

Project Report on

**AWS Web Server Deployment with CI/CD, Prometheus Monitoring, and IPS Security using Snort**

**Submitted by**

**Shradha Gaikwad (240844223014)**

**Dhanshri Rokde (240844223010)**

**Sneha Kalsait (240844223054)**

**Prathamesh Panjabi (240844223033)**

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:

**Shradha Gaikwad Dhanshri Rokde Sneha Kalsait Prathamesh Panjabi**

**(240844223014) (240844223010) (240844223054) (240844223033)**

**CERTIFICATE**

This is to certify that the project report entitled **“AWS Web Server Deployment with CI/CD, Prometheus Monitoring, and IPS Security using Snort”**, submitted by **Shradha Gaikwad** 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 **“AWS Web Server Deployment with CI/CD, Prometheus Monitoring, and IPS Security using Snort”**, submitted by **Dhanshri Rokde** 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 **“AWS Web Server Deployment with CI/CD, Prometheus Monitoring, and IPS Security using Snort”**, submitted by **Sneha Kalsait** 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 **“AWS Web Server Deployment with CI/CD, Prometheus Monitoring, and IPS Security using Snort”**, submitted by **Prathamesh Panjabi** 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 **“AWS Web Server Deployment with CI/CD, Prometheus Monitoring, and IPS Security using Snort”** by **Shradha Gaikwad (240844223014)** 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 **“AWS Web Server Deployment with CI/CD, Prometheus Monitoring, and IPS Security using Snort”** by **Dhanshri Rokde (240844223010)** 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 **“AWS Web Server Deployment with CI/CD, Prometheus Monitoring, and IPS Security using Snort”** by **Sneha Kalsait (240844223054)** 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 **“AWS Web Server Deployment with CI/CD, Prometheus Monitoring, and IPS Security using Snort”** by **Prathamesh Panjabi (240844223033)** 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. **INTRODUCTION** | 1 |
| 1.1 Application | 1 |
| 1.2 Organization and Project Plan | 4 |
| 1. **LITERATURE SURVEY** | 5 |
| Paper 1 | 5 |
| Paper 2 | 5 |
| Paper 3 | 6 |
| **3. SYSTEM DEVELOPMENT AND DESIGN** | 7 |
| 3.1 Proposed System | 7 |
| 3.2 Flow Chart | 7 |
| 3.3 Technology used | 8 |
| 3.3.1 AWS EC2 | 8 |
| 3.3.2 Jenkins | 9 |
| 3.3.3 Git | 10 |
| 3.3.4 Docker | 11 |
| 3.3.5 SonarQube | 12 |
| 3.3.6 Snort | 13 |
| 3.3.7 Prometheus | 14 |
| 3.3.8 Grafana | 15 |
| **4. PROJECT OUTPUT** | 16 |
| **5. CONCLUSION** | 20 |
| 5.1 Conclusion | 20 |
| 5.2 Future Scope | 20 |
| **REFERENCES** | 21 |

**ABSTRACT**

The rapid evolution of web applications has created a demand for automated deployment systems that are secure, efficient, and capable of monitoring performance in real-time. This project addresses these needs by designing and implementing a robust CI/CD pipeline integrated with intrusion prevention and system monitoring capabilities. The deployment process is distributed across multiple virtual machines (VMs). The source code repository with Git, along with the application code, Dockerfile, and Jenkinsfile, is hosted on the base machine, from which the developer will push the code. The first VM runs Jenkins to automate builds, integrate SonarQube for static code analysis, and use Docker to create container images, which are pushed to Docker Hub.

The second VM is an Amazon Web Services (AWS) Elastic Compute Cloud (EC2) instance where the application is deployed. This VM is secured using Snort, configured as an Intrusion Prevention System (IPS), to monitor and block suspicious network activities, ensuring robust application security. The third VM hosts Prometheus, which is integrated with other VM which is having EC2 instance to monitor system performance, container health, and resource utilization. Prometheus provides real-time metrics and alerts, allowing administrators to proactively address performance bottlenecks and potential issues, while Grafana is integrated for intuitive data visualization.

The project demonstrates how integrating modern DevOps practices, such as CI/CD automation, with security measures like Snort IPS and monitoring tools like Prometheus, can enhance the reliability, scalability, and security of web application deployments. This scalable and efficient approach provides a solution for dynamic application environments while addressing key challenges in deployment automation and system protection.

i

1. **INTRODUCTION**

The rapid growth of web applications and cloud-based infrastructure has transformed the way businesses deploy and manage their services. With increasing demands for faster release cycles, improved security, and higher system reliability, it has become essential for developers and IT operations teams to adopt automation and robust monitoring practices. Traditional deployment methods can often be slow and error-prone, which impacts both the speed of updates and the security of applications. As businesses scale and the complexity of their applications grows, ensuring that deployments are secure, efficient, and continuously monitored is crucial.

This project seeks to address these challenges by implementing a fully automated web application deployment solution using modern DevOps practices. It combines Amazon Web Services (AWS) EC2 instances for scalable infrastructure, Jenkins for continuous integration and delivery (CI/CD), Prometheus and Grafana for performance monitoring, and Snort for network intrusion prevention. The system is designed to provide efficient, secure, and continuously monitored web server deployment, ensuring seamless updates and robust protection against potential security threats.

The project employs AWS EC2 instances to provide a flexible and scalable environment for hosting the web applications, enabling quick adaptation to traffic demands. Jenkins automates the CI/CD pipeline, which reduces human intervention, accelerates development cycles, and ensures a consistent deployment process. Prometheus and Grafana are integrated for real-time monitoring and visualizing system metrics, allowing proactive identification and resolution of performance issues. Snort serves as an Intrusion Prevention System (IPS) that protects the deployment from potential security threats by monitoring network traffic and blocking malicious activities.

Through this comprehensive approach, the project ensures that the web application deployment is efficient, secure, and continuously monitored, providing organizations with the ability to quickly release updates while maintaining high security and system reliability.

Web Server Deployment using AWS EC2:

Amazon EC2 (Elastic Compute Cloud) provides scalable virtual servers for deploying web applications. In this project, an EC2 instance is configured to host a web server using NGINX, ensuring reliable web hosting with efficient load handling.

