



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

## Project By

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

## Under the guidance of

Mr. Sandeep Valvekar  
Sunbeam Institute of Information  
Technology,  
Pune (Maharashtra)



## Objective

The goal is to securely deploy a web application on Amazon Web Services (AWS) using an automated CI/CD pipeline managed by Jenkins, while ensuring continuous monitoring and intrusion detection using Nagios and Snort. This deployment should emphasize security, automation, and real-time monitoring to safeguard the application against potential threats and ensure high availability.



## AWS Account Creation

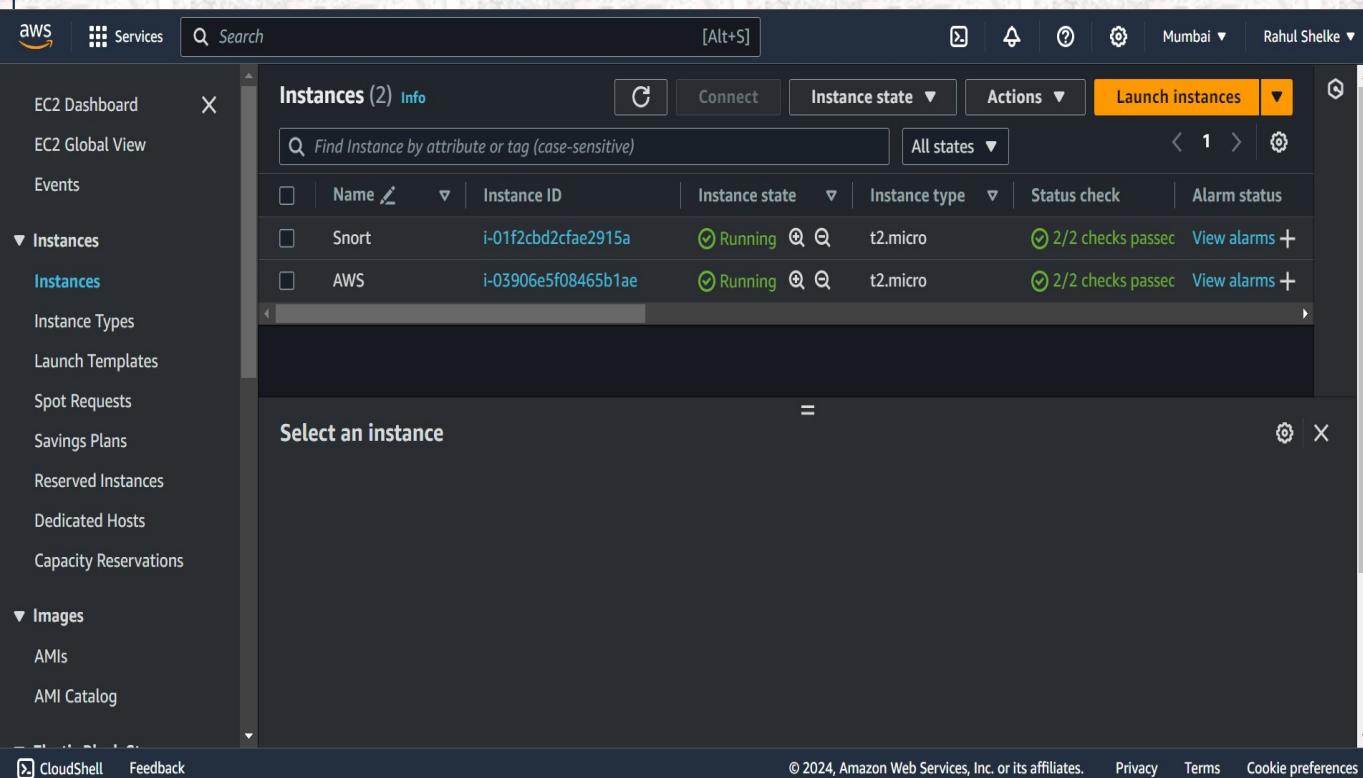
- Visit AWS Website: <https://aws.amazon.com/>
- Sign Up: Visit the AWS signup page, provide email, account name, and select account type.
- Verify and Pay: Complete phone verification, enter payment details, and choose a support plan.
- Access and Configure: Log in to the AWS Management Console, configure account settings, and start using AWS services.

