deployment, scaling, and management of applications using containerization. Containers are lightweight, portable, and isolated environments that package an application and its dependencies together.

**Key features of Docker:**

**Containerization:** Docker allows you to create containers that encapsulate applications and their dependencies, including libraries, runtime, and system tools. Containers ensure consistency between different environments, from development laptops to production servers.

**Portability:** Docker containers can run consistently across different environments, such as local development machines, testing servers, and cloud-based production environments. This eliminates the "it works on my machine" problem.

**Efficiency:** Containers share the host operating system's kernel and resources, which makes them more lightweight and efficient compared to traditional virtual machines. This leads to faster startup times and reduced resource consumption.

**Version Control:** Docker images are versioned, allowing you to track changes to your application and its dependencies over time. This facilitates rollback and ensures that you can recreate previous versions of your application easily.

**Docker Images:** Docker images are the blueprints for containers. They are created from a set of instructions defined in a Dockerfile, which specifies the base image, environment, configuration, and application code.

### 3.3.4 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.

### 3.3.5 Nessus

### Nessus is a comprehensive and widely utilized vulnerability assessment tool developed by Tenable. Its primary function is to identify and evaluate security vulnerabilities in computer systems, networks, and applications. Nessus helps organizations detect potential weaknesses before they can be exploited by attackers, thereby enhancing overall cybersecurity.

**Key features of Nessus:**

**Vulnerability Scanning:**

**Network Scanning:** Detects vulnerabilities in network devices, servers, and workstations.

**Web Application Scanning:** Identifies weaknesses in web applications, including common vulnerabilities such as SQL injection and cross-site scripting (XSS).

**Configuration Scanning:** Assesses system configurations for compliance with security policies and best practices.

**Comprehensive Plugin Library:**

Nessus uses a plugin-based architecture with thousands of plugins that are regularly updated. These plugins help in detecting known vulnerabilities, misconfigurations, and compliance issues.

**Customizable Scanning:**

Users can create custom scan policies to tailor scans to specific needs, including selecting specific ports, services, or types of vulnerabilities.

**Automated Scanning:**

Supports automated and scheduled scans, allowing for regular assessment without manual intervention.

**Detailed Reporting:**

Provides detailed reports on discovered vulnerabilities, including severity, affected systems, and recommended remediation steps. Reports can be customized and exported in various formats like PDF, HTML, and CSV.

**Real-Time Updates:**

Nessus regularly updates its vulnerability database to include the latest threats and vulnerabilities, ensuring that scans are effective against the most recent exploits.

**Compliance Checks:**

Includes templates for compliance checks against various standards and regulations, such as PCI-DSS, HIPAA, and CIS benchmarks.

**Advanced Features:**

**Credentialed Scanning:** Allows for deeper scanning by logging into systems with valid credentials to uncover vulnerabilities that are not detectable from the network alone.

**Vulnerability Analysis:** Provides analysis and prioritization of vulnerabilities based on risk factors like exploitability and impact.

**Integration Capabilities:**

Integrates with other security tools and platforms for enhanced security management, including SIEM systems and ticketing systems for automated incident management.

**User-Friendly Interface:**

Nessus features an intuitive web-based interface that makes it easier for users to configure scans, view results, and generate reports.

**3.3.6Nagios**

### Nagios is an open-source monitoring system that provides comprehensive monitoring and alerting capabilities for IT infrastructure components. It helps organizations monitor the health and performance of their networks, servers, applications, and services, enabling proactive identification and resolution of issues before they impact business operations.

### 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.

# 3.3.7 Snort

# Snort is an open-source Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) that is widely used for network security monitoring. It helps detect and prevent unauthorized access, malicious activities, and security threats within network traffic. Snort is designed to analyze network packets in real-time and generate alerts when it identifies patterns that match known attack signatures or behavior anomalies.