Continuous Integration and Delivery (CI/CD):

To automate the software deployment process, a CI/CD pipeline is established using Jenkins. The pipeline begins with developers pushing their code to a Git repository such as GitHub. Jenkins is configured to automatically pull the latest code, trigger a build process, and perform static code analysis using SonarQube. This ensures that any security vulnerabilities or code quality issues are detected before deployment.

Once the code passes quality checks, Jenkins deploys it to the AWS EC2 web server, updating the application without manual intervention. This automation minimizes downtime and ensures that new features, updates, and bug fixes are deployed rapidly and efficiently. The entire process helps in maintaining consistency, reliability, and faster delivery of web applications.

Prometheus Monitoring for Proactive Oversight:

Monitoring is essential to track the performance and health of the system. Prometheus is integrated into the setup to collect real-time metrics from the EC2 instance, including CPU utilization, memory usage, disk space, and network traffic. These metrics help administrators detect anomalies, prevent resource exhaustion, and ensure smooth operation of the web server.

To provide a user-friendly visualization of these metrics, Grafana is used to create dynamic dashboards. Grafana helps in identifying patterns and trends over time, allowing for proactive troubleshooting. Alerting mechanisms can also be configured in Prometheus, where administrators receive notifications if critical thresholds (such as high CPU load or disk space exhaustion) are reached. This setup ensures that potential issues are detected before they impact application performance, maintaining a high level of reliability and availability.

Enhancing Security with Snort IPS:

Security is a critical aspect of any web deployment. This project includes Snort, an open-source Intrusion Prevention System (IPS), to detect and mitigate cyber threats. Snort continuously monitors network traffic on the EC2 instance, analyzing packets for malicious activities such as brute force attacks, SQL injection attempts, denial-of-service (DoS) attacks, and unauthorized access attempts.

Snort operates using predefined and custom rules to identify and block suspicious activity before it affects the web server. If an attack pattern is detected, Snort can either log the event, generate alerts, or actively block the malicious traffic, adding an extra layer of security to the infrastructure.

By implementing Snort, the system is protected from common cyber threats and vulnerabilities, ensuring that the deployed web application remains secure and operational without being compromised.

**1.1 APPLICATIONS**

1. **Healthcare Applications**:

Healthcare organizations and platforms, such as electronic health record (EHR) systems or telemedicine platforms, require secure and highly available web applications. Regular updates, real-time performance monitoring, and robust security mechanisms ensure patient data is protected while providing seamless access.

2. **Education and E-Learning Platforms**:

Educational platforms like Coursera, Udemy, and Khan Academy need scalable infrastructure and continuous delivery of new features and updates. CI/CD pipelines and monitoring ensure uptime for students and educators while maintaining security for sensitive user data.

3. **Advertising and Analytics Platforms**:

Advertising platforms like Google Ads or analytics tools like Google Analytics need to be constantly updated to add new features, optimize performance, and handle large data volumes. CI/CD and real-time monitoring help maintain their services, ensuring minimal downtime and security breaches.

4. **Cybersecurity Firms**:

Cybersecurity companies like Palo Alto Networks, McAfee, and CrowdStrike use automated deployment and continuous integration pipelines to ensure their software solutions are up-to-date and secure. Real-time monitoring and intrusion detection prevent cyberattacks and protect sensitive data.

5. **Government Services**:

Public sector organizations, including government portals and e-Government services, rely on web application deployment pipelines to provide continuous access to public services. Security, uptime, and monitoring are critical for ensuring these platforms serve the public efficiently and securely.

**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 Design and Implement of IPS Based on Snort**

**Author:** [Jianrong Xi](https://ieeexplore.ieee.org/author/38236940000)

**Description:** With the development of application based on Internet, network security highlights its place increasing. Firewall and IDS are the equipment often used in Internet, but both of them can not run automatically. If we can reconfigure the firewall using the result of IDS, the security must be enhanced to a high level. In this paper, we designed an intrusion prevent system (IPS) based on Snort and Net filter by researching kernel codes of Snort and Net filter. The policy control module of the system was written in Multi-thread technologies. Meanwhile, the Algorithm of IDS and rule set of firewall was optimized to improve system efficiency. The system can block the attack source by dynamically modify firewall rules according to IDS.

**Paper 2: Implementing an Effective Infrastructure Monitoring Solution with Prometheus and Grafana**

**Author:** Pragathi B.C., Hrithik Maddirala, Sneha M.

**Description:** This paper investigates the implementation of a robust monitoring solution using Prometheus, Grafana, and Node Exporter in a Kubernetes environment. Motivated by the need for real-time insights and proactive management of Kubernetes clusters, the study delves into the cause of infrastructure monitoring challenges and explores the methodology employed to address them. Through a meticulous deployment process, Prometheus, functioning as the central monitoring component, adopts a pull-based approach to collect metrics from Kubernetes nodes and pods. Meanwhile, Grafana complements Prometheus by offering powerful visualization capabilities, facilitating the creation of dynamic dashboards for monitoring system performance and resource utilization. Node Exporter further enhances the monitoring system by providing detailed system metrics, including CPU usage, memory utilization, and disk I/O, at the node level. The integration of these tools enables organizations to gain comprehensive insights into their Kubernetes infrastructure, facilitating timely anomaly detection, efficient resource allocation, and proactive management. The study presents compelling results demonstrating the effectiveness of the monitoring solution in improving infrastructure visibility, enhancing operational efficiency, and ensuring the reliability of Kubernetes deployments.

**Paper 3: The Proposed Pre-Configured Deployment Model for Amazon EC2 Cloud Services**

**Author**: [Anurag Choudhary](https://ieeexplore.ieee.org/author/37089688389), [Pradeep Kumar Verma](https://ieeexplore.ieee.org/author/37088663819), [Piyush Rai](https://ieeexplore.ieee.org/author/37089885189)

**Description:** Small, medium, and big organizations get several advantages from cloud computing, but it also presents obstacles. Whether a firm is in the financial, technology, or engineering sector, a cloud component might be beneficial. Though there are numerous obstacles associated with cloud computing, experts think that the benefits outweigh the drawbacks. The issues will be addressed when more research in the field of cloud computing is conducted. Cloud services are provided by a number of significant companies, including Amazon Web Services, Microsoft Azure, and Google Cloud Platform, among others. Among them, AWS (Amazon Web Services) is one of the fine cloud service providers that comprises several features, including the AWS EC2 (Elastic Compute Cloud) which is one of the widely used by many organizations. Amazon's Elastic Compute Cloud Web service delivers highly adjustable processing capacity throughout the cloud, allowing developers to construct applications with incredible scalability. Using EC2 (Elastic Compute Cloud) by using the proposed deployment method can be more effort saver for any IT development and deployment team for any organization. There should be an easy deployment method that auto-configures EC2 Instance. The aim of this research paper is to showcase the current deployment and service models provided by Amazon Web Services EC2 and present the proposed solution in order to the existing scenario. Furthermore, its advantages are also present so that it becomes easier to select the most appropriate one for deployment and research development.