## AWS Instance Creation

- Log In to AWS Management Console
- Access the AWS Management Console
- Navigate to EC2 Dashboard
- Select “EC2” from the “Services” menu to open the EC2 dashboard.
- Launch Instance



# AWS Running Instances:



The screenshot shows the AWS EC2 Instances page with the following details:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
Snort	i-01f2cbd2cfae2915a	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>
AWS	i-03906e5f08465b1ae	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>

Below the table, a modal window titled "Select an instance" is open, listing the same two instances: Snort and AWS.

## Instance1: AWS Instance

We Configured Jenkins and Nagios on this instance.

## Instance2: Snort

Snort is configured on this instance.

## Note:

Both the instances are on same region having same VPC so that we can ping those instances with each other.

# Jenkins Installation & Configuration



Sign in to Jenkins

Username

Password

Keep me signed in

**Sign in**

## Install Jenkins:

On Debian/Ubuntu, install Jenkins using apt-get command.

## Access Jenkins Web Interface:

Open Jenkins in your browser at `http://your-server-ip:8080` and retrieve the initial admin password with sudo cat

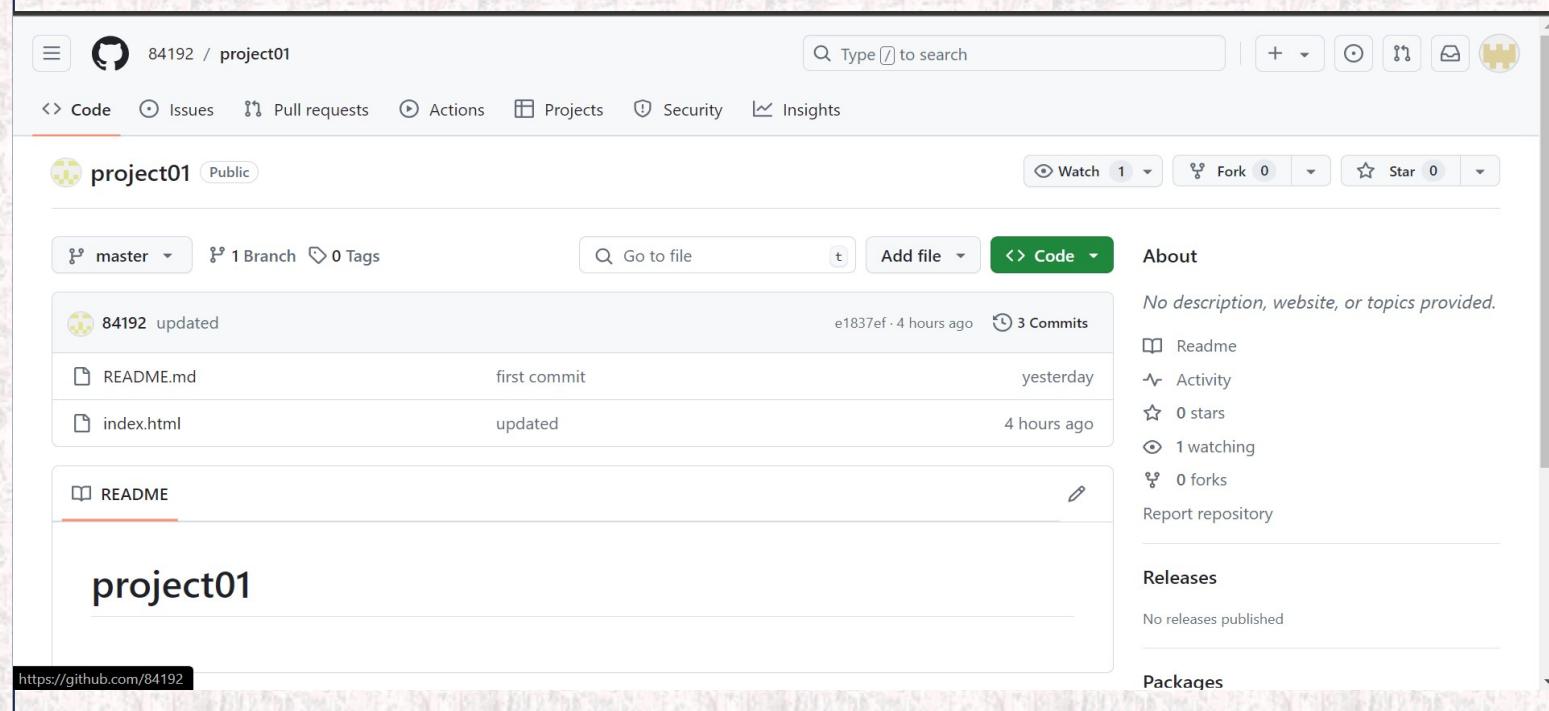
`/var/lib/jenkins/secrets/initialAdminPassword`.

