

ADVANCE DEVOPS CASE STUDY

CASE STUDY TOPIC:

Automated Deployment with Monitoring

- **Concepts Used:** Jenkins, EC2, Nagios.
- **Problem Statement:** "Set up a Jenkins CI/CD pipeline to deploy a simple web application on an EC2 instance. Configure Nagios to monitor the deployed application's availability."
- **Tasks:**
 - Create a Jenkins pipeline that builds and deploys a sample web app to an EC2 instance.
 - Install and configure Nagios to monitor the HTTP status of the deployed application.
 - Verify the pipeline by triggering a build and checking the monitoring status in Nagios.

INTRODUCTION

Case Study Overview:

1. Title: Automated Deployment with Monitoring
2. Focus: This case study explores the integration of Continuous Integration and Continuous Deployment (CI/CD) practices through Jenkins for deploying a web application on AWS EC2.
3. Objective: The primary aim is to automate the deployment process, allowing for faster release cycles and reduced risk of human error, thereby enhancing the overall efficiency of the software development lifecycle.
4. Monitoring Aspect: After deployment, Nagios is configured to monitor the application's availability and health, providing real-time alerts and insights into the application's operational status.

Key Feature and Application:

1. Automation:
 - Jenkins Pipeline: The Jenkins CI/CD pipeline automates the entire build, test, and deployment process. This allows developers to focus on writing code rather than managing deployments.
 - Continuous Integration: Changes made by developers can be automatically tested and integrated into the main branch, ensuring that the application remains stable and functional.
2. Real-time Monitoring:

- Nagios Monitoring: Nagios continuously checks the HTTP status of the deployed application, enabling proactive identification of issues before they affect users.
 - Alerting System: The monitoring setup includes an alerting mechanism that sends notifications to the development team whenever a critical issue arises, ensuring quick response times to mitigate downtime.
3. Benefits:
- Enhanced Reliability: By automating deployment and monitoring, organizations can ensure that their applications are always available and performing optimally.
 - Improved User Trust: Reliable service availability fosters user confidence, encouraging them to engage more frequently with the application.
 - Reduced Downtime: Prompt detection and resolution of issues minimize downtime, leading to a better user experience.

Third-Year Project Integration:

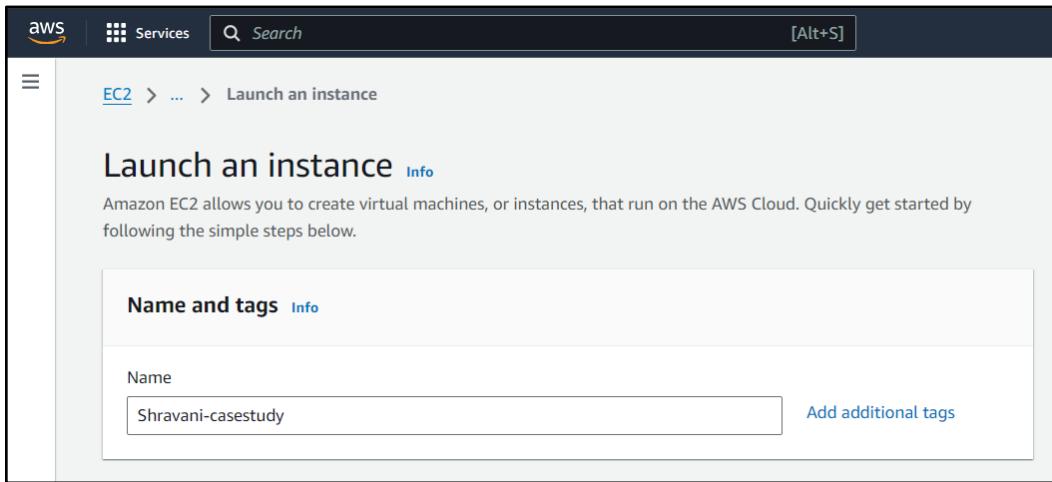
1. Project Title: Glamease – A salon appointment booking system.
2. Project Features:
 - User Capabilities: Customers can book appointments, purchase products, and compare various services based on pricing and ratings.
 - User Interface: The platform features a user-friendly interface that simplifies navigation and enhances user engagement.
3. Application of Case Study Principles:
 - Automated Deployment:
 - Integrating a Jenkins CI/CD pipeline for Glamease will facilitate the automation of deploying new features and updates. This allows for quicker rollouts of enhancements and bug fixes.
 - Ensures that every code change is validated and deployed seamlessly, maintaining high software quality.
 - Monitoring:
 - Implementing Nagios for monitoring critical services within the Glamease platform, such as appointment bookings and product transactions, ensures continuous availability and reliability.
 - With Nagios, I can set up alerts for key performance indicators, allowing for proactive management of system resources and services.
4. Outcome:
 - Enhanced Reliability: The combination of Jenkins and Nagios will provide a robust foundation for the Glamease platform, ensuring it can handle user demands effectively.
 - User Experience: By automating deployment and ensuring ongoing monitoring, users will benefit from a seamless experience when booking appointments and purchasing products.

IMPLEMENTATION

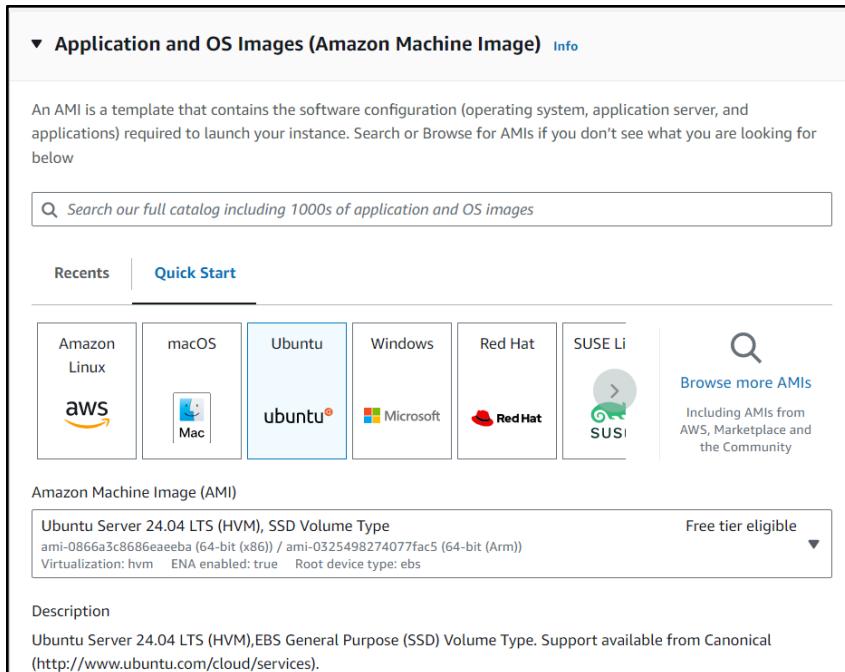
A] Create a Jenkins pipeline that builds and deploys a sample web app to an EC2 instance.

1. Launch EC2 instance:

- Click on "Launch Instance" and configure:



- AMI: Select Amazon Linux 2 or Ubuntu Server 22.04.



- **Instance type:** t2.micro (Free-tier eligible).
- **Key pair:** Create or use an existing key pair for SSH access.