**3. SYSTEM DEVELOPMENT AND DESIGN**

**3.1 Proposed System**

The proposed system automates the deployment, monitoring, and security of web applications using AWS EC2, Jenkins, Docker, SonarQube, Prometheus, Grafana, and Snort. AWS EC2 instances provide scalable infrastructure for hosting web servers, while Jenkins manages the CI/CD pipeline, automating code testing, building, and deployment. SonarQube is integrated into the pipeline to ensure code quality by performing static code analysis. Prometheus and Grafana are used for real-time monitoring, offering insights into system performance through customized dashboards. Snort, an Intrusion Prevention System (IPS), provides security by detecting and alerting on potential threats. This system ensures efficient, secure, and continuously monitored web application deployment and management.

**3.2 Flow Chart**

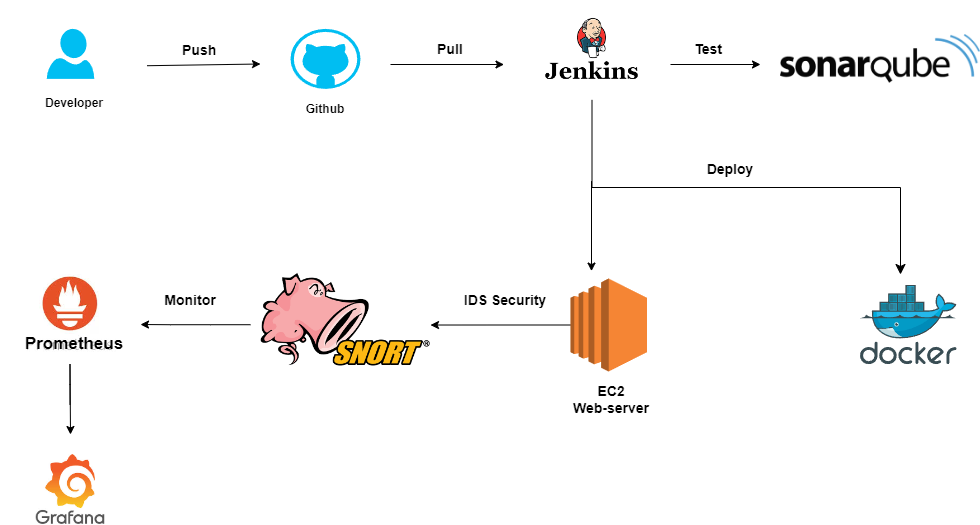


Figure: Flow Chart

**3.3 Technology used**

**3.3.1 Amazon Web Services (AWS) EC2**

AWS EC2 (Elastic Compute Cloud) is a web service that provides resizable compute capacity in the cloud. It is primarily used to host and run applications in virtual machines (called instances). EC2 enables scalable cloud computing, allowing users to provision instances based on their computing needs.

**Key Features:**

* Scalable Infrastructure: EC2 enables you to scale up or down based on demand. You can add or remove instances depending on the size of the workload, which makes it ideal for variable traffic demands.
* Multiple Instance Types: EC2 offers different instance families optimized for compute, memory, storage, and GPU workloads, giving users the flexibility to choose the right instance type. Auto Scaling: Automatically adjusts the number of instances based on traffic demands.
* Load Balancing: Distributes incoming traffic across multiple instances for high availability and fault tolerance.
* Auto Scaling: Automatically adjusts the number of instances to handle changes in traffic. This feature helps ensure that the application has enough capacity during traffic spikes and reduces costs when traffic is low.
* Security and Compliance: EC2 integrates with AWS Identity and Access Management (IAM) to provide fine-grained access control. EC2 also supports Virtual Private Cloud (VPC), security groups, and network ACLs for secure network configurations.

**3.3.2. Jenkins**

Jenkins is an open-source automation server used to implement Continuous Integration (CI) and Continuous Delivery/Deployment (CD) pipelines. It automates various stages of software development, such as code integration, testing, and deployment, helping teams deliver high-quality software more efficiently.

**Key Features:**

* CI/CD Pipelines: Jenkins allows for the automated creation and management of CI/CD pipelines, which automate processes like code compilation, testing, and deployment. This results in faster and more reliable software releases.
* Extensive Reporting and Notifications**:** Jenkins provides real-time feedback and notifications for build statuses and test results, helping teams to quickly identify issues in the development process.
* Pipeline as Code**:** Defines the build and deployment pipelines in a version-controlled Jenkinsfile.
* Distributed Builds: Jenkins supports distributed builds where multiple nodes or machines can be used to perform the tasks. This optimizes the build process by parallelizing tasks across different resources.
* Declarative Pipelines: Jenkins supports the use of pipeline-as-code using a domain-specific language (DSL) called the Jenkinsfile, which enables developers to version control their build and deployment pipelines.
* Automation of Repetitive Tasks: Reduces human error and manual interventions by automating repetitive tasks.

**3.3.3** **GitHub**

GitHub is a cloud-based version control and collaboration platform that allows developers to store, manage, and share code. It uses Git for version control and offers collaborative features for development teams, such as pull requests and code reviews.

**Key Features:**

* Version Control: GitHub uses Git for version control, which allows developers to track and manage changes to the codebase. Developers can create branches for different features or fixes and later merge them into the main codebase after testing.
* Branching & Merging: Supports the creation of feature branches and merging code changes seamlessly.
* Pull Requests**:** GitHub enables collaboration via pull requests, where team members can review code changes before merging them into the main repository. This fosters code quality and peer review practices.Integrated Issue Tracking: Provides a platform for managing bugs, feature requests, and other project tasks.
* Security Features: GitHub provides features such as Dependabot to automatically scan and update dependencies for security vulnerabilities, ensuring the integrity of your codebase.
* Collaboration Tools: GitHub includes issue tracking, project boards, wikis, and other tools to help teams collaborate on projects. It helps track progress and manage feature requests and bug reports.
* GitHub Actions: GitHub Actions is a feature that enables the automation of workflows within the repository. This allows for continuous integration, testing, and deployment directly within GitHub.