## Complete Initial Setup:

Use the admin password to unlock Jenkins, install suggested plugins, and set up an admin user through the setup wizard.



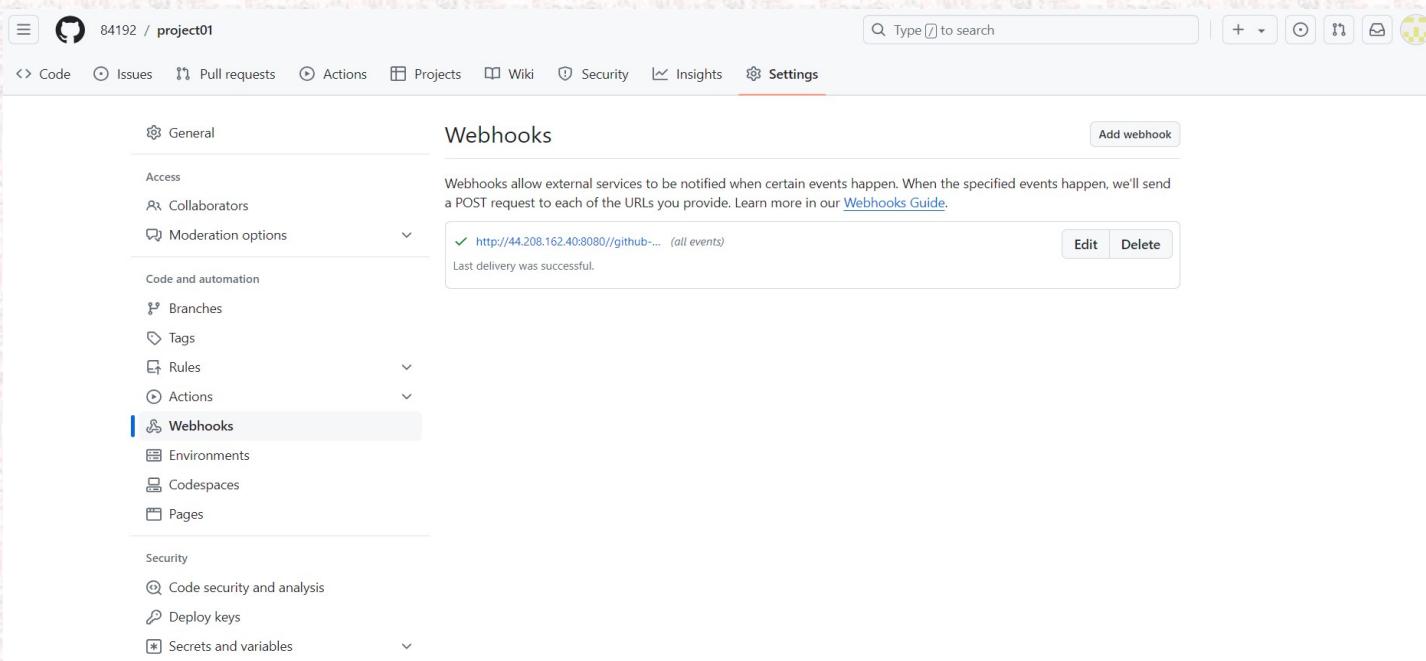
# GitHub:



The screenshot shows a GitHub repository page for 'project01'. The repository is public and has 84192 commits. It contains 1 branch ('master') and 0 tags. The most recent commit is 'first commit' by '84192' updated yesterday. The repository also contains 'README.md' and 'index.html' files. The 'About' section notes 'No description, website, or topics provided.' and lists 3 commits, 0 stars, 1 watching, and 0 forks. The 'Readme' file is shown with its content: 'project01'. The URL 'https://github.com/84192' is visible at the bottom.