Instance type [Info](#) | [Get advice](#)

Instance type

t2.micro	Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true	
On-Demand Windows base pricing: 0.0162 USD per Hour	
On-Demand SUSE base pricing: 0.0116 USD per Hour	
On-Demand RHEL base pricing: 0.026 USD per Hour	
On-Demand Linux base pricing: 0.0116 USD per Hour	

All generations Compare instance types

Additional costs apply for AMIs with pre-installed software

Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

[Create new key pair](#)

- **Launch the instance.**

Instances (1/2) [Info](#) Last updated less than a minute ago [C](#) [Connect](#) [Instance state ▾](#) [Actions ▾](#) [Launch instances](#) [▼](#)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Shravani-cases...	i-04940de4104bcaa62	Stopped	t2.micro	-	View alarms +	us-east-1b
Shravani-Case...	i-06d298e63d264a37d	Running	t2.micro	Initializing	View alarms +	us-east-1b

i-06d298e63d264a37d (Shravani-CaseStudy)

[Details](#) | [Status and alarms](#) | [Monitoring](#) | [Security](#) | [Networking](#) | [Storage](#) | [Tags](#)

Instance summary [Info](#)

Instance ID i-06d298e63d264a37d	Public IPv4 address 44.212.228.250 open address	Private IPv4 addresses 172.31.36.248
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-44-212-228-250.compute-1.amazonaws.com open address
Hostname type IP name: ip-172-31-36-248.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-36-248.ec2.internal	

2. Configure inbound rules

- SSH (TCP 22): Your IP.
- HTTP (TCP 80): Open to all (0.0.0.0/0).
- Nagios (TCP 5666): Open to all (0.0.0.0/0).
- All TCP (TCP 0-65535): Open to all (0.0.0.0/0).
- SMTP (TCP 587): Open to all (0.0.0.0/0)
- 8080 (Jenkins)

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
	Info				
sgr-02048d8cdb31f0c27	Custom TCP	TCP	5666	Custom	0.0.0.0 X
sgr-0621b691341f60f42	All traffic	All	All	Custom	0.0.0.0 X
sgr-0f387cb8fa1c3882b	SSH	TCP	22	Custom	0.0.0.0 X
sgr-0365fbcb93ecd31cd	SMTP	TCP	25	Custom	0.0.0.0 X
sgr-05cc3359274f1c64a	All ICMP - IPv6	IPv6 ICMP	All	Custom	0.0.0.0 X
sgr-04347aac480e086f7	All ICMP - IPv4	ICMP	All	Custom	0.0.0.0 X
sgr-00f34dc5a3a33121a	HTTP	TCP	80	Custom	0.0.0.0 X
sgr-00d3ba95d44ef91c	Custom TCP	TCP	8080	Custom	0.0.0.0 X
sgr-08db4b22d9836649b	HTTPS	TCP	443	Custom	0.0.0.0 X

3. Configure Outbound rule for email-notification

- SMTP (TCP 587): Open to all (0.0.0.0/0), for sending emails through Gmail's SMTP server.

Outbound rules (2)						
	Name	Security group rule...	IP version	Type	Protocol	Port range
<input type="checkbox"/>	-	sgr-070338294a733a...	IPv4	SMTP	TCP	25
<input type="checkbox"/>	-	sgr-08e746222e3177...	IPv4	All traffic	All	0.0.0.0/0

4. Allocate an Elastic IP Address

- Go to the AWS Management Console.
- Open the EC2 service.
- In the left-hand menu, under Network & Security, click on Elastic IPs.
- Click the Allocate Elastic IP address button.
- Choose the Amazon pool of IPv4 addresses, then click Allocate.

The screenshot shows the 'Allocate Elastic IP address' wizard. The navigation path is EC2 > Elastic IP addresses > Allocate Elastic IP address. The main section is titled 'Elastic IP address settings' with an 'Info' link. It shows the 'Public IPv4 address pool' configuration. The 'Amazon's pool of IPv4 addresses' option is selected. Other options include 'Public IPv4 address that you bring to your AWS account with BYOIP' (disabled), 'Customer-owned pool of IPv4 addresses created from your on-premises network for use with an Outpost' (disabled), and 'Allocate using an IPv4 IPAM pool' (disabled). Below this, the 'Network border group' field contains 'us-east-1'. At the bottom, there is a note about AWS Global Accelerator providing global static IP addresses.

- **Associate Elastic IP with Your EC2 Instance**

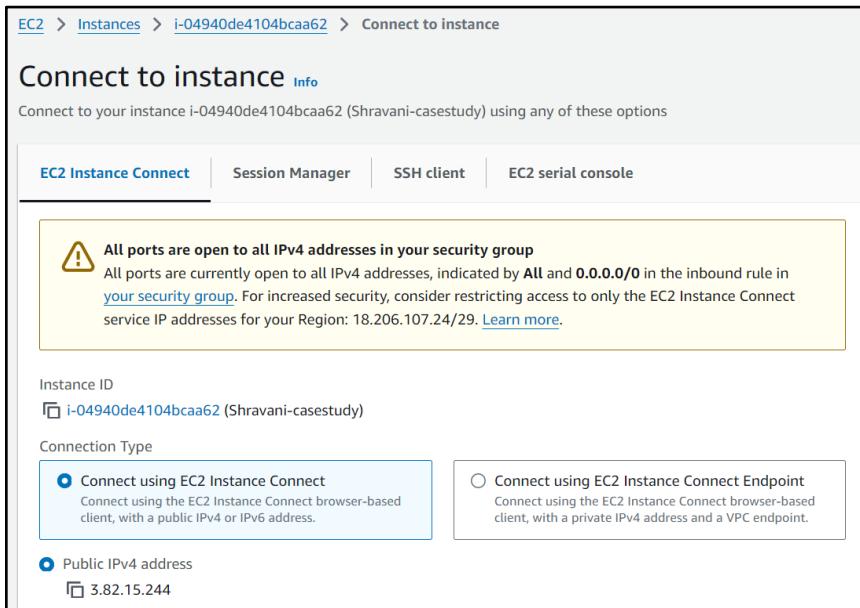
- Click Actions and choose Associate Elastic IP address.
- In the Instance field, select the EC2 instance you want to associate with the Elastic IP.
- Under Private IP address, choose the private IP associated with the instance (usually the default one).

The screenshot shows the 'Elastic IP addresses' list. A dropdown menu is open for 'Elastic IP addresses (1)'. A search bar at the top allows filtering. The table lists one item: a Public IP address '44.212.228.250' allocated from the 'amazon' address pool, with an allocation ID of 'eipalloc-0756b6f661e45df09'.

Name	Allocated IPv4 address	Type	Address pool	Allocation ID
-	44.212.228.250	Public IP	amazon	eipalloc-0756b6f661e45df09

5. Connect to the Instance:

- Click the Connect button at the top of the Instances page.
- In the Connect to instance window, choose the EC2 Instance Connect option.
- Ensure the username is set correctly (e.g., `ec2-user` for Amazon Linux, `ubuntu` for Ubuntu).
- Click the Connect button.



