Project Report on

**End to End Security Automation in CI/CD pipeline with Monitoring Alerts and Patch Management**

**Submitted by**

**Prathamesh Karve : 240344223029**

**Vineet Patil : 240344223025**

**Shrikant Raut : 240344223030**

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:

**Prathamesh Karve Vineet Patil Shrikant Raut:**

**(240344223029) (240344223025) (240344223030)**

**CERTIFICATE**

This is to certify that the project report entitled “**End to End Security Automation in CI/CD pipeline with Monitoring Alerts and Patch Management**”, submitted by **Prathamesh Karve** 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 “**End to End Security Automation in CI/CD pipeline with Monitoring Alerts and Patch Management**”, submitted by **Vineet 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 “**End to End Security Automation in CI/CD pipeline with Monitoring Alerts and Patch Management**”, submitted by **Shrikant Raut** 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 “**End to End Security Automation in CI/CD pipeline with Monitoring Alerts and Patch Management**” by **Prathamesh Karve** (240344223029) 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 “**End to End Security Automation in CI/CD pipeline with Monitoring Alerts and Patch Management**” by Vineet Patil(240344223025) 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 “**End to End Security Automation in CI/CD pipeline with Monitoring Alerts and Patch Management**” by **Shrikant Raut** (240344223030) 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** | **12** |
| **1.INTRODUCION** | **14** |
| **1.1 Applications** | **15** |
| **1.2 Organization and Project Plan** | **16** |
| **2. LITERATURE SURVEY** | **17** |
| **3. SYSTEM DEVELOPMENT AND DESIGN** | **20** |
| **3.1 Proposed System** | **20** |
| **3.2 Flow Chart** | **23** |
| **3.3 Technology used** | **23** |
| **AWS EC2** | **23** |
| **Jenkins** | **24** |
| **Git** | **24** |
| **SonarQube** | **24** |
| **Trivy File Scan & Image Scan** | **24** |
| **Docker Swarm** | **25** |
| **OSSEC** | **25** |
| **Puppet** | **25** |
| **4. PROJECT OUTPUT** | **26** |
| **5. 1 CONCLUSION** | **28** |
| **5.2 Future Scope** | **29** |
| **References** | **29** |

**ABSTRACT**

In today's dynamic business environment, automating the security and deployment of web applications is crucial to ensuring efficiency, consistency, and protection against vulnerabilities. This project focuses on creating an automated pipeline that integrates Jenkins, Git, SonarQube, Aqua Trivy, Docker Swarm, OSSEC, and Puppet to streamline the deployment process while fortifying the application's security posture.

The solution begins with Jenkins, which automates the retrieval of source code and Docker images from Git repositories, triggering a build process whenever changes are detected. This ensures continuous integration and keeps the development pipeline active and responsive.

Security checks are paramount in this workflow. Dependency Check analyzes project dependencies to identify and mitigate security vulnerabilities. Following this, SonarQube performs a comprehensive static code analysis, detecting bugs, code smells, and potential vulnerabilities. Aqua Trivy further secures the deployment by scanning Docker images for any hidden risks, ensuring that only safe and compliant images proceed to the next stage.

Once the code and dependencies pass all security checks, Docker Swarm is employed to deploy the web application. Docker Swarm’s orchestration capabilities ensure that the application is hosted with load balancing and high availability, making it resilient to fluctuations in demand.

Post-deployment, OSSEC monitors the application for suspicious activities or attacks, providing real-time alerts to maintain the security integrity of the deployed application. This proactive approach helps in early detection and mitigation of potential threats.

Finally, Puppet is utilized to maintain the system's health by automating the application of security patches and configuration updates. This ensures that the infrastructure remains secure and up-to-date without manual intervention.

In conclusion, this project provides a comprehensive automated solution for the deployment and security management of web applications. By integrating these powerful tools, the solution not only enhances operational efficiency but also significantly strengthens the security framework, making it an essential asset for modern web application management.

1. **INTRODUCTION**

In the rapidly evolving landscape of web development, ensuring the security and efficiency of deployment processes is essential. This project is designed to address these needs by automating the security and deployment workflow for a web application. Leveraging a combination of industry-leading tools—Jenkins, Git, SonarQube, Aqua Trivy, Docker Swarm, OSSEC, and Puppet—this project integrates these technologies to create a streamlined, robust, and secure deployment pipeline.