**3.3.4 Docker**

Docker is an open-source platform that enables developers to build, deploy, and manage applications inside lightweight, portable containers. Containers package the application and its dependencies, ensuring that the application runs consistently across different environments.

**Key Features:**

* Containerization: Docker isolates applications and their dependencies in containers, ensuring that they run consistently on any environment (developer machines, testing environments, production servers).
* Portability: Docker containers can be moved seamlessly across environments (e.g., from a developer's laptop to a test server or from a local server to the cloud), providing flexibility in deployment.
* Efficiency: Unlike virtual machines, containers share the host system’s kernel, which makes them more lightweight and faster to start. This leads to better resource utilization and quicker deployments.
* Simplified Deployment: Allows applications to be deployed easily and quickly across different platforms.
* Docker Hub: Docker Hub is a cloud-based registry that allows users to share container images. It includes both public and private repositories, making it easy to store and access prebuilt images.
* Docker Compose: Docker Compose allows developers to define multi-container applications using a simple YAML file. This tool enables the creation of complex setups with linked containers.

**3.3.5 SonarQube**

SonarQube is an open-source platform for continuous inspection of code quality. It performs static code analysis to identify bugs, vulnerabilities, and code smells in the codebase. SonarQube integrates with CI/CD pipelines to automate code quality checks.

**Key Features:**

* Code Quality Analysis: SonarQube detects code issues such as bugs, vulnerabilities, and duplications, which helps improve code quality and maintainability.
* Code Coverage and Unit Testing**:** It measures unit test coverage, helping developers understand which parts of the code are covered by tests and which are not.
* Static Code Analysis: Analyzes code without executing it, ensuring comprehensive insights.
* Integration with CI/CD: Easily integrates with Jenkins or other CI tools to automate quality checks.
* Support for Multiple Languages: SonarQube supports numerous programming languages, including Java, C#, Python, JavaScript, and more, making it a versatile tool for diverse development teams.Customizable Rules: SonarQube provides customizable rules for code quality, enabling teams to define their own standards according to the project requirements.

**3.3.6 Snort (IDS/IPS):**

Snort is an open-source Intrusion Detection and Prevention System (IDS/IPS) that monitors network traffic to detect potential threats and intrusions. It analyzes packets for suspicious activity and generates alerts when attacks or anomalies are detected.

**Key Features:**

* Real-time Traffic Analysis Snort inspects network packets in real time, identifying potential threats and vulnerabilities as they occur.
* Intrusion Prevention: As an IPS, Snort can actively block malicious traffic based on predefined rules and patterns, stopping attacks before they compromise the system.
* Signature-based Detection: Snort uses signature-based detection, where it matches network traffic against known attack signatures to identify potential intrusions.
* Customizable Rules: Snort allows users to define custom detection rules to suit specific security needs, making it flexible for different environments.
* Logging and Alerting: Snort generates detailed logs and alerts for every potential security incident, providing insight into the nature of the attack and enabling quick response actions.

**3.3.7** **Prometheus**

Prometheus is an open-source monitoring and alerting toolkit designed for reliability and scalability. It collects and stores time-series data, primarily for monitoring cloud-native applications, infrastructure, and services.

**Key Features:**

* Time-Series Data Collection: Prometheus stores data as time-series, allowing users to track changes in metrics (e.g., CPU usage, memory, and application-specific metrics) over time.
* Powerful Query Language (PromQL): Prometheus uses PromQL (Prometheus Query Language) to query metrics. PromQL is flexible and allows users to aggregate, filter, and process metrics data efficiently.
* Alerting System: Alerts administrators based on thresholds and conditions defined by users.
* Built-In Storage: Prometheus includes its own storage layer, enabling long-term storage of monitoring data.
* Integration with Grafana: Prometheus integrates seamlessly with Grafana to provide powerful dashboards for visualizing collected metrics.
* Scalable: Designed for monitoring highly dynamic, cloud-native environments like Kubernetes.

**3.3.8 Grafana**

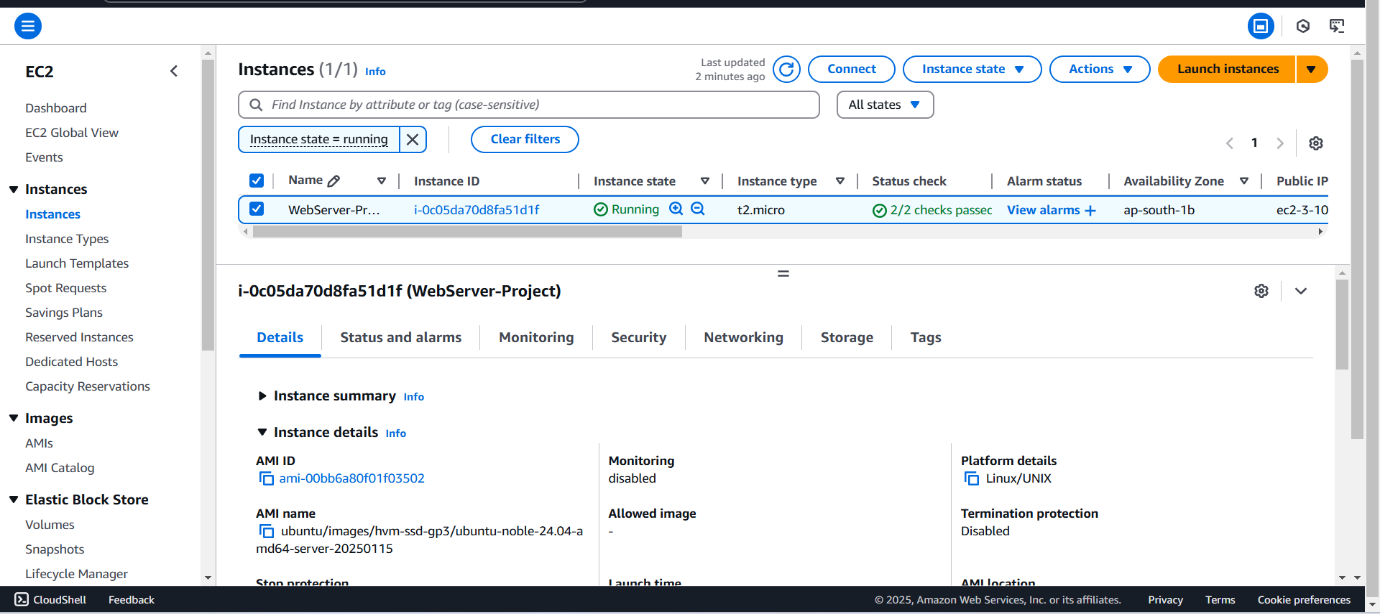
Grafana is an open-source data visualization and monitoring tool that integrates with Prometheus to create visual dashboards. It is widely used to visualize time-series data from various data sources, providing real-time insights into system and application performance.