6. Install Apache (if not installed)

- `sudo apt update`
- `sudo apt install apache2`

```
ubuntu@ip-172-31-33-21:~$ sudo apt update
sudo apt install apache2
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [431 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [597 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [146 kB]
```

```
ubuntu@ip-172-31-33-21:~$ sudo service apache2 start
ubuntu@ip-172-31-33-21:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Fri 2024-10-18 21:29:37 UTC; 1min 21s ago
     Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 2717 (apache2)
    Tasks: 55 (limit: 1130)
   Memory: 5.3M (peak: 5.6M)
      CPU: 36ms
     CGroup: /system.slice/apache2.service
             └─2717 /usr/sbin/apache2 -k start
                 ├─2720 /usr/sbin/apache2 -k start
                 ├─2721 /usr/sbin/apache2 -k start
                 └─2721 /usr/sbin/apache2 -k start

Oct 18 21:29:37 ip-172-31-33-21 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Oct 18 21:29:37 ip-172-31-33-21 systemd[1]: Started apache2.service - The Apache HTTP Server.
ubuntu@ip-172-31-33-21:~$ █
```

Enable Apache to Start on Boot

```
sudo systemctl enable apache2
```

```
ubuntu@ip-172-31-33-21:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
ubuntu@ip-172-31-33-21:~$ █
```



The screenshot shows the Apache2 Default Page, which is a standard test page for web servers. It features the Ubuntu logo (a red square with a white circle and dots) and the word "Ubuntu" in a large, lowercase, sans-serif font. To the right of "Ubuntu" is a red button with the white text "It works!". Above the "It works!" button is the title "Apache2 Default Page". Below the main heading, there is a paragraph of text explaining the purpose of the page and instructions for replacing the default file. At the bottom, there is a section titled "Configuration Overview" with a brief description of the configuration files.

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
/-- apache2.conf
```

7. JENKINS Setup

Install Required Jenkins Plugins:

- Log in to Jenkins at <http://localhost:8080>.
- Go to Manage Jenkins -> Manage Plugins.
- Install the following plugins:
 - AWS EC2 Plugin (for managing EC2 instances).
 - SSH Agent Plugin (to SSH into your EC2 instance during deployment).

The screenshot shows the Jenkins Plugins page. The search bar at the top right contains the text "SSH Agent". On the left sidebar, the "Installed plugins" tab is selected. A warning message in an orange box states: "Warning: This Jenkins instance requires a restart. Changing the state of plugins at this time is strongly discouraged. Restart Jenkins before proceeding." Below the warning, two plugins are listed:

Name	Enabled
SSH Agent Plugin 376.v8933585c69d3	<input checked="" type="checkbox"/> Report an issue with this plugin
SSH Build Agents plugin 2.973.v0fa_8c0dea_f9f	<input checked="" type="checkbox"/> Report an issue with this plugin

The screenshot shows the Jenkins Plugins page. The search bar at the top right contains the text "AWS". On the left sidebar, the "Installed plugins" tab is selected. Three AWS-related plugins are listed:

Name	Enabled
Amazon Web Services SDK :: EC2 1.12.772-474.v7f79a_2046a_fb_	<input checked="" type="checkbox"/>
Amazon Web Services SDK :: Minimal 1.12.772-474.v7f79a_2046a_fb_	<input checked="" type="checkbox"/>
AWS Credentials Plugin 231.v08a_59f17d742	<input checked="" type="checkbox"/>

Steps to Handle the SSH Key in Jenkins:

1. Add SSH Credentials in Jenkins:

- Go to Manage Jenkins → Manage Credentials → (global) → Add Credentials.
- Under Kind, select SSH Username with private key.
- Fill in the Username (for example, ubuntu for Ubuntu instances).
- For Private Key, select Enter directly, then paste the content of yourcasestudy.pem file.
- Give it an ID, e.g., ec2-ssh-credential, for reference in the pipeline.

The screenshot shows the 'New credentials' form under the 'Manage Jenkins > Credentials > System > Global credentials (unrestricted)' path. The 'Kind' field is set to 'SSH Username with private key'. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc)'. The 'ID' field is empty. The 'Description' field is also empty. The 'Username' field contains 'ubuntu'. A blue 'Create' button is at the bottom.

The screenshot shows the 'Credentials' list page under the 'Manage Jenkins > Credentials' path. It displays two entries:

T	P	Store ↓	Domain	ID	Name
file	user	System	(global)	2fc6e0c4-51cc-48f9-ab2e-f98ce6155fda	Secret text
key	user	System	(global)	ec2-ssh-credential	ubuntu

Below the table, a section titled 'Stores scoped to Jenkins' shows a single entry:

P	Store ↓	Domains
user	System	(global)

System Configuration of SSH site hosts

- Select **Configure System**.
- Scroll to **SSH Remote Hosts**.
- Click **Add SSH Host**.
 - **Hostname:** 44.212.228.250
 - **Username:** ec2-user
- Select credentials if configured.
- Click **Test Connection** to verify.

The screenshot shows the 'SSH remote hosts' configuration interface. At the top, it says 'SSH sites' and 'SSH sites that projects will want to connect'. Below this, there's a form for adding a new host:

- Hostname**: 44.212.228.250
- Port**: 22
- Credentials**: ubuntu (selected from a dropdown)
- Pty**: (checkbox is unchecked)
- serverAliveInterval**: 0

Below this, there's a section for 'timeout': 0. It also shows a 'Successfull connection' message with a 'Check connection' button. At the bottom, there are 'Add', 'Metrics', and 'Access keys' sections, along with 'Save' and 'Apply' buttons.

Build a Jenkins Pipeline:

New Item

Enter an item name
ShravaniCaseStudy

Select an item type

 Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

 Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

 Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

OK

Jenkins

Search (CTRL+K)

Shravani Rasam log out

Dashboard > ShravaniCaseStudy > Configuration

Configure General Enabled

General

Description

Plain text [Preview](#)

Discard old builds ?
 Do not allow concurrent builds
 Do not allow the pipeline to resume if the controller restarts
 GitHub project

Save **Apply**

```
pipeline {
    agent any
    stages {
        stage('Checkout Code') {
            steps {
                // Clone your repository to get the latest index.html
                git url: 'https://github.com/ShravaniR2412/CaseStudy.git', branch: 'main'
            }
        }
        stage('Build') {
            steps {
                echo 'Building the application...'
            }
        }
        stage('Test SSH Connection') {
            steps {
                script {
                    echo "Testing SSH connection to EC2 instance..."
                    try {
                        sshagent(['ec2-ssh-credential']) {
                            // Attempt to run a simple command on the EC2 instance
                            sh """
                                ssh -o StrictHostKeyChecking=no -i
                                "C:/Users/Shravani/Downloads/casestudy.pem" ubuntu@ec2-44-212-228-250.compute-
                                1.amazonaws.com "echo 'SSH connection successful"
                                """
                            }
                            echo "SSH connection test successful."
                        } catch (Exception e) {
                            echo "SSH connection test failed: ${e.getMessage()}"
                        }
                    }
                }
            }
        }
        stage('Deploy') {
            steps {
                script {
                    // Set the path to the index.html file in the cloned repository
                    def indexPath = "${WORKSPACE}/index.html"

                    echo "Workspace directory: ${WORKSPACE}"
                    echo "Index file path: ${indexPath}"

                    try {

```

```

echo "Copying index.html to EC2 instance..."
sshagent(['ec2-ssh-credential']) {
    sh """
        scp -o StrictHostKeyChecking=no -i
        "C:/Users/Shravani/Downloads/casestudy.pem" ${indexFilePath} ubuntu@ec2-44-212-228-
        250.compute-1.amazonaws.com:/var/www/html/index.html
    """
}

echo "Restarting Apache server..."
sshagent(['ec2-ssh-credential']) {
    sh """
        ssh -o StrictHostKeyChecking=no -i
        "C:/Users/Shravani/Downloads/casestudy.pem" ubuntu@ec2-44-212-228-250.compute-
        1.amazonaws.com "sudo systemctl restart apache2"
    """
}

} catch (Exception e) {
    echo "Error during deployment: ${e.getMessage()}"
}
}
}
}

```

Configure

Pipeline

General

Advanced Project Options 

Pipeline

Definition

Pipeline script

Script 