The workflow begins with Jenkins, an automation server that plays a central role in the project by automatically pulling the latest source code or Docker images from Git repositories. This setup triggers the build process whenever changes are detected, ensuring continuous integration and prompt deployment.

Security is a cornerstone of this project, with multiple layers of checks integrated into the pipeline. Dependency Check is used to scrutinize project dependencies for vulnerabilities, ensuring they are secure and up-to-date. SonarQube follows with a detailed static code analysis, identifying potential bugs, vulnerabilities, and code smells. Aqua Trivy further strengthens security by scanning Docker images for vulnerabilities after dependency checks and code analysis have been completed.

Once all security checks are passed, Docker Swarm is responsible for deploying the web application. Docker Swarm ensures that the application is hosted with high availability and load balancing, making it resilient to demand fluctuations.

The security posture of the deployed application is maintained through continuous monitoring by OSSEC, a host-based intrusion detection system. OSSEC is configured to monitor system logs and other indicators of compromise, providing real-time alerts via email in the event of any suspicious activity.

Finally, the project includes the use of Puppet, a configuration management tool that automates the application of security patches and system configurations. This ensures that all systems within the deployment environment remain secure and up-to-date, without the need for manual intervention.

Through the integration of these tools and processes, this project not only automates the deployment and security of a web application but also ensures a consistent and secure operational environment.

1.1 Application:

* **Automated Security and Deployment:**
* The project automates the entire security and deployment process for a web application, minimizing human intervention and ensuring a seamless, secure workflow.
* **Continuous Integration and Deployment:**

Utilizes Jenkins to automate the continuous integration and deployment pipeline. It automatically pulls the latest code from Git repositories, triggers builds, and ensures that any changes are tested and deployed swiftly.

* **Dependency and Code Quality Assurance:**

Incorporates tools like Dependency Check and SonarQube to ensure that all dependencies are secure, up-to-date, and that the codebase is free of vulnerabilities, bugs, and code smells.

* **Vulnerability Scanning:**

Aqua Trivy scans Docker images and files to detect vulnerabilities before the application is deployed, ensuring that only secure images are used in production.

* **Scalable and Resilient Hosting:**

Docker Swarm orchestrates the hosting environment, providing load balancing and high availability. This ensures that the application can handle varying levels of traffic without downtime.

* **Intrusion Detection and Monitoring:**

OSSEC monitors the deployed application, providing real-time alerts for any suspicious activity or potential security breaches. This enhances the security posture by detecting and responding to threats promptly.

* **Automated Maintenance and Patching:**

Puppet manages the configuration of the system, ensuring that all components are consistently up-to-date with the latest security patches, reducing the risk of vulnerabilities being exploited.

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 |  |  |  |  |

1. **LITERATURE SURVEY**

**1. Automation in DevOps: Enhancing Efficiency and Reliability**

**Key Concepts:** The adoption of automation in DevOps processes is crucial for improving efficiency, reducing human error, and ensuring consistent deployment and security practices. Jenkins, as an automation server, plays a vital role in streamlining CI/CD pipelines by automating tasks such as code integration, testing, and deployment. This automation is further extended with tools like Git for version control, SonarQube for static code analysis, Aqua Trivy for security scanning, Docker Swarm for orchestration, and Puppet for configuration management.

**Relevant Studies:** Explore studies and articles that discuss the impact of automation in DevOps, including case studies that highlight the benefits of using Jenkins and other automation tools in reducing deployment times, minimizing errors, and enhancing security practices.

**2. Security in Continuous Integration and Continuous Deployment (CI/CD)**

**Key Concepts:** Integrating security checks within the CI/CD pipeline is critical to ensuring that applications are secure before they are deployed. Dependency Check, SonarQube, and Aqua Trivy are integral to this process, providing automated security analysis at various stages of development. Dependency Check focuses on identifying vulnerabilities in project dependencies, while SonarQube and Aqua Trivy provide code and container image analysis, respectively.