- Create a Account on GitHub
- Create a New Repository in GitHub
- Push the index.html file which is Created with the html Code of Static Website.

# Webhook



The screenshot shows the GitHub settings interface for a repository named 'project01'. The 'Webhooks' section is selected in the sidebar. A single webhook is listed with the URL `http://44.208.162.40:8080/github-webhook/` and the event type `(all events)`. The status indicates that the last delivery was successful.

- For Webhook Creation in GitHub, go to the settings, then click on the Webhook.
- Add Webhook
- Paste Payload URL  
Example:  
`http://44.208.162.40:8080//github-webhook/`
- Select Content Type:  
`Application/x-www-form-urlencoded`
- Click on Create Webhook



# Pipeline Creation:

Dashboard > DITISS Project > Configuration

## Configure

Pipeline

Definition

Pipeline script

```
Script ?
```

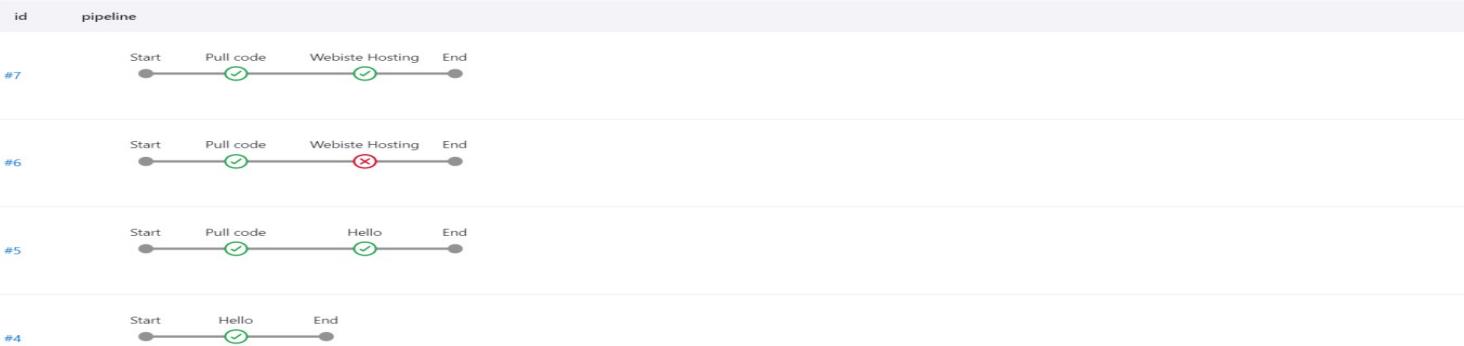
```

1 pipeline {
2   agent any
3   stages {
4     stage ('SCM Checkout') {
5       steps {
6         git branch: 'master', url: 'https://github.com/84192/project01'
7         sh 'ls -l'
8       }
9     }
10   }
11 }
12 }
```

Save Apply

- In Jenkins, click on New Item
- Enter Item Name
- Select Pipeline Project
- Enter Project Description
- Select GitHub Project and Paste its https URL.
- Build Triggers, Click on GitHub Hook Trigger GITScm Pooling.
- Select Pipeline Script & write your Pipeline Script over there.
- Apply & Save
- After that, click on Build
- Check the Console Output
- If there is any Error, Troubleshoot it and Resolve it.