```
1 * pipeline {
2     agent any
3     stages {
4         stage('Checkout Code') {
5             steps {
6                 // Clone your repository to get the latest index.html
7                 git url: 'https://github.com/ShravaniR2412/CaseStudy.git', branch: 'main'
8             }
9         }
10        stage('Build') {
11            steps {
12                echo 'Building the application...'
13            }
14        }
15        stage('Test SSH Connection') {
16            steps {
17            }
18        }
19    }
20}
```

Use Groovy Sandbox 

Pipeline Syntax

Save **Apply**

Dashboard > ShravaniCaseStudy >

ShravaniCaseStudy

Stage View

Average stage times:
(Average full run time: ~4s)

	Checkout Code	Build	Test SSH Connection	Deploy
#104 Oct 22 19:28 No Changes	2s	140ms	453ms	441ms
#103 Oct 20 20:42 No Changes	2s	104ms	397ms	408ms
#102 Oct 20 20:29 No Changes	1s	113ms	289ms	302ms

Build History trend ▾

Filter... /

#104 | Oct 22, 2024, 7:28 PM

#103

Permalinks

- Last build (#104) 38 min ago

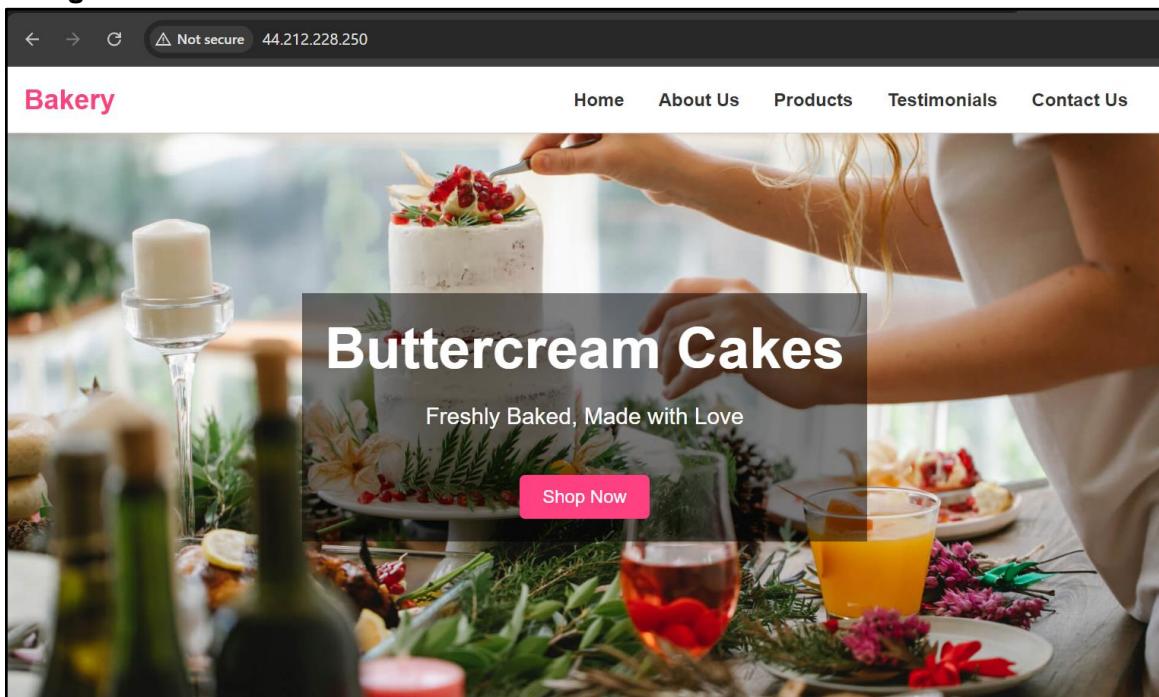
Jenkins

Dashboard > ShravaniCaseStudy > #104

Console Output

Started by user Shravani Rasam
 [Pipeline] Start of Pipeline
 [Pipeline] node
 Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\ShravaniCaseStudy
 [Pipeline] {
 [Pipeline] stage
 [Pipeline] { (Checkout Code)
 [Pipeline] git
 The recommended git tool is: NONE
 No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\ShravaniCaseStudy\.git #
 timeout=10
 Fetching changes from the remote Git repository
 > git.exe config remote.origin.url https://github.com/ShravaniR2412/CaseStudy.git # timeout=10
 Fetching upstream changes from https://github.com/ShravaniR2412/CaseStudy.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.43.0.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/ShravaniR2412/CaseStudy.git
 +refs/heads/*:refs/remotes/origin/* # timeout=10

Navigate to the Elastic IP address in the browser



Our Best Products



Chocolate Cake

Order Now



Strawberry Cake

Order Now



Red Velvet Cake

Order Now

What Our Customers Say

"The best cakes I've ever had! They made my birthday so special with a custom cake just the way I wanted it."

- Jane Doe

"I ordered a cake for my wedding and it was a hit! The design was perfect and the taste was even better."

- John Smith

"Fantastic service and even better cakes! Will definitely be coming back for more."

- Emily Johnson

© 2024 Cake Shop | Designed with [Love](#) by Us

B] Install and configure Nagios to monitor the HTTP status of the deployed application.

Connect to Your EC2 Instance

- SSH into your EC2 instance or use EC2 Instance Connect from the browser

```
sudo apt update
```

```
sudo apt install nagios4 nagios-plugins-contrib nagios-nrpe-plugin -y
```

```
ubuntu@ip-172-31-36-248:~$ sudo apt update
sudo apt upgrade -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [114 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [306 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [21.2 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:13 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [7212 B]
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [51.9 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [208 B]
Get:16 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Fetched 881 kB in 1s (944 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
25 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-36-248:~$ sudo apt upgrade -y
Reading package lists... Done
Building dependency tree... Done
```

```
sudo apt install apache2 php libapache2-mod-php build-essential libgd-dev -y
```

```
ubuntu@ip-172-31-36-248:~$ sudo apt install -y apache2 libapache2-mod-php php gcc make autoconf libgd-dev libmcrypt-dev wget unzip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.58-1ubuntu8.4).
wget is already the newest version (1.21.4-1ubuntu4.1).
wget set to manually installed.
The following additional packages will be installed:
 automake autotools-dev binutils binutils-common binutils-x86-64-linux-gnu bzip2-doc cpp cpp-13-x86-64-linux-gnu
 cpp-x86-64-linux-gnu fontconfig-config fonts-dejavu-core fonts-dejavu-mono gcc-13 gcc-13-base gcc-13-x86-64-linux-gnu
 gcc-x86-64-linux-gnu libao-dev libaom3 libapache2-mod-php8.3 libasan8 libatomic1 libbinutils libbrotli-dev libbz2-dev libc-dev-bin
 libc-devtools libc6-dev libcrypt-dev libctf-nobfd0 libctf0 libdavid-dev libdavid1d7 libde265-0 libde265-dev libdeflate-dev
 libdeflated libexpat1-dev libfontconfig-dev libfontconfig1 libfreetype-dev libgcc-13-dev libgd3 libgomp1 libgprofng0 libheif-dev
 libheif-plugin-aomdec libheif-plugin-aomenc libheif-plugin-libde265 libheif1 libhwasan0 libisl23 libitm1 libjbig-dev libjbig2
 libjpeg-dev libjpeg-turbo8 libjpeg-turbo8-dev libjpeg8 libjpeg8-dev liblrc4 liblsan0 liblzma-dev libmcrypt4 libmpc3
 libpkconfs libpng-dev libpng-tools libpthread-stubs0-dev libquadmath0 libsharpayuv-dev libsharpayuv0 libtiff-dev libtiff6
 libtiffxx6 libtsan2 libubsan1 libvpx-dev libvpx9 libwebp-dev libwebp7 libwebpdecoder3 libwebpdemux2 libwebpmux3 libx11-dev
 libx265-199 libx265-dev libxau-dev libxcb1-dev libxdmcp-dev libxpm-dev libzstd-dev linux-libc-dev m4 manpages-dev
```

Download and install Nagios:

- cd /tmp
- wget <https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz>
- tar -zxvf nagios-4.4.6.tar.gz
- cd nagios-4.4.6/