**Relevant Studies:** Review literature on the importance of embedding security in the CI/CD pipeline, with a focus on the effectiveness of tools like SonarQube and Aqua Trivy in identifying and mitigating security risks during the development process.

**3. Container Orchestration and Deployment with Docker Swarm**

**Key Concepts:** Docker Swarm provides container orchestration, enabling the deployment of applications in a distributed environment with load balancing and high availability. The integration of Docker Swarm within a CI/CD pipeline allows for seamless deployment of applications that have passed security checks, ensuring that they are not only scalable but also secure.

**Relevant Studies:** Analyze studies that discuss the use of Docker Swarm in production environments, particularly its role in automating deployment, managing containers at scale, and ensuring application availability and reliability.

**4. Monitoring and Intrusion Detection in Automated Deployments**

**Key Concepts:** Continuous monitoring and intrusion detection are essential components of a secure deployment process. OSSEC, as a host-based intrusion detection system (HIDS), provides real-time monitoring of deployed applications, alerting administrators to potential security threats. Its integration within an automated deployment pipeline ensures that applications are continuously monitored for security breaches.

**Relevant Studies:** Investigate the role of HIDS in modern DevOps practices, focusing on the effectiveness of OSSEC in detecting and responding to security incidents in real-time within automated environments.

**5. Configuration Management and Patch Automation**

**Key Concepts:** Configuration management tools like Puppet are essential for maintaining the security and consistency of deployed applications. Puppet automates the process of applying security patches and managing system configurations, ensuring that all components within the deployment pipeline are up-to-date and secure.

**Relevant Studies:** Explore the literature on the role of configuration management in DevOps, with a focus on how automation tools like Puppet contribute to maintaining secure and consistent environments.

**6. Case Studies and Industry Implementations**

**Key Concepts:** Real-world case studies provide valuable insights into the implementation of automated security and deployment processes. These case studies often highlight the challenges, solutions, and outcomes of adopting a fully automated CI/CD pipeline in various industries.

**Relevant Studies:** Review case studies from different industries that have successfully implemented automation in their deployment pipelines, particularly those that have used Jenkins, Git, SonarQube, Aqua Trivy, Docker Swarm, OSSEC, and Puppet to enhance security and efficiency.

**7. Emerging Trends in DevSecOps**

**Key Concepts:** The integration of security within DevOps practices, known as DevSecOps, is an emerging trend that emphasizes the importance of security automation in the development and deployment process. This trend is driven by the need for continuous security in an increasingly automated and agile development environment.

**Relevant Studies:** Investigate emerging trends in DevSecOps, including the use of AI and machine learning for predictive security, the integration of cloud-native security tools, and the adoption of automated compliance checks within CI/CD pipelines.

# 3. System Development and Design

* 1. **Proposed System:**
* The proposed system aims to automate the security and deployment process for a web application by integrating several tools and technologies. This system is designed to enhance efficiency, security, and scalability, ensuring that the web application is securely deployed and maintained with minimal manual intervention. Below is a detailed outline of the proposed system:

**System Architecture:**

* **Primary Components:**
* **AWS EC2 Instances (Compute Infrastructure):**

**Function:** AWS EC2 provides scalable and flexible cloud-based

compute resources for hosting and running all components of the

CI/CD pipeline

**Workflow:** All primary components of the architecture, such as

Jenkins, SonarQube, Docker Swarm, and others, are hosted on

AWS EC2 instances. EC2 instances are provisioned to run these

services, with the ability to scale up or down based on the

pipeline’s requirements, ensuring high availability, performance

and security of the CI/CD process

* **Jenkins (Automation Server):**

**Function:** Jenkins serves as the central automation server, responsible for orchestrating the entire CI/CD pipeline.

**Workflow:** Jenkins automatically pulls source code or Docker images from Git repositories. It triggers the build and deployment process whenever changes are detected in the repository, ensuring that the application is continuously updated with the latest code

* **Git (Source Code Management):**

**Function:** Git is used for version control, managing the source code, and tracking changes.

* + **Workflow:** Jenkins is configured to monitor Git repositories. Upon detecting changes, Jenkins triggers a build process, integrating the latest code into the deployment pipeline.
* **Dependency Check (Security Analysis):**

**Function:** Dependency Check analyzes project dependencies for security vulnerabilities.

**Workflow:** After pulling the latest code, Jenkins runs Dependency Check to ensure that all dependencies are secure and up-to-date, identifying any potential security risks.