# 

# 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

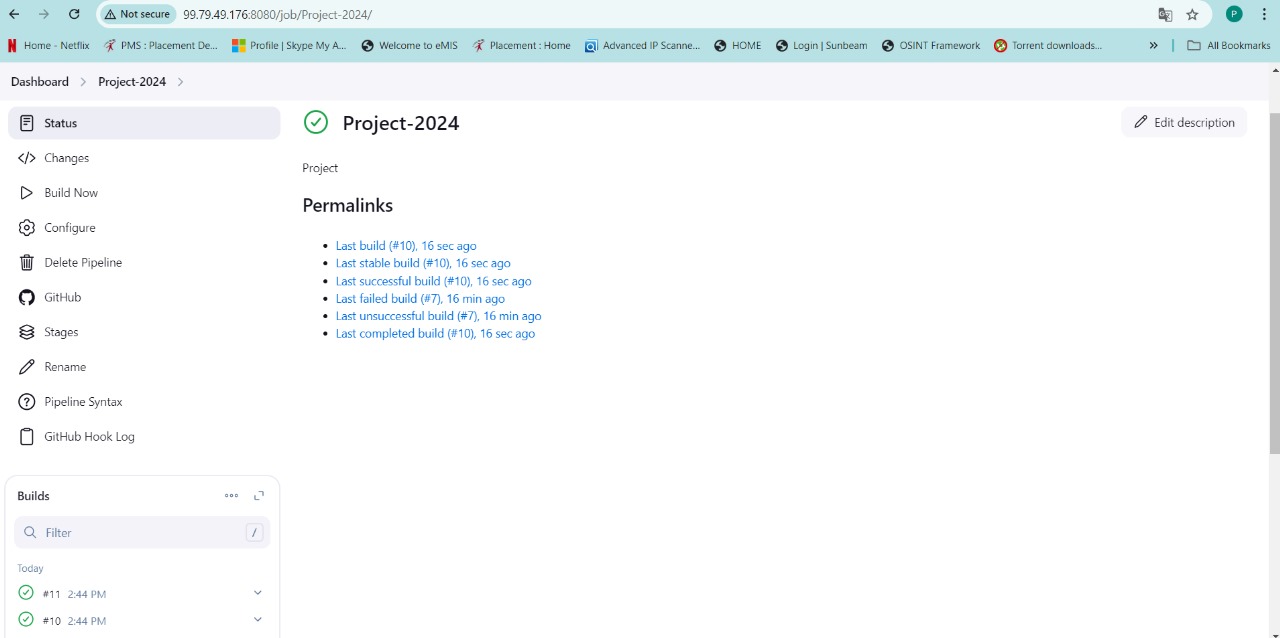
# .