```
ubuntu@ip-172-31-36-248:/tmp/nagios-4.4.11$ sudo apt-get install -y autoconf gcc libc6-dev make wget apache2 php libapache2-mod-php
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
autoconf is already the newest version (2.71-3).
gcc is already the newest version (4:13.2.0-7ubuntu1).
libc6-dev is already the newest version (2.39-0ubuntu8.3).
libc6-dev set to manually installed.
make is already the newest version (4.3-4.1build2).
wget is already the newest version (1.21.4-1ubuntu4.1).
apache2 is already the newest version (2.4.58-1ubuntu8.4).
php is already the newest version (2:8.3+93ubuntu2).
libapache2-mod-php is already the newest version (2:8.3+93ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Compile and install Nagios:

- ./configure --with-command-group=nagioscmd
- sudo make all
- sudo make install-groups-users
- sudo make install
- sudo make install-daemoninit
- sudo make install-commandmode
- sudo make install-config
- sudo make install-webconf

```
ubuntu@ip-172-31-36-248:/tmp/nagios-4.4.11$ sudo ./configure --with-command-group=nagios
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for ANSI C header files... yes
checking whether time.h and sys/time.h may both be included... yes
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
```

```
ubuntu@ip-172-31-36-248:/tmp/nagios-4.4.11$ sudo make all
cd ./base && make
make[1]: Entering directory '/tmp/nagios-4.4.11/base'
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o nagios.c
nagios.c: In function 'main':
nagios.c:611:25: warning: ignoring return value of 'asprintf' declared wi
  611 |           asprintf(&mac->x[MACRO_PROCESSSTARTTIME], ^
  |           ~~~~~
nagios.c:841:25: warning: ignoring return value of 'asprintf' declared wi
  841 |           asprintf(&mac->x[MACRO_EVENTSTARTTIME], ^
  |           ~~~~~
nagios.c: In function 'nagios_core_worker':
nagios.c:176:17: warning: ignoring return value of 'read' declared with a
  176 |           read(sd, response + 3, sizeof(response) - 4);
  |           ~~~~~
nagios.c: In function 'test_path_access':
nagios.c:122:17: warning: ignoring return value of 'asprintf' declared wi
  122 |           asprintf(&path, "%s/%s", p, program);
  |           ~~~~~
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o broker.o broker.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nebmods.o nebmods.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ../common/shared.o
```

```
ubuntu@ip-172-31-36-248:/tmp/nagios-4.4.11$ sudo make install
cd ./base && make install
make[1]: Entering directory '/tmp/nagios-4.4.11/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiosstats /usr/local/nagios/bin
make[1]: Leaving directory '/tmp/nagios-4.4.11/base'
cd ./cgi && make install
make[1]: Entering directory '/tmp/nagios-4.4.11/cgi'
make install-basic
make[2]: Entering directory '/tmp/nagios-4.4.11/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
    /usr/bin/install -c -s -m 775 -o nagios -g nagios $file /usr/local/nagios/
done
make[2]: Leaving directory '/tmp/nagios-4.4.11/cgi'
make[1]: Leaving directory '/tmp/nagios-4.4.11/cgi'
cd ./html && make install
make[1]: Entering directory '/tmp/nagios-4.4.11/html'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/media
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/styleshe
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/contexth
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/docs
```

```
ubuntu@ip-172-31-36-248:/tmp/nagios-4.4.11$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-available/nagios.conf
if [ 1 -eq 1 ]; then \
    ln -s /etc/apache2/sites-available/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi
*** Nagios/Apache conf file installed **

ubuntu@ip-172-31-36-248:/tmp/nagios-4.4.11$ sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.service
ubuntu@ip-172-31-36-248:/tmp/nagios-4.4.11$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
```

```
ubuntu@ip-172-31-36-248:~$ ubuntu@ip-172-31-36-248:~$ sudo systemctl restart nagios
Warning: The unit file, source configuration file or drop-ins of nagios.service changed on disk.
d units.
ubuntu@ip-172-31-36-248:~$ sudo systemctl daemon-reload
ubuntu@ip-172-31-36-248:~$ sudo systemctl restart nagios
ubuntu@ip-172-31-36-248:~$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.4.11
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2023-04-14
License: GPL

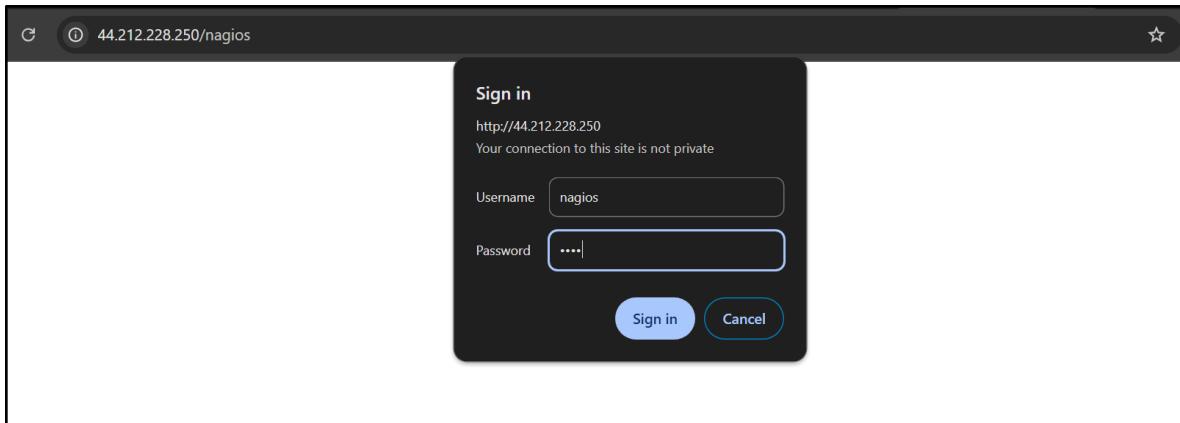
Website: https://www.nagios.org
Reading configuration data...
  Read main config file okay...
  Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
  Checked 9 services.
  Checked 2 hosts.
  Checked 1 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 24 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
```

Start Nagios and Apache:

- sudo systemctl enable apache2
- sudo systemctl enable nagios
- sudo systemctl start apache2
- sudo systemctl start nagios



A screenshot of the Nagios Core landing page. The URL in the address bar is 44.212.228.250/nagios/. The page features the Nagios logo and the text "Nagios® Core™ Version 4.4.11 April 14, 2023 Check for updates". A blue banner at the top right says "A new version of Nagios Core is available! Visit nagios.org to download Nagios 4.5.6." On the left, there's a navigation menu with sections like General, Current Status, Problems, and Reports. The "Current Status" section is active. On the right, there are "Get Started" and "Quick Links" boxes.

A screenshot of the "Current Network Status" page. The URL in the address bar is 44.212.228.250/nagios/. The page displays "Host Status Totals" and "Service Status Totals" with counts for OK, Warning, Unknown, Critical, and Pending. Below this is a table titled "Host Status Details For All Host Groups" showing two hosts: "localhost" and "your_app_name", both marked as UP. The table includes columns for Host, Status, Last Check, Duration, and Status Information. A message at the bottom says "Results 1 - 2 of 2 Matching Hosts". The left sidebar shows the same navigation menu as the previous page.