* **SonarQube (Static Code Analysis):**

**Function:** SonarQube performs static code analysis to identify bugs, vulnerabilities, and code smells.

**Workflow:** Following the dependency analysis, SonarQube runs a comprehensive analysis of the codebase, generating a detailed report on the code’s quality and security.

* **Aqua Trivy (Docker Image and File Scanning):**

**Function:** Trivy scans Docker images and files for vulnerabilities.

**Workflow:** After SonarQube analysis, Trivy is triggered to scan the Docker images, ensuring that the containers used for deployment are free of security vulnerabilities.

* **Docker Swarm (Container Orchestration):**

**Function:** Docker Swarm is used for container orchestration, managing the deployment of the web application across multiple nodes.

**Workflow:** Once all security checks are passed, Docker Swarm deploys the web application, providing load balancing and ensuring high availability of the application.

* **OSSEC (Host-Based Intrusion Detection System):**

**Function:** OSSEC monitors the deployed application for security breaches and unauthorized activities.

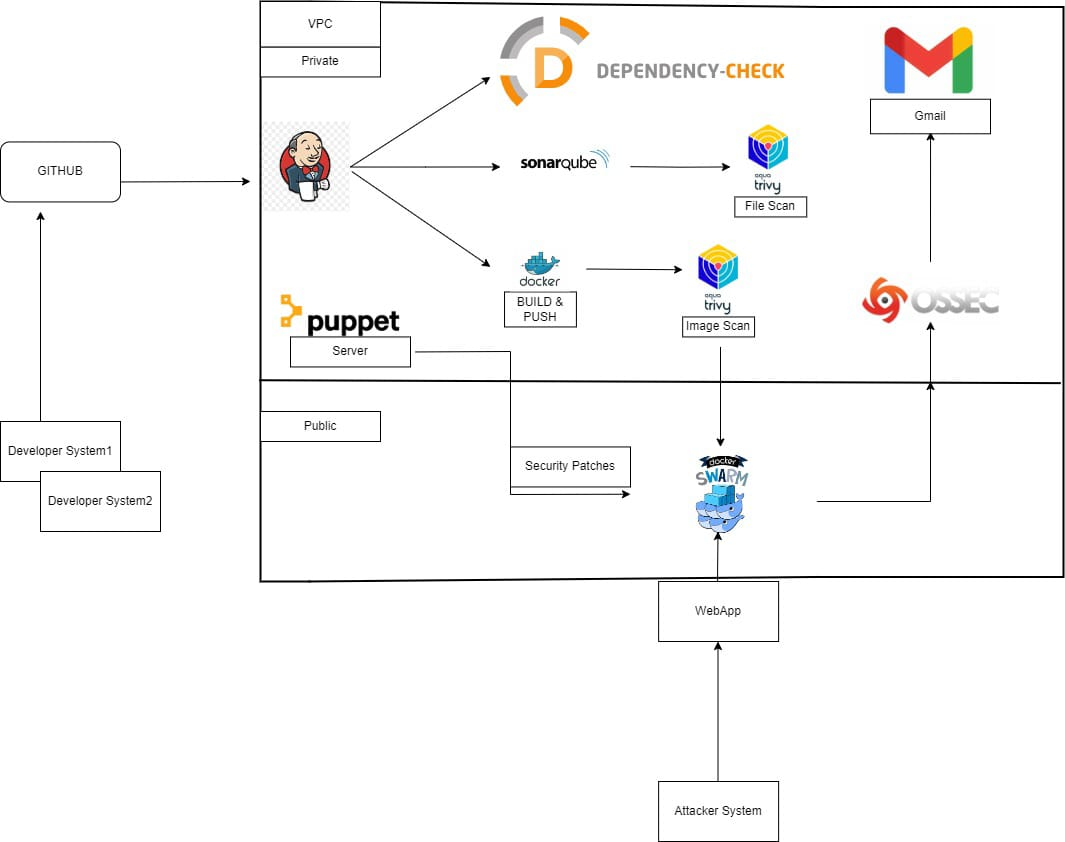
**Workflow:** OSSEC is configured to continuously monitor system logs and other indicators of compromise. It sends real-time email alerts to the IT team if any suspicious activity is detected.