**Key Features:**

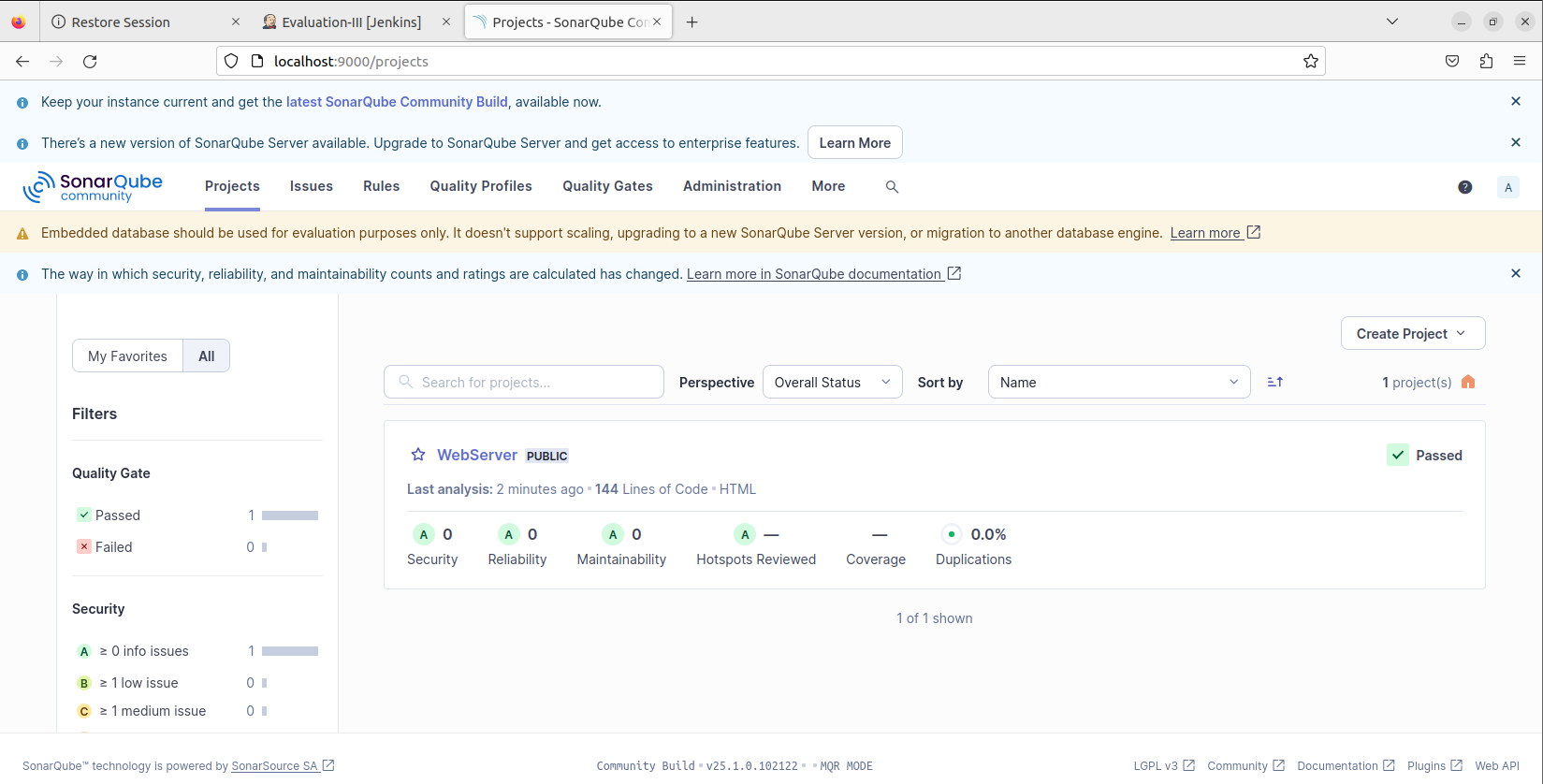
* Custom Dashboards: Grafana allows users to create personalized and interactive dashboards to visualize different types of data in real-time.
* Integration with Multiple Data Sources**:** It supports multiple data sources like Prometheus, Elasticsearch, MySQL, and more, enabling you to visualize metrics from a variety of systems.
* Alerting and Notifications: Grafana can trigger alerts based on specific conditions and notify users via email, Slack, or other channels.
* User-Friendly Interface: Offers an intuitive, easy-to-use interface for creating and managing visualizations.
* Annotations: Allows adding notes or markers to visualizations for contextual information.