### Build project02



- Status
- Changes
- Console Output
- Edit Build Information
- Delete build '#6'
- Timings
- Git Build Data
- Pipeline Overview
- Pipeline Console
- Restart from Stage
- Replay
- Pipeline Steps

Success

### Console Output

Download  Copy  View as plain text

```

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

```



# Nagios

Not secure 15.206.88.102/nagios/

**Nagios®**

- General
  - Home
  - Documentation
- Current Status
  - Tactical Overview
  - Map (Legacy)
  - Hosts
  - Services
  - Host Groups
    - Summary
    - Grid
  - Service Groups
    - Summary
    - Grid
  - Problems
    - Services (Unhandled)
    - Hosts (Unhandled)
    - Network Outages
  - Quick Search:
- Reports
  - Availability
  - Trends (Legacy)
  - Alerts
    - History
    - Summary
    - Histogram (Legacy)
  - Notifications
  - Event Log

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

**Host Status Totals**  
Up Down Unreachable Pending  
1 0 0 0 0  
All Problems All Types  
0 1

**Service Status Totals**  
Ok Warning Unknown Critical Pending  
7 0 0 1 0  
All Problems All Types  
1 8

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

**Service Status Details For All Hosts**

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	08-13-2024 21:37:01	0d 10h 5m 50s	1/4	OK - load average: 0.00, 0.00, 0.00
localhost	Current Users	OK	08-13-2024 21:37:39	0d 10h 5m 12s	1/4	USERS OK - 3 users currently logged in
localhost	HTTP	OK	08-13-2024 21:33:16	0d 10h 4m 35s	1/4	HTTP OK - HTTP/1.1 200 OK - 2877 bytes in 0.001 second response time
localhost	PING	OK	08-13-2024 21:33:54	0d 10h 3m 57s	1/4	PING OK - Packet loss = 0%, RTA = 0.04 ms
localhost	Root Partition	OK	08-13-2024 21:34:31	0d 10h 3m 20s	1/4	DISK OK - free space: / 4145 MB (55.88% inode=86%)
localhost	SSH	OK	08-13-2024 21:35:09	0d 10h 7m 42s	1/4	SSH OK - OpenSSH_9_2p1 Debian-2+deb12u3 (protocol 2.0)
localhost	Total Processes	OK	08-13-2024 21:36:24	0d 10h 6m 27s	1/4	PROCS OK: 36 processes with STATE = R/SZDT