C] Verify the pipeline by triggering a build and checking the monitoring status in Nagios.

Find or create a contact definition to specify who will receive notifications.

- sudo nano /usr/local/nagios/etc/objects/contacts.cfg

```
GNU nano 7.2                                         /usr/local/nagios/etc/objects/contacts.cfg
# This contact definition inherits a lot of default values from the
# 'generic-contact' template which is defined elsewhere.

define contact {

    contact_name      nagiosadmin          ; Short name of user
    use               generic-contact       ; Inherit default values from generic-contact template
    alias             Nagios Admin
    service_notification_commands  notify-service-by-email ; Full name of user
    email             shravanistudy02@gmail.com ; <***** CHANGE THIS TO YOUR EMAIL ADDRESS *****
}

#####
#
```

Edit Service and Host Configurations (Interval of monitoring after every 60 mins)

- sudo nano /usr/local/nagios/etc/objects/hosts.cfg

```
GNU nano 7.2                                         /usr/local/nagios/etc/objects/hosts.cfg *
define host {
    use           linux-server          ; Inherit default values from the linux-server template
    host_name     ec2-web-app-instance ; The name of the host
    alias         EC2 Web App Instance ; A more descriptive name
    address       44.212.228.250      ; Public IP address of your EC2 instance
    max_check_attempts 3              ; Number of attempts before marking the host as down
    check_period   24x7              ; Check period (you can set specific times if needed)
    notification_interval 60          ; Time interval (in minutes) between notifications
    notification_period 24x7          ; Period during which notifications can be sent
    notification_options d,r          ; Options for notifications: d = down, r = recovery
    contacts      nagiosadmin        ; Specify the contact for notifications
}
```

Add a New Service Definition)

- sudo nano /usr/local/nagios/etc/objects/services.cfg

```
# Check Swap Usage
define service {
    use           generic-service      ; Replace with your server name
    host_name     your_server_name
    service_description Swap Usage
    check_command  check_swap!20!10   ; Warning at 20% usage, Critical at 10% usage
    notifications_enabled 1          ; Enable notifications
    contacts      nagiosadmin        ; Notify this contact on alerts
}

# Check HTTP
define service {
    use           generic-service      ; Replace with your server name
    host_name     your_server_name
    service_description HTTP
    check_command  check_http!http://your_web_app_url ; Replace with your web app URL
    notifications_enabled 1          ; Enable notifications
    contacts      nagiosadmin        ; Notify this contact on alerts
}

# Check SSH
```

(Services needed to be monitored Swap Uagae ,HTTP, SSH ,Ping)

```
# Check Swap Usage
define service {
    use          generic-service
    host_name    your_server_name      ; Replace with your server name
    service_description  Swap Usage
    check_command   check_swap!20!10    ; Warning at 20% usage, Critical at 10%
usage
    notifications_enabled 1           ; Enable notifications
    contacts        nagiosadmin       ; Notify this contact on alerts
}

# Check HTTP
define service {
    use          generic-service
    host_name    your_server_name      ; Replace with your server name
    service_description  HTTP
    check_command   check_http!http://your_web_app_url ; Replace with your web app URL
    notifications_enabled 1           ; Enable notifications
    contacts        nagiosadmin       ; Notify this contact on alerts
}

# Check SSH
define service {
    use          generic-service
    host_name    your_server_name      ; Replace with your server name
    service_description  SSH
    check_command   check_ssh         ; Default command to check SSH service
    notifications_enabled 1           ; Enable notifications
    contacts        nagiosadmin       ; Notify this contact on alerts
}

# Check Ping
define service {
    use          generic-service
    host_name    your_server_name      ; Replace with your server name
    service_description  PING
    check_command   check_ping!100.0%!500.0 ; Warning if > 100ms, Critical if > 500ms
    notifications_enabled 1           ; Enable notifications
    contacts        nagiosadmin       ; Notify this contact on alerts
}
```

```
ubuntu@ip-172-31-36-248:~$ ls /usr/local/nagios/bin/
nagios nagiosstats
ubuntu@ip-172-31-36-248:~$ ^C
ubuntu@ip-172-31-36-248:~$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.4.11
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2023-04-14
License: GPL

Website: https://www.nagios.org
Reading configuration data...
  Read main config file okay...
  Read object config files okay...

Running pre-flight check on configuration data...
```

Configure Email Notifications

Install Postfix

- sudo apt install postfix

```
ubuntu@ip-172-31-36-248:~$ sudo apt install postfix
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
postfix is already the newest version (3.8.6-1build2).
postfix set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-36-248:~$ █
```

Set it to send mail directly using SMTP relay, then configure [/etc/postfix/main.cf](#) with the following:

```
relayhost = [smtp.gmail.com]:587
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_sasl_security_options = noanonymous
smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
smtp_use_tls = yes
```