**4. Project Output**

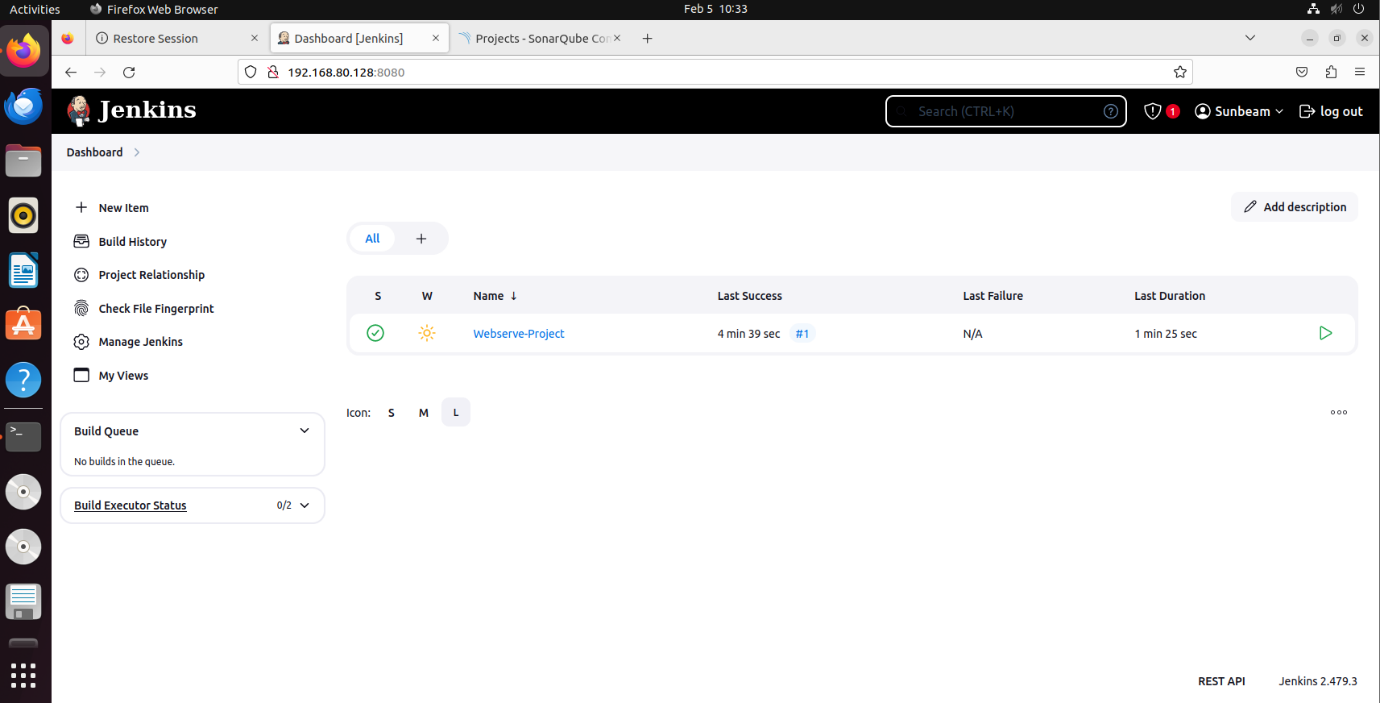
* 1. **AWS EC2**



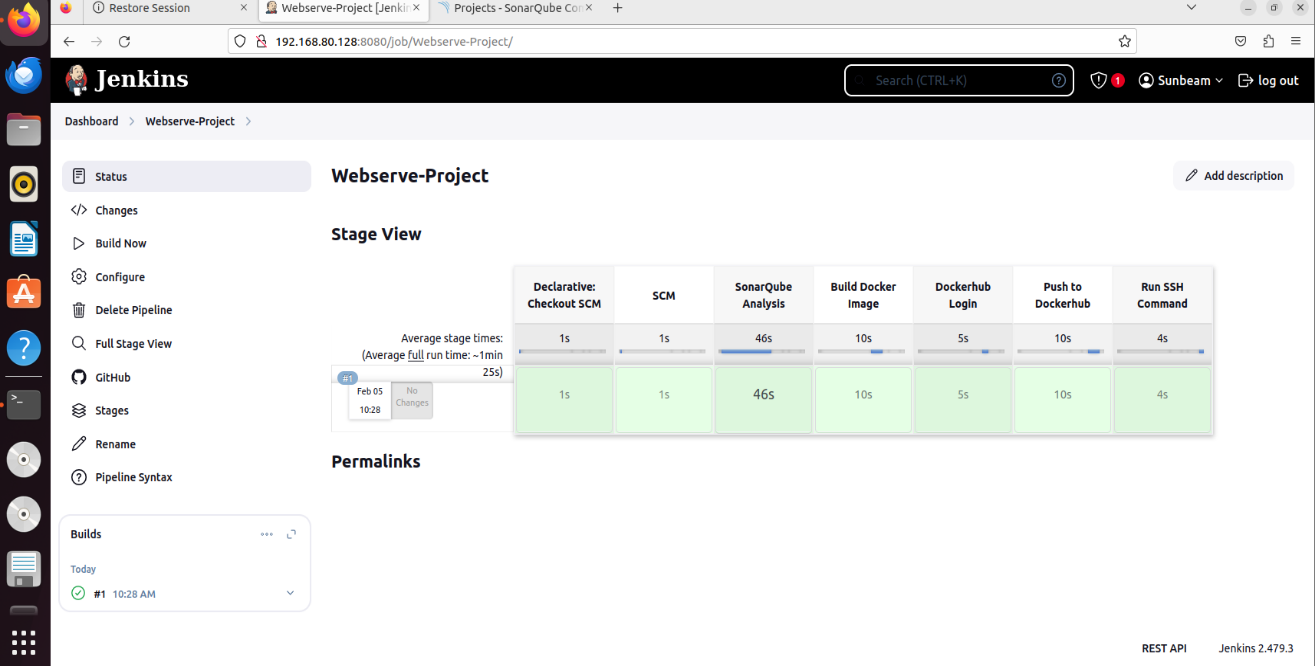
* 1. **SonarQube**



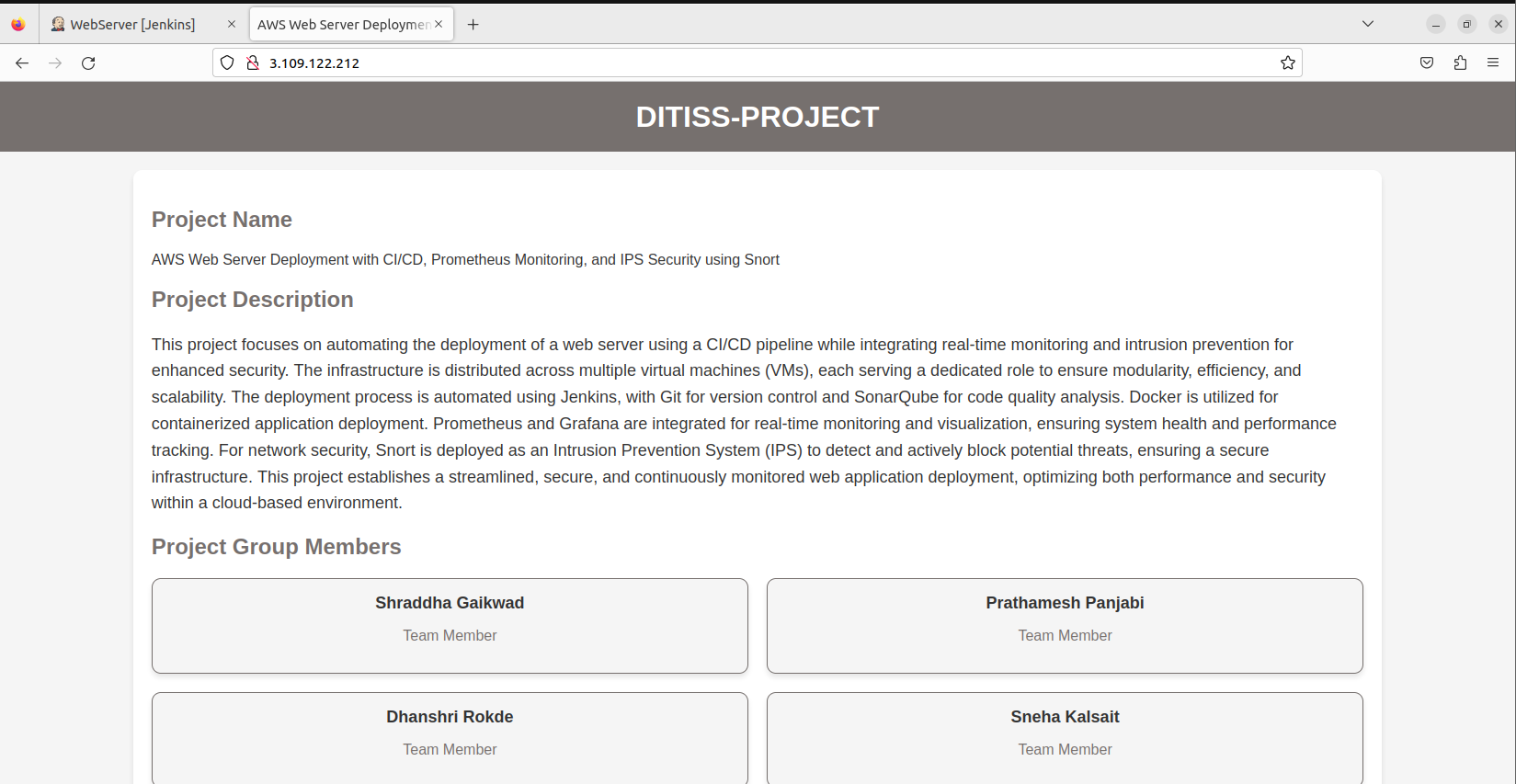
* 1. **Jenkins**



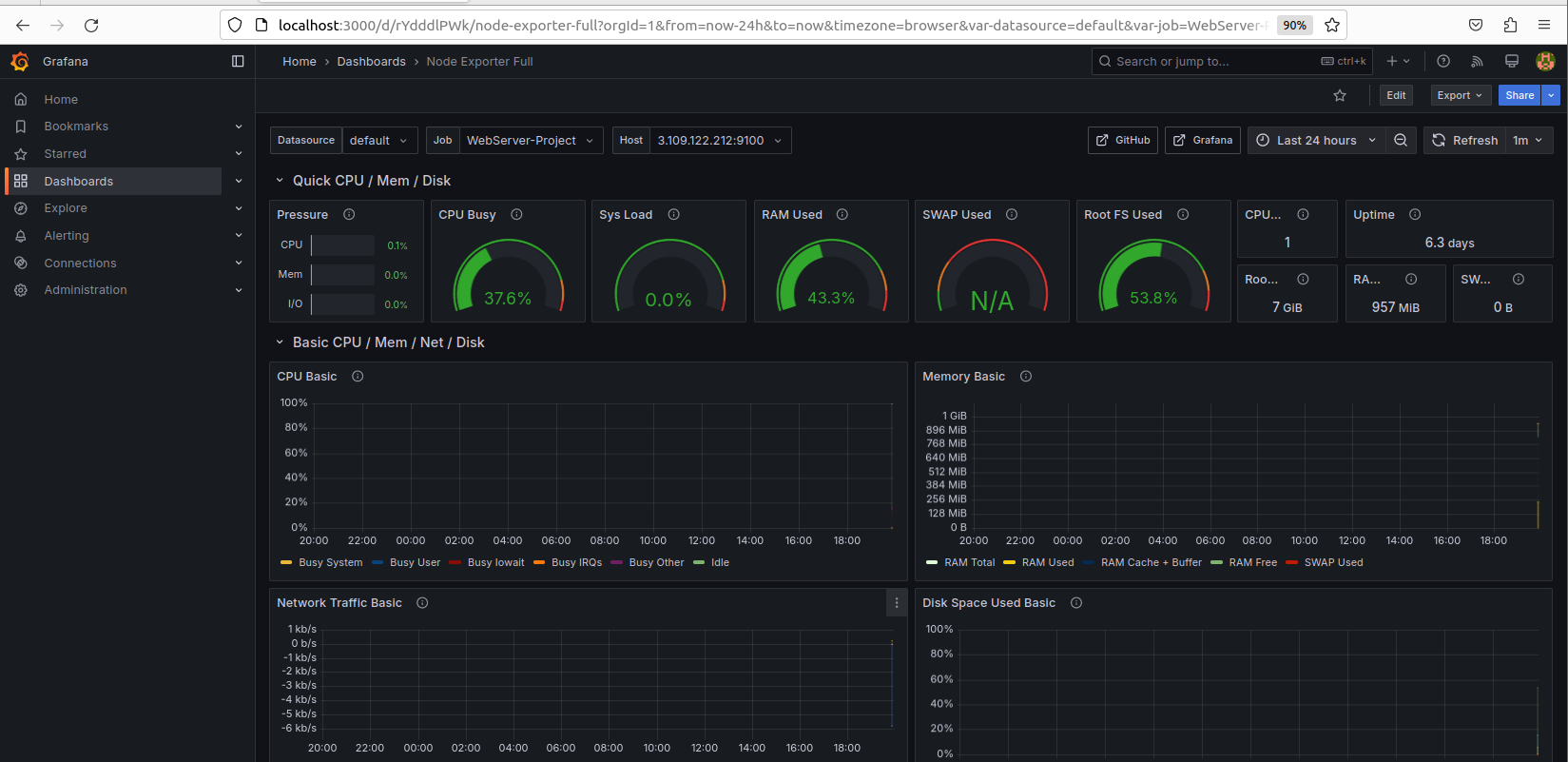
* + 1. **Jenkins Stage View**



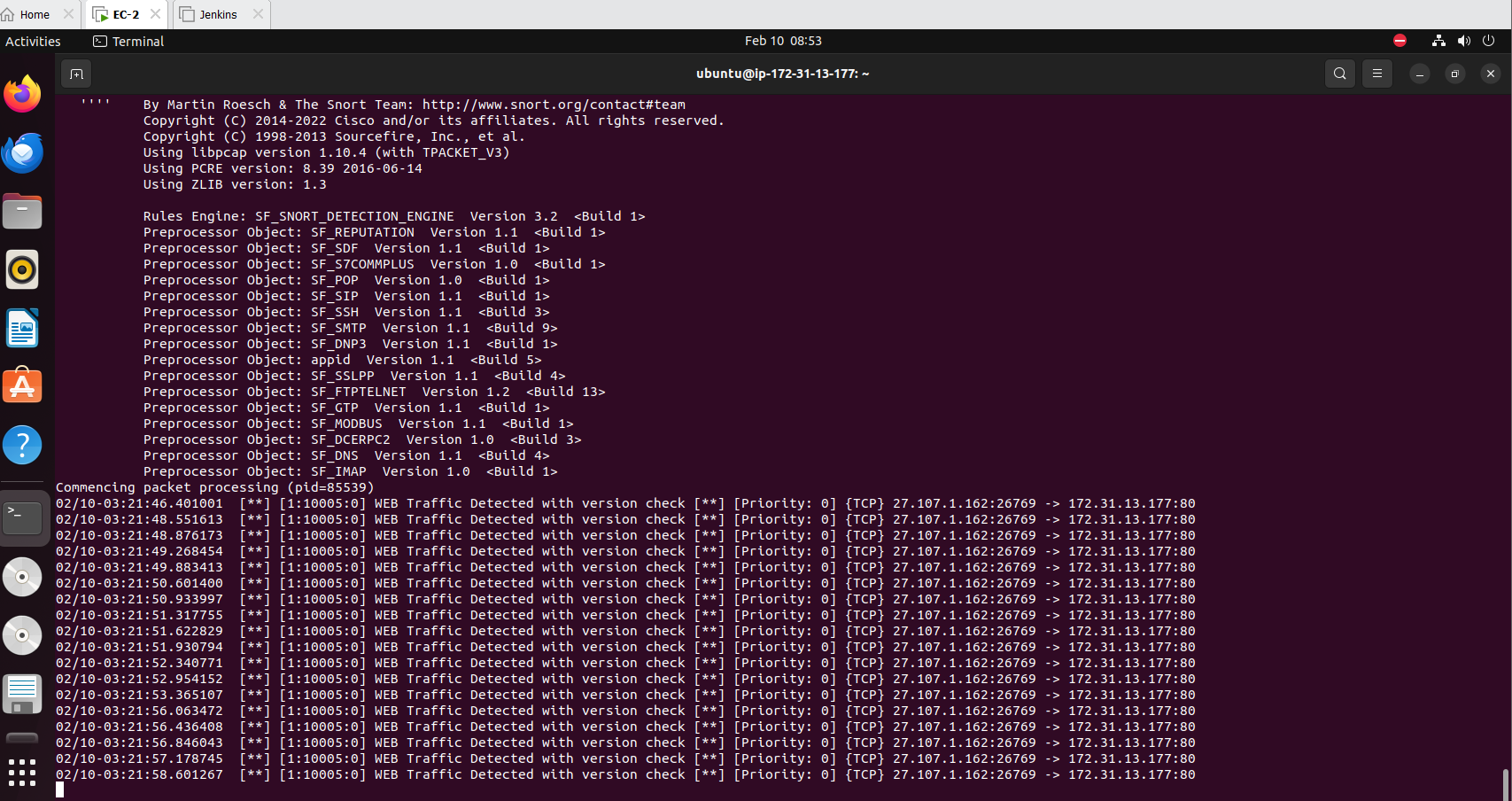
* 1. **Web Application**



* 1. **Prometheus Integrated with Grafana**



* 1. **Snort**



**5. CONCLUSION**

**5.1 Conclusion**

This project successfully implemented a comprehensive solution for deploying and managing web applications with a focus on automation, performance monitoring, and security. By utilizing AWS EC2 for scalable infrastructure and Jenkins for CI/CD pipelines, the project enabled seamless deployment and faster release cycles. Prometheus and Grafana provided real-time monitoring, ensuring proactive system management, while Snort served as an Intrusion Detection System to safeguard the network. SonarQube further enhanced code