Results 1 - 7 of 7 Matching Services

Page Tour

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



# Snort

admin@ip-172-31-46-193: ~

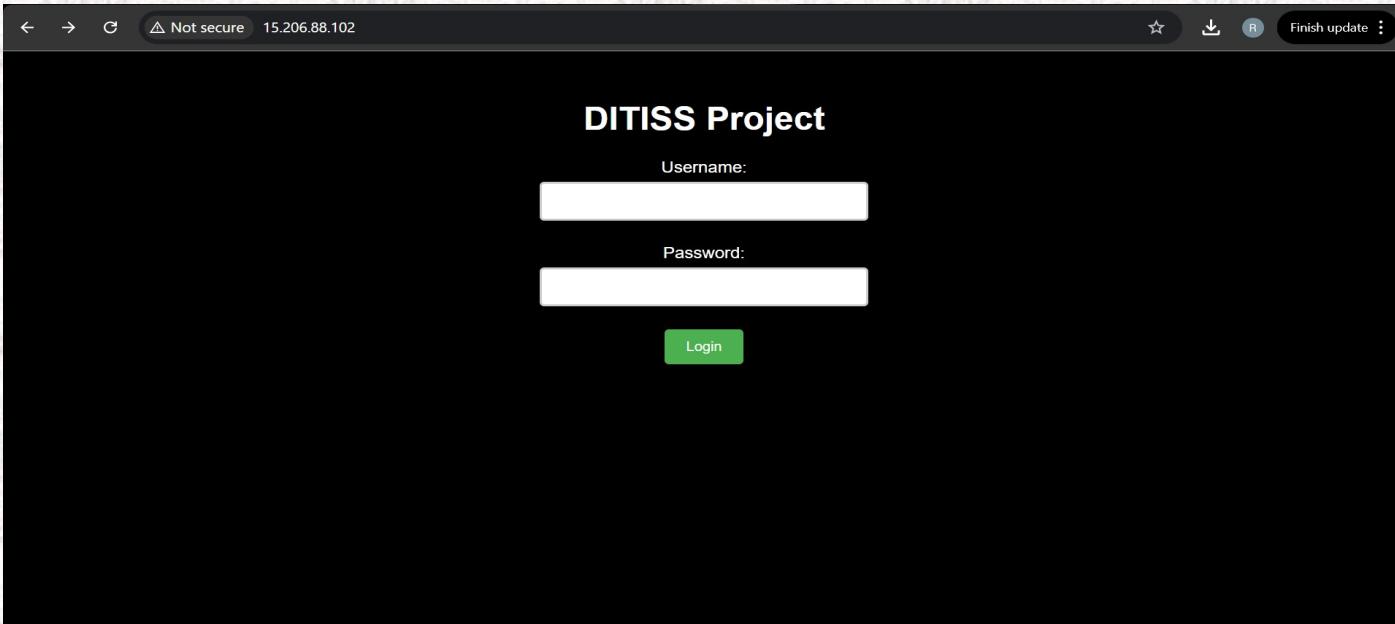
```
08/13-16:17:47.855570 [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 194.26.29.221:44465 -> 172.31.46.193:8398
08/13-16:17:47.855595 [**] [1:1000001:0] 'This is my Project' [**] [Priority: 0] {TCP} 172.31.46.193:8398 -> 194.26.29.221:44465
08/13-16:17:48.310694 [**] [1:1000001:0]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

## Snort Does the following work:

- Detect Intrusions:** Identify and alert on malicious or unauthorized network activities, such as attacks or breaches.
- Prevent Attacks:** Block or mitigate threats by using IPS capabilities to prevent harmful traffic from reaching the network.
- Analyze Network Traffic:** Provide detailed logs and reports for forensic analysis and network security assessments.
- Enhance Security:** Improve overall network security by integrating with other security measures and offering customizable rules for detecting a wide range of threats.

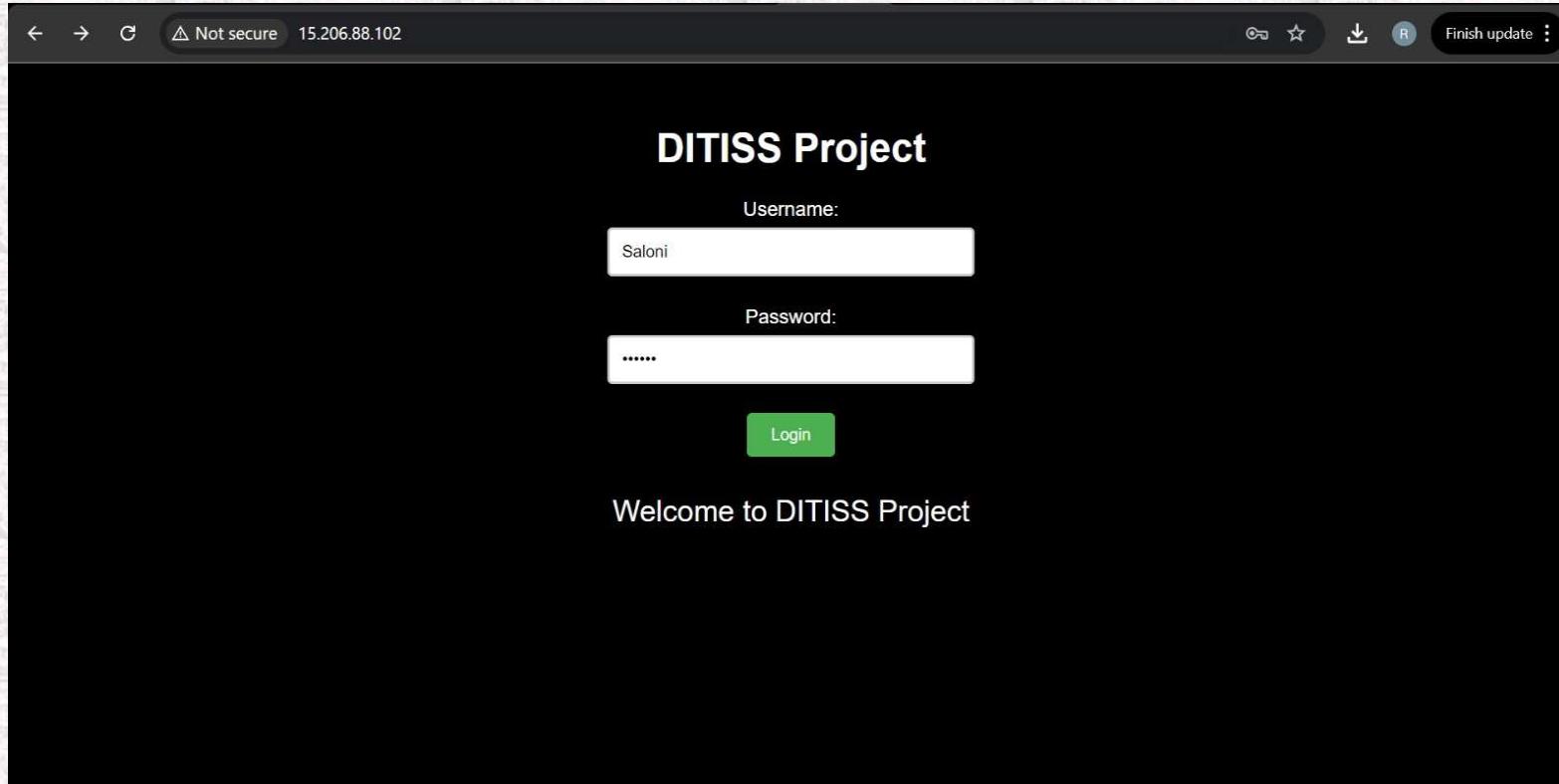