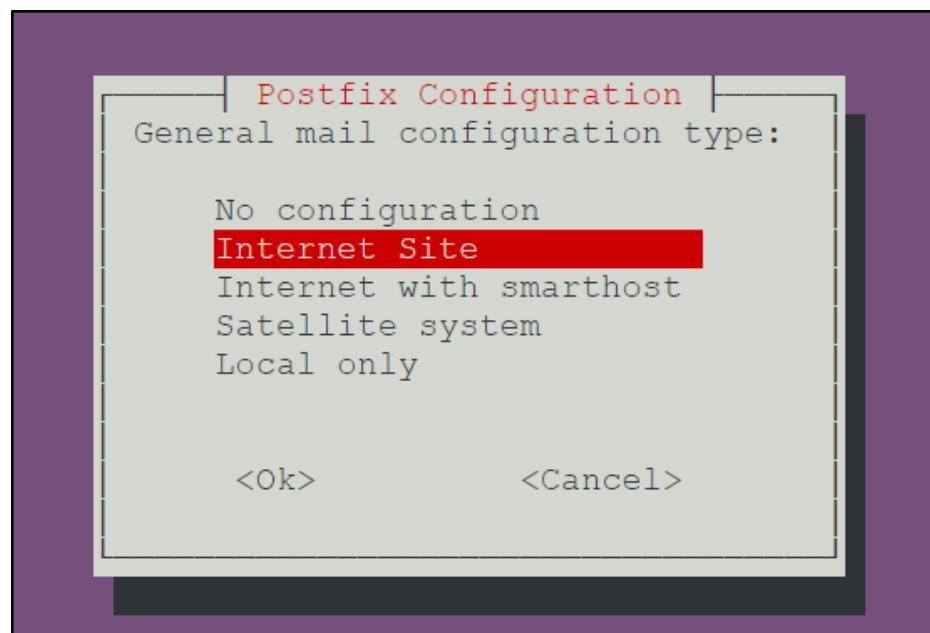
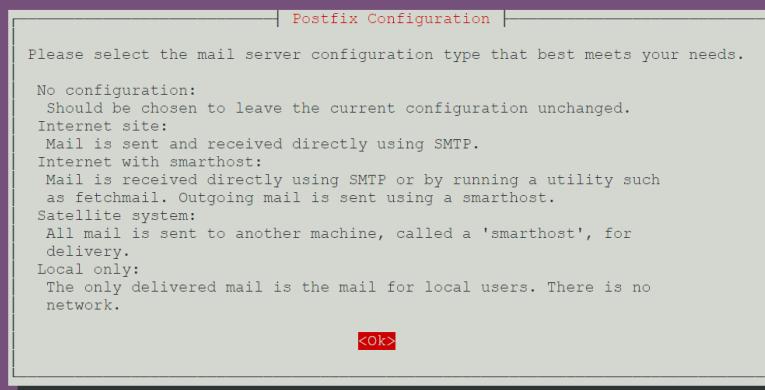
```
ubuntu@ip-172-31-36-248:~$ sudo nano /usr/local/nagios/etc/contacts.cfg
ubuntu@ip-172-31-36-248:~$ sudo nano /usr/local/nagios/etc/objects/contacts.cfg
ubuntu@ip-172-31-36-248:~$ sudo systemctl restart nagios
ubuntu@ip-172-31-36-248:~$ sudo nano /etc/postfix/main.cf
ubuntu@ip-172-31-36-248:~$ sudo nano /etc/postfix/sasl_passwd
ubuntu@ip-172-31-36-248:~$ sudo chmod 600 /etc/postfix/sasl_passwd
ubuntu@ip-172-31-36-248:~$ sudo postmap /etc/postfix/sasl_passwd
ubuntu@ip-172-31-36-248:~$ sudo systemctl restart postfix
ubuntu@ip-172-31-36-248:~$ echo "Test email from Nagios" | mail -s "Test Subject" shravanistudy021@gmail.com
ubuntu@ip-172-31-36-248:~$ sudo tail -f /var/log/mail.log
2024-10-21T16:42:22.357496+00:00 ip-172-31-36-248 postfix/qmgr[6181]: 3528080522: from=<ubuntu@ip-172-31-36-24
2024-10-21T16:42:22.591843+00:00 ip-172-31-36-248 postfix/smtp[6186]: 3528080522: SASL authentication failed;
aid: 535-5.7.8 Username and Password not accepted. For more information, go to?535 5.7.8 https://support.google.com
052e-460d3c62f4dsm19593181cf.28 - gsmtp
2024-10-21T16:42:22.592511+00:00 ip-172-31-36-248 postfix/smtp[6186]: connect to smtp.gmail.com[2607:f8b0:4004
```

```
sudo postmap /etc/postfix/sasl_passwd
```

```
sudo systemctl restart postfix
```

```
ubuntu@ip-172-31-36-248:~$ sudo postmap /etc/postfix/sasl_passwd
sudo systemctl restart postfix
ubuntu@ip-172-31-36-248:~$ sudo systemctl restart postfix
ubuntu@ip-172-31-36-248:~$ sudo postmap /etc/postfix/sasl_passwd
ubuntu@ip-172-31-36-248:~$ sudo nano /etc/postfix/sasl_passwd
ubuntu@ip-172-31-36-248:~$ sudo postmap /etc/postfix/sasl_passwd
ubuntu@ip-172-31-36-248:~$ sudo systemctl restart postfix
ubuntu@ip-172-31-36-248:~$ echo "Test email from Nagios-This is Shravani" | mail -s "Test Subject" shravanistudy02@gmail.com
ubuntu@ip-172-31-36-248:~$ sudo nano /usr/local/nagios/etc/objects/contacts.cfg
ubuntu@ip-172-31-36-248:~$ sudo systemctl restart nagios
sudo systemctl restart postfix
ubuntu@ip-172-31-36-248:~$ sudo systemctl restart nagios
ubuntu@ip-172-31-36-248:~$ sudo systemctl restart postfix
ubuntu@ip-172-31-36-248:~$ sudo nano /usr/local/nagios/etc/nagios.cfg
ubuntu@ip-172-31-36-248:~$ sudo nano /usr/local/nagios/etc/objects/hosts.cfg
ubuntu@ip-172-31-36-248:~$ sudo tail -f /var/log/mail.log
2024-10-21T17:44:18.371232+00:00 ip-172-31-36-248 postfix/qmgr[8227]: 586CA803F2: from=<nagios@ip-172-31-36-248>, size=707, nra
2024-10-21T17:44:18.982527+00:00 ip-172-31-36-248 postfix/smtp[8475]: 586CA803F2: to=<shravanistudy02@gmail.com>, relay=smtp.g
]:587, delay=0.63, delays=0.02/0.01/0.2/0.4, dsn=2.0.0, status=sent (250 2.0.0 OK 1729532658 6a1803df08f44-6ce008acf3bsm19730
2024-10-21T17:44:18.982851+00:00 ip-172-31-36-248 postfix/qmgr[8227]: 586CA803F2: removed
```

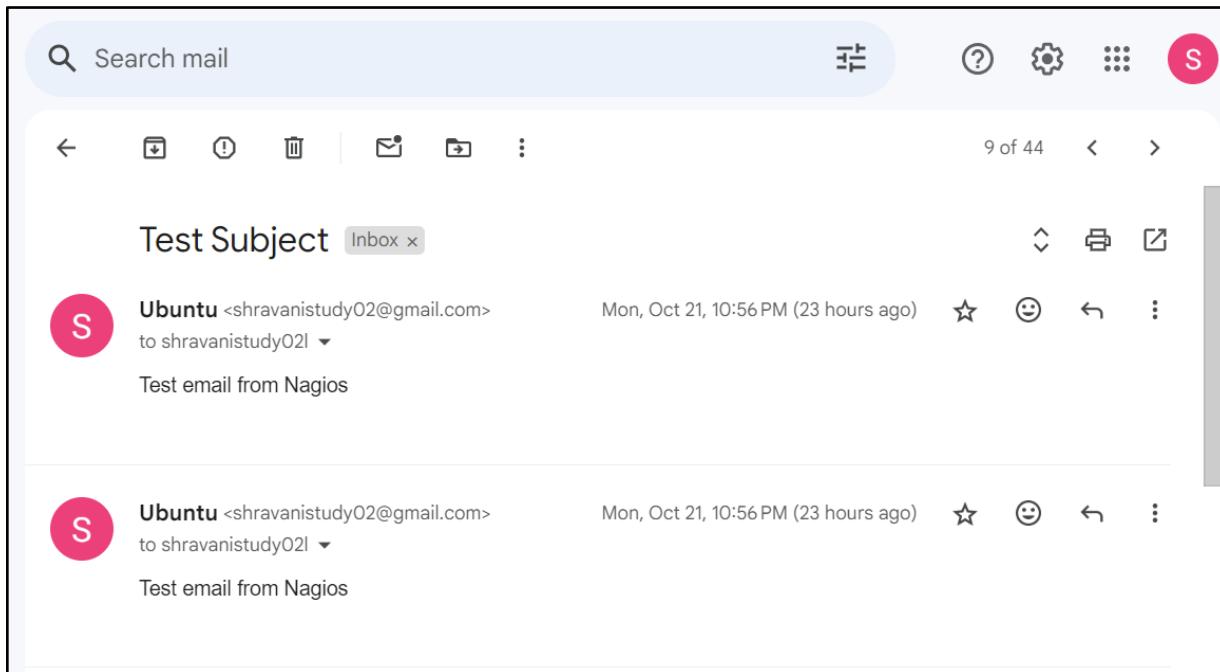
Package configuration



Test Email Notifications

- Trigger an alert and check the email sent to shravani0212@gmail.com.

```
ubuntu@ip-172-31-36-248:~$ ^C
ubuntu@ip-172-31-36-248:~$ sudo postmap /etc/postfix/sasl_passwd
sudo systemctl restart postfix
ubuntu@ip-172-31-36-248:~$ sudo systemctl restart postfix
ubuntu@ip-172-31-36-248:~$ sudo nano /etc/postfix/sasl_passwd
ubuntu@ip-172-31-36-248:~$ sudo postmap /etc/postfix/sasl_passwd
ubuntu@ip-172-31-36-248:~$ sudo systemctl restart postfix
ubuntu@ip-172-31-36-248:~$ echo "Test email from Nagios-This is Shravani" | mail -s "Test Subject" shravanistudy02@gmail.com
ubuntu@ip-172-31-36-248:~$
```



Website Monitoring Interval:

- Nagios will monitor the website every 60 minutes to check its status.