* **Puppet (Configuration Management):**

**Function:** Puppet automates the application of security patches and maintains system configurations.

**Workflow:** Puppet is used to ensure that all systems involved in the deployment process are consistently updated with the latest security patches and configurations, minimizing vulnerabilities.

* 1. **Flow chart**



* 1. **Technology used:**

1. Amazon EC2: Cloud Computing Service

* **Key Features**:
  + **Scalable Compute Capacity**: Amazon EC2 provides resizable compute capacity in the cloud, allowing you to quickly scale your application up or down as needed.
  + **Customizable Instances**: Choose from a wide variety of instance types optimized for different workloads, including compute-intensive,

memory-intensive, and storage-optimized applications.

* + **Elastic Load Balancing**: Distribute incoming application traffic across multiple EC2 instances, ensuring high availability and reliability.
  + **Security and Compliance**: EC2 offers robust security features, including VPC (Virtual Private Cloud) for network isolation, IAM roles for
  + **Flexible Pricing Options**: Pay for only the compute capacity you actually use, with options like On-Demand, Reserved, and Spot Instances to optimize

costs based on your workload.

**2. Jenkins: Automation Server**

* **Key Features**:
  + **Continuous Integration/Continuous Deployment (CI/CD)**: Jenkins automates the process of building, testing, and deploying applications, ensuring rapid and reliable delivery.
  + **Plugin Ecosystem**: Supports a vast array of plugins to extend its functionality, allowing seamless integration with various tools like Git, SonarQube, Docker, etc.
  + **Pipeline as Code**: Jenkins enables you to define your build and deployment pipelines as code, making them version-controlled and easy to manage.
  + **Cross-Platform**: Runs on various operating systems, including Windows, Linux, and macOS.

**3. Git: Distributed Version Control System**

* **Key Features**:
  + **Branching and Merging**: Facilitates easy branching for new features or bug fixes and smooth merging back into the main codebase.
  + **Distributed Architecture**: Allows every developer to have a full copy of the repository, ensuring redundancy and enabling offline work.
  + **Version Control**: Tracks changes to the codebase, making it easy to revert to previous versions or compare changes.
  + **Collaboration Tools**: Supports workflows like pull requests and code reviews, promoting collaboration and code quality.

**4. SonarQube: Static Code Analysis Tool**

* **Key Features**:
  + **Code Quality and Security**: Analyzes source code for potential bugs, security vulnerabilities, and code smells.
  + **Multi-Language Support**: Supports a wide range of programming languages, making it versatile for different projects.
  + **Continuous Inspection**: Integrates with CI/CD pipelines to provide real-time feedback on code quality during the development process.
  + **Customizable Rules**: Allows customization of rules and quality gates to meet specific project requirements.

**3. Aqua Trivy: Vulnerability Scanner for Docker Images**

* **Key Features**:
  + **Comprehensive Scanning**: Scans Docker images, filesystems, and Git repositories for known vulnerabilities.
  + **Integration with CI/CD**: Easily integrates with CI/CD pipelines to ensure that only secure images are deployed.
  + **Open Source**: Trivy is an open-source tool, providing robust security scanning without licensing costs.
  + **Lightweight and Fast**: Designed to be lightweight, with quick scanning capabilities, making it ideal for integration into automated processes.

**5. Docker Swarm: Container Orchestration Tool**

* **Key Features**:
  + **High Availability**: Provides fault tolerance and high availability for containerized applications by distributing them across multiple nodes.
  + **Load Balancing**: Automatically balances load across containers to ensure optimal resource utilization and application performance.
  + **Easy Setup**: Simple to set up and manage, making it suitable for projects that require a straightforward orchestration solution.
  + **Integrated with Docker**: As part of the Docker ecosystem, Docker Swarm integrates seamlessly with Docker CLI and other Docker tools.