# Static Website Output



The screenshot shows a web browser window with a black background. At the top, the address bar displays "Not secure 15.206.88.102". The main content area is titled "DITISS Project". It contains two input fields: one labeled "Username:" and another labeled "Password:", both with redacted content. Below these fields is a green "Login" button.

- The Output of our Project, that is a **Static Website** will look like this.



Not secure 15.206.88.102

DITISS Project

Username:

Saloni

Password:

.....

Login

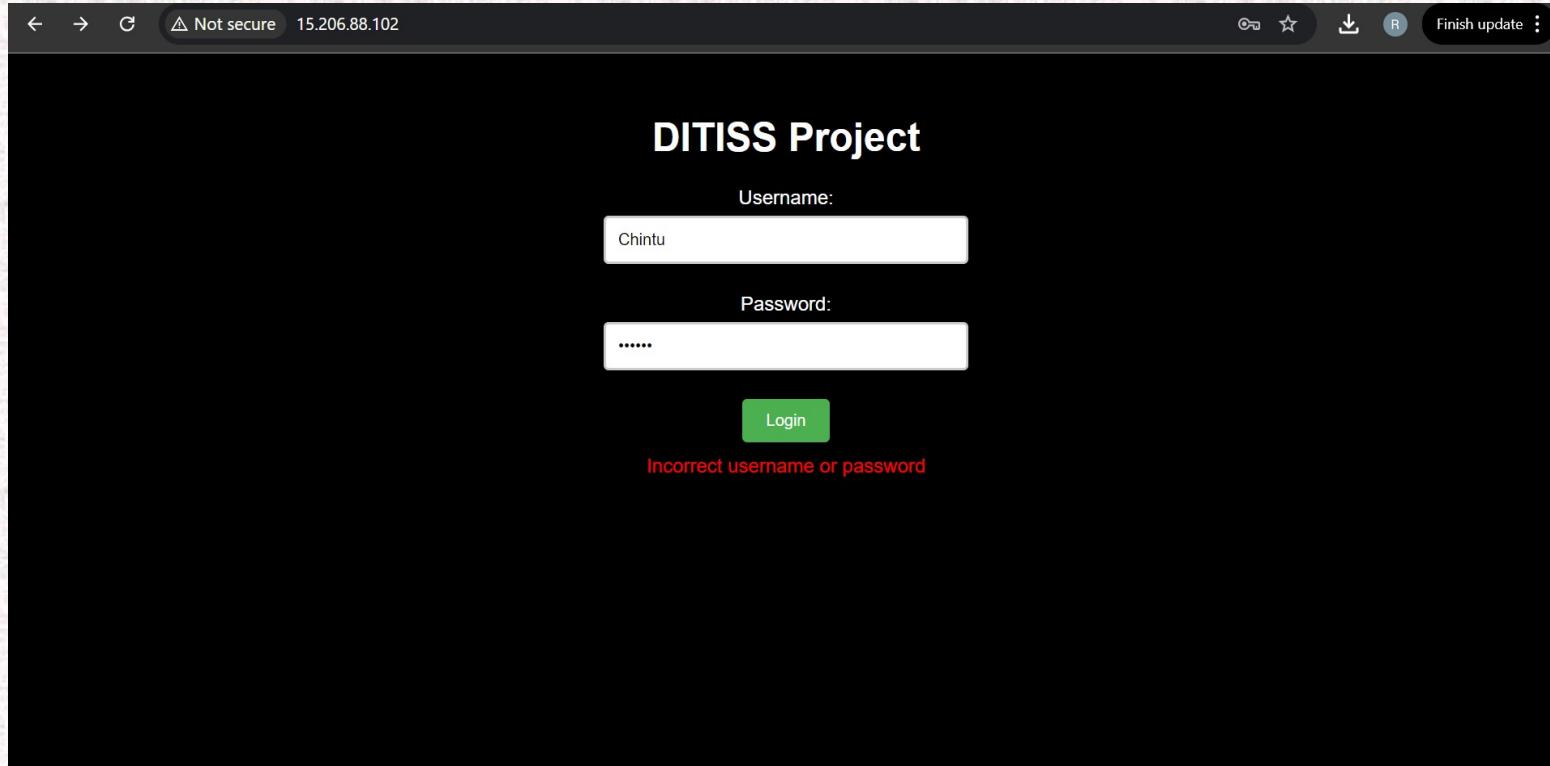
Welcome to DITISS Project

We have added Credentials of our 4 Group Members with Usernames:

- **Rahul**
- **Saloni**
- **Tejeswar**
- **Jayesh**

Each Member has given a particular Password.

If the Credentials Entered are Correct, it will show “**Welcome to DITISS Project**”.



Not secure 15.206.88.102

Finish update

## DITISS Project

Username:

Password:

Login

Incorrect username or password

If the Credentials are not Correct, it will show  
**“Incorrect username or password”**

## 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.



## Future Scope:

The future scope for Secure Web Application Deployment on AWS with Jenkins CI/CD, Nagios monitoring, and Snort IDS involves advancing automation, security, and scalability. Enhanced integration of Jenkins with Infrastructure-as-Code (IaC) and serverless technologies will streamline deployments and improve efficiency. Advanced threat detection and response can be achieved by integrating Snort with machine learning and other security tools, while comprehensive monitoring solutions combining Nagios with real-time analytics will offer better visibility and proactive management. Additionally, adopting containerization, improving disaster recovery strategies, and ensuring compliance with evolving regulations will further bolster the resilience and security of web applications.



## CERTIFICATE

This is to certify that the project report entitled "**Web Application Deployment on AWS with Jenkins CI/CD Pipeline, Nagios Monitoring and Snort IDS**", submitted by **Saloni Chambhare (PRN Number: 240344223007)** is the authentic work completed under our careful supervision and guidance. This project fulfills the requirements for the award of Post Graduate Diploma in IT Infrastructure, Systems, and Security (PG-DITISS) at Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune (Maharashtra)

Date: 14-08-2024

Mr. Sandeep Walvekar  
**(Guide)**

Mr. Nitin Kudale  
**(CEO)**

Mr. Vishal Salunkhe  
**(Course Coordinator)**

**Sunbeam Institute of Information  
Technology**

**Pune (M.S.) – 411057**



प्रगत संगणन विकास केंद्र  
CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING

# Thank You!

PG-DITISS SUNBEAM