# 4. Project Output

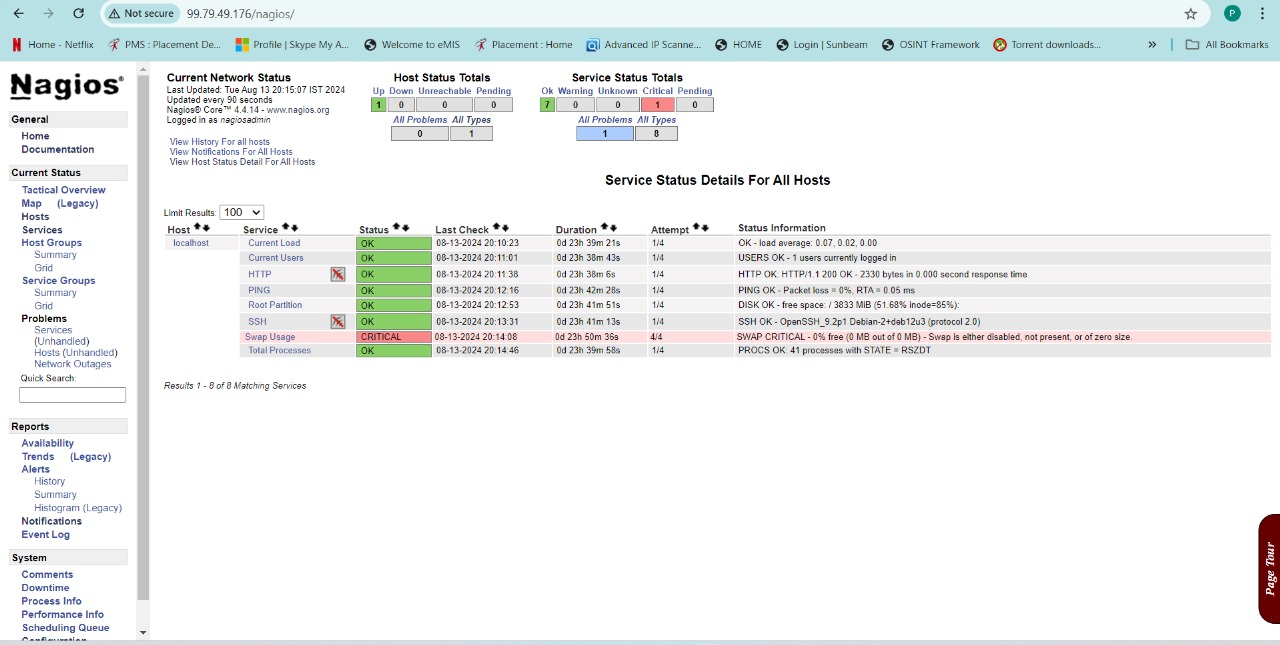
## AWS

## 

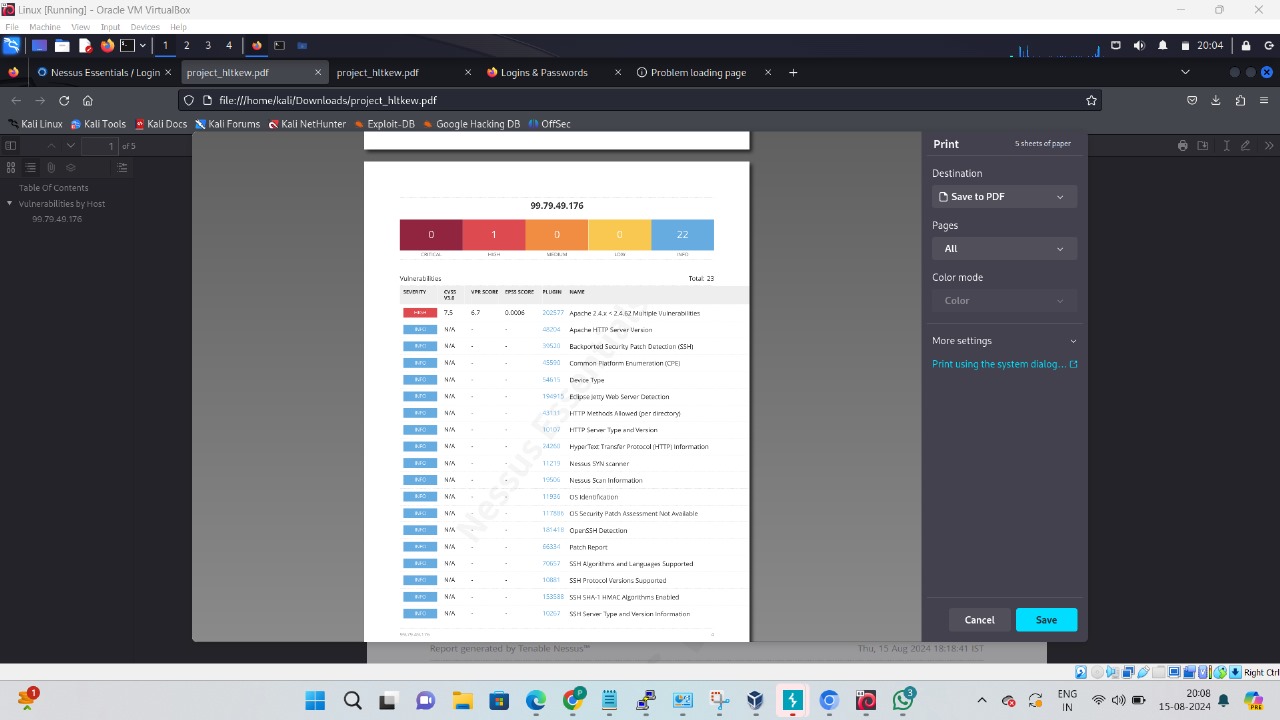
**4.2 Jenkins :**



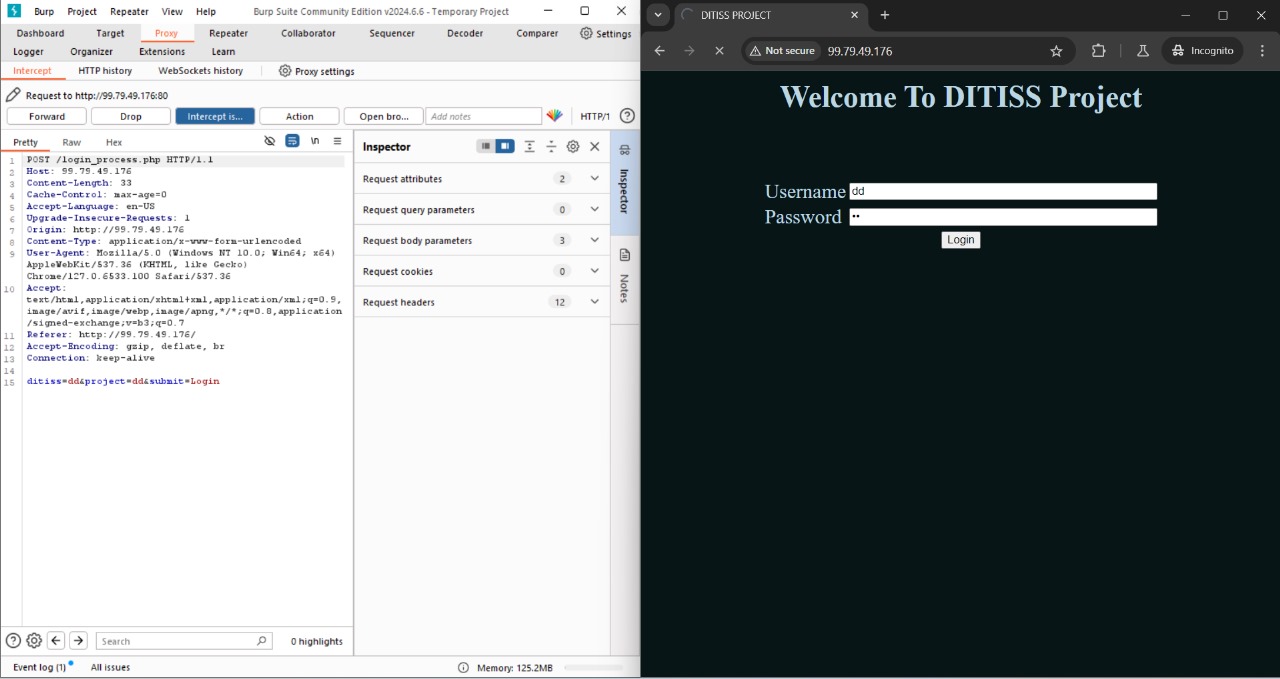
**4.3 Nagios**

****

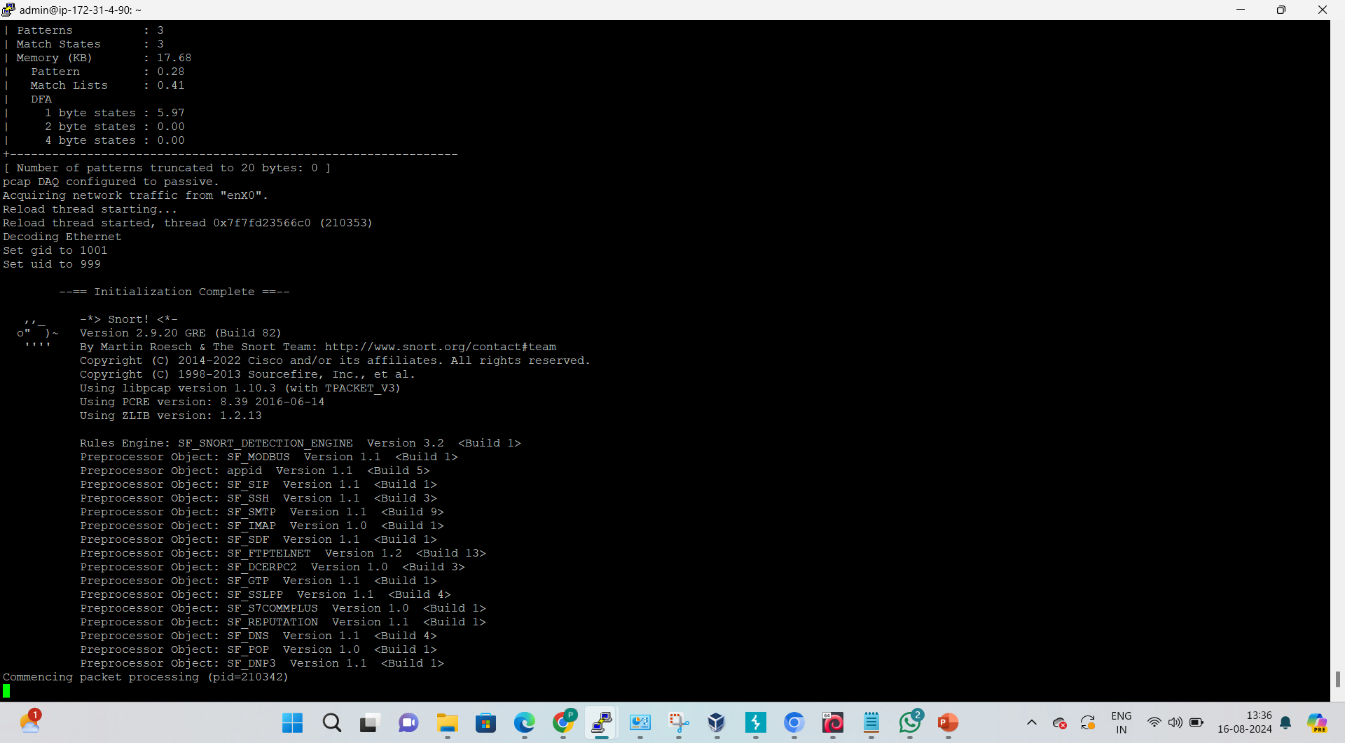
**4.4 Nessus**



**4.5 Attack**



**4.6 Snort**



# 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

Docker is majorly considered as a best solution for service availability. It can be attached to implement continuous integration and continuous development i.e., CICD. It can be used in development when the software is getting develop this tool can used there to continuous security evaluation of software so that developers can program it more securely. As we have developed this tool for small scale. In future It can be used for large scale.

# 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.