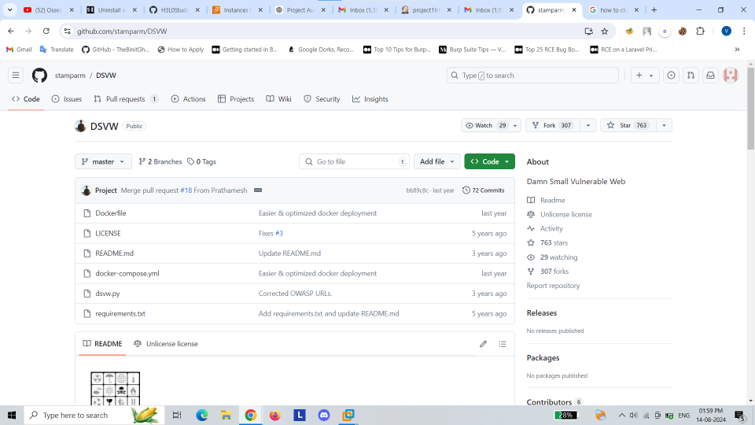
**6 . OSSEC: Host-Based Intrusion Detection System (HIDS)**

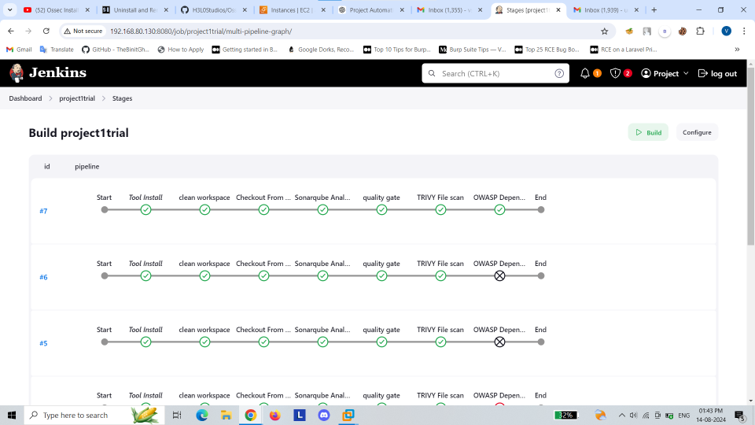
* **Key Features**:
  + **Real-Time Monitoring**: Monitors system logs and files for suspicious activity in real time.
  + **Alerting Mechanism**: Sends immediate alerts through email or other configured channels upon detecting potential security breaches.
  + **Customizable Rules**: Allows customization of detection rules to fit specific security requirements.
  + **Multi-Platform Support**: Works on multiple operating systems, making it versatile for different environments.

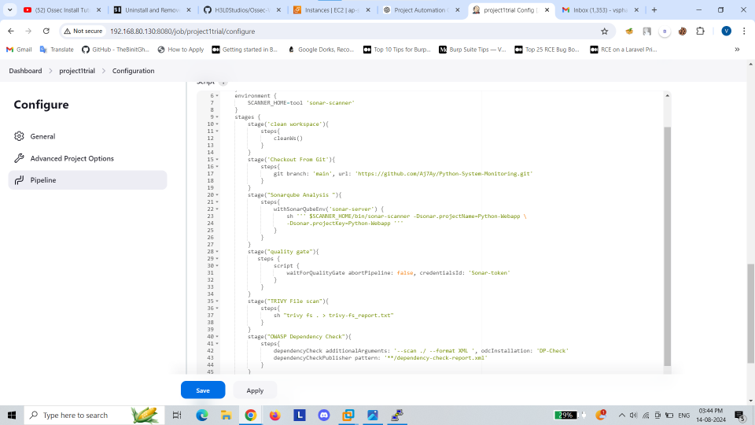
**7. Puppet: Configuration Management Tool**