quality throughout the development lifecycle. This integrated approach ensures improved operational efficiency, security, and scalability for modern web applications.

**5.2 Future Scope**

The project can be extended by integrating serverless computing with AWS Lambda to simplify infrastructure management and scale automatically. Kubernetes can be used for more efficient container orchestration and to manage large-scale deployments.

Another important area for improvement is the CI/CD pipeline. Currently, Jenkins is used for automation, but incorporating Infrastructure as Code (IaC) tools like Terraform or AWS CloudFormation would streamline and automate the provisioning of AWS infrastructure. This would eliminate the need for manual instance setup and make deployments more efficient.

Security in the pipeline could also be improved by integrating OWASP ZAP for security testing and Trivy for container vulnerability scanning, ensuring that the deployed application remains secure and compliant with industry best practices.

On the security front, Snort serves as an excellent intrusion prevention system, but a more advanced security solution could involve AI-driven threat detection using tools like Wazuh (SIEM) or Suricata. These tools provide real-time anomaly detection, log correlation, and automated incident response, offering better protection against sophisticated cyber threats.

**REFERENCES**

**Paper 1:** A Design and Implement of IPS Based on Snort

Author: Jianrong Xi

**Paper 2:** Implementing an Effective Infrastructure Monitoring Solution with Prometheus and Grafana

Author: Pragathi B.C., Hrithik Maddirala, Sneha M.

**Paper 3:** The Proposed Pre-Configured Deployment Model for Amazon EC2 Cloud Services

Author: [Anurag Choudhary](https://ieeexplore.ieee.org/author/37089688389), [Pradeep Kumar Verma](https://ieeexplore.ieee.org/author/37088663819), [Piyush Rai](https://ieeexplore.ieee.org/author/37089885189)