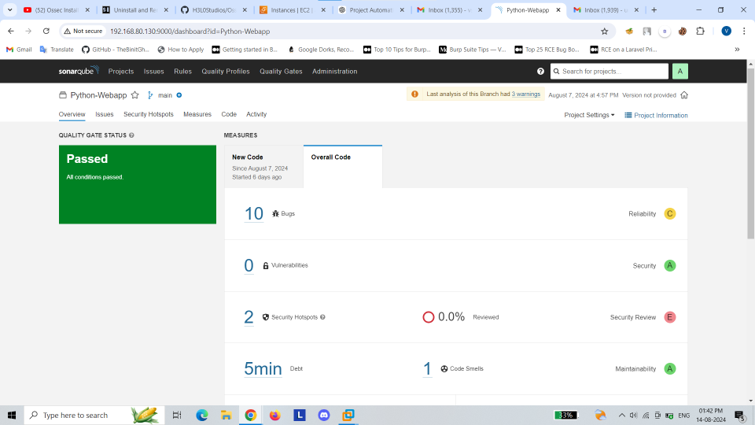
* **Key Features**:
  + **Automated Configuration**: Manages and automates the configuration of systems, ensuring consistency across environments.
  + **Infrastructure as Code**: Defines system configurations as code, enabling version control and repeatability.
  + **Scalability**: Easily scales to manage thousands of nodes, making it suitable for large environments.
  + **Compliance Enforcement**: Ensures that systems adhere to security policies by automatically applying patches and updates.

1. **Project Output**







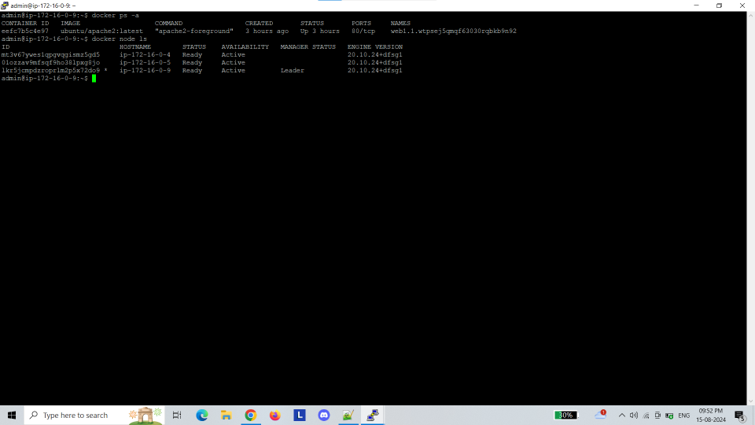


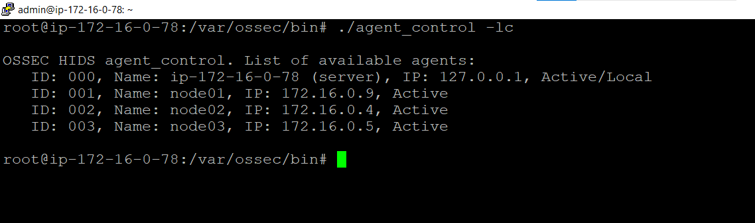
A screenshot of a computer

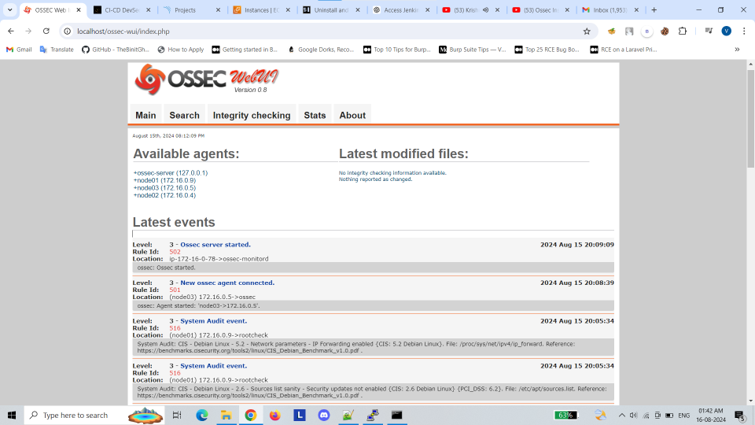
Description automatically generated

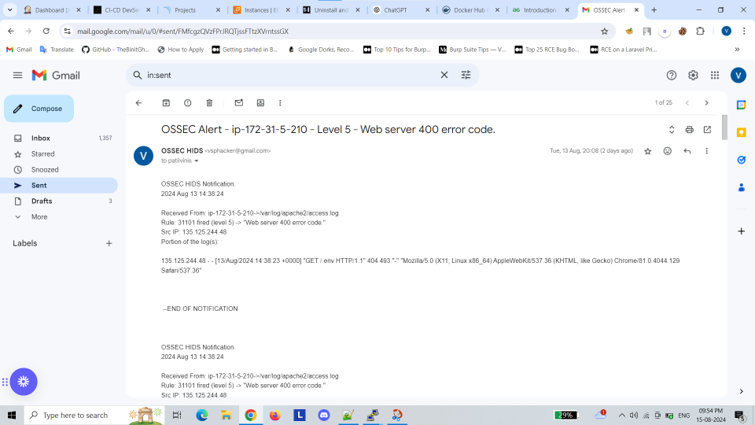
A screenshot of a computer

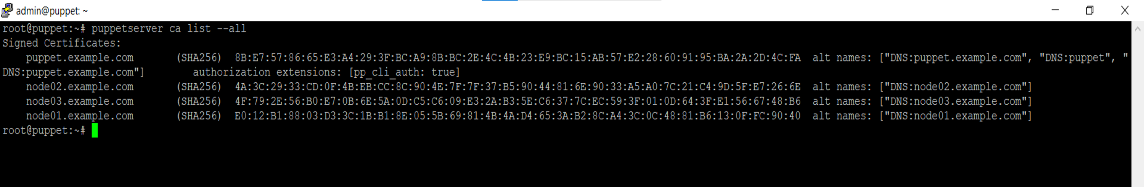
Description automatically generated

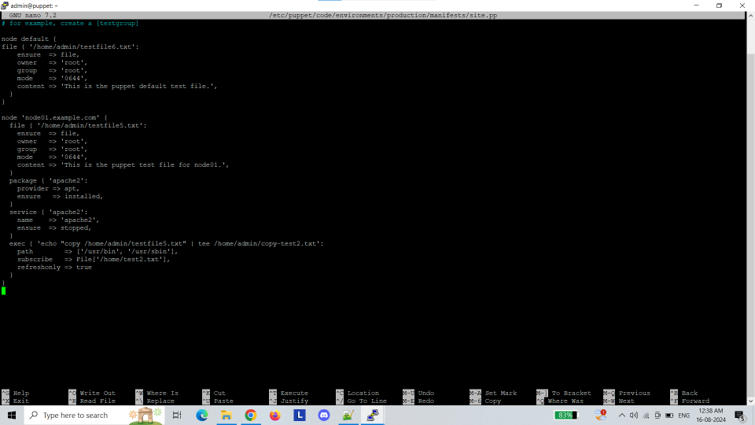












**5. CONCLUSION**

**5.1 Conclusion:**

This project successfully demonstrates the automation of security and deployment processes for web applications by leveraging an integrated stack of tools including Jenkins, Git, SonarQube, Aqua Trivy, Docker Swarm, OSSEC, and Puppet. The automation framework ensures consistent and efficient deployment, enhances security by identifying and addressing vulnerabilities early in the development process, and maintains the integrity of the deployed application through continuous monitoring and timely patching. By combining these technologies, the project reduces manual intervention, mitigates risks associated with security breaches, and ensures the seamless operation of the web application in a production environment.

**5.2 Future Scope:**

The future scope of this project lies in enhancing the automation capabilities and expanding the security measures to adapt to evolving threats and technological advancements. Potential areas for future work include integrating advanced threat detection tools with machine learning capabilities to anticipate and respond to new security challenges proactively. Additionally, the project can be expanded to support multi-cloud environments, enabling more flexible and scalable deployments. Enhancements in container orchestration with Kubernetes, instead of Docker Swarm, could further optimize resource management and application scaling. Continuous improvement in configuration management with more sophisticated automation tools will also contribute to maintaining robust security postures in increasingly complex and dynamic application environments.

**REFERENCES**

https://github.com/aquasecurity/trivy

<https://www.youtube.com/>

<https://stackoverflow.com/>