Critical Service Notification:

- If any service becomes critical during the monitoring, Nagios will immediately send a notification email to the configured contact group.

Recovery Notification:

- When the critical service is restored, Nagios will automatically send a recovery email to notify that the service is back to normal.

Service Details:

- The service being monitored is the website, with the check performed via the HTTP protocol.

Swap Usage is Critical

Current Network Status

Last Updated: Tue Oct 22 16:44:26 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.4.11 - www.nagios.org
Logged in as nagiosadmin

Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0

All Problems All Types

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
8	0	0	1	0

All Problems All Types

Service Status Details For All Hosts

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	10-22-2024 16:42:27	2d 0h 6m 52s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	10-22-2024 16:43:34	2d 0h 6m 14s	1/4	USERS OK - 1 users currently logged in
	HTTP	OK	10-22-2024 16:39:41	2d 0h 5m 37s	1/4	HTTP OK: HTTP/1.1 200 OK - 7939 bytes in 0.000 second response time
	PING	OK	10-22-2024 16:40:47	2d 0h 4m 59s	1/4	PING OK - Packet loss = 0%, RTA = 0.03 ms
	Root Partition	OK	10-22-2024 16:42:35	2d 0h 4m 22s	1/4	DISK OK - free space: / 3153 MiB (46.00% inode=87%)
	SSH	OK	10-22-2024 16:43:01	2d 0h 3m 44s	1/4	SSH OK - OpenSSH_9_6p1 Ubuntu-3ubuntu13.5 (protocol 2.0)
	Swap Usage	CRITICAL	10-22-2024 16:42:07	0d 0h 2m 19s	4/4	SWAP CRITICAL - 0% free MB out of 0 MB) - Swap is either disabled, not present or of zero size.
	Total Processes	OK	10-22-2024 16:40:14	2d 0h 2m 29s	1/4	PROCS OK: 45 processes with STATE = RSZDT
your_app_name	HTTP	OK	10-22-2024 16:41:21	1d 1h 57m 40s	1/5	HTTP OK: HTTP/1.1 200 OK - 7939 bytes in 0.003 second

Page Tour

Nagios will monitor the service after every 60 minutes and notify via mail if service is still critical

The screenshot shows a Gmail inbox with the following notifications:

- ** PROBLEM Service Alert: localhost/Swap Usage is CRITICAL ** - ***** Nagios *****... 3:14 AM
- ** PROBLEM Service Alert: localhost/Swap Usage is CRITICAL ** - ***** Nagios *****... 2:14 AM
- ** PROBLEM Service Alert: localhost/Swap Usage is CRITICAL ** - ***** Nagios *****... 1:14 AM
- ** PROBLEM Service Alert: localhost/Swap Usage is CRITICAL ** - ***** Nagios *****... 12:14 AM
- ** PROBLEM Service Alert: localhost/Swap Usage is CRITICAL ** - ***** Nagios *****... Oct 21

The screenshot shows an email message from Nagios:

**** PROBLEM Service Alert: localhost/Swap Usage is CRITICAL ****

From: shravanistudy02@gmail.com
To: me

***** Nagios *****

Notification Type: PROBLEM

Service: Swap Usage
Host: localhost
Address: 127.0.0.1
State: CRITICAL

Date/Time: Mon Oct 21 18:44:18 UTC 2024

Additional Info:

SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.

Fixing the Swap Usage Service

```
ubuntu@ip-172-31-36-248:~$ sudo swapon --show
ubuntu@ip-172-31-36-248:~$ sudo fallocate -l 1G /swapfile
ubuntu@ip-172-31-36-248:~$ sudo chmod 600 /swapfile
ubuntu@ip-172-31-36-248:~$ sudo mkswap /swapfile
Setting up swapspace version 1, size = 1024 MiB (1073737728 bytes)
no label, UUID=08e2608c-00d2-4ae5-99cd-2a79115486c7
ubuntu@ip-172-31-36-248:~$ echo '/swapfile none swap sw 0 0' | sudo tee -a /etc/fstab
/swapfile none swap sw 0 0
ubuntu@ip-172-31-36-248:~$ free -h
      total        used        free     shared  buff/cache   available
Mem:    957Mi       375Mi      286Mi      3.4Mi      451Mi      581Mi
Swap:      0B         0B         0B
ubuntu@ip-172-31-36-248:~$ sudo swapon --show
ubuntu@ip-172-31-36-248:~$
```

Current Network Status

Last Updated: Tue Oct 22 16:32:31 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.4.11 - www.nagios.org
Logged in as **nagiosadmin**

Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
9	0	0	0	0

Service Status Details For All Hosts

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	10-22-2024 16:32:27	1d 23h 54m 57s	1/4	OK - load average: 0.00, 0.00, 0.00
localhost	Current Users	OK	10-22-2024 16:28:34	1d 23h 54m 19s	1/4	USERS OK - 1 users currently logged in
localhost	HTTP	OK	10-22-2024 16:29:41	1d 23h 53m 42s	1/4	HTTP OK: HTTP/1.1 200 OK - 7939 bytes in 0.000 second response time
localhost	PING	OK	10-22-2024 16:30:47	1d 23h 53m 4s	1/4	PING OK - Packet loss = 0%, RTA = 0.03 ms
localhost	Root Partition	OK	10-22-2024 16:27:35	1d 23h 52m 27s	1/4	DISK OK - free space: / 3153 MiB (46.00% inode=87%):
localhost	SSH	OK	10-22-2024 16:28:01	1d 23h 51m 49s	1/4	SSH OK - OpenSSH_9.6p1 Ubuntu-3ubuntu13.5 (protocol 2.0)
localhost	Swap Usage	OK	10-22-2024 16:29:07	0d 2h 36m 54s	1/4	SWAP OK - 100% free (1023 MB out of 1023 MB)
localhost	Total Processes	OK	10-22-2024 16:30:14	1d 23h 50m 34s	1/4	PROCS OK: 45 processes with STATE = RSZDT
your_app_name	HTTP	OK	10-22-2024 16:31:21	1d 1h 45m 45s	1/5	HTTP OK: HTTP/1.1 200 OK - 7939 bytes in 0.002 second response time

Recovery Mail - Swap Usage OK

The screenshot shows an email inbox with a single message from 'shravanistudy02@gmail.com' titled '** RECOVERY Service Alert: localhost/Swap Usage is OK **'. The message was sent at 7:25 PM (2 hours ago) and includes the following content:

***** Nagios *****

Notification Type: RECOVERY

Service: Swap Usage
Host: localhost
Address: 127.0.0.1
State: OK

Date/Time: Tue Oct 22 13:55:37 UTC 2024

Additional Info:

SWAP OK - 100% free (1023 MB out of 1023 MB)

CONCLUSION

In conclusion, the **Automated Deployment with Monitoring** case study has effectively demonstrated the integration of CI/CD practices using Jenkins and real-time monitoring through Nagios for a simple web application. This implementation has significantly streamlined the deployment process, reducing manual intervention and minimizing the risk of errors. By automating deployments and leveraging Nagios for proactive monitoring, the system ensures high availability and reliability, promptly alerting the development team to any critical issues. This approach not only enhances the overall efficiency of application management but also sets a strong foundation for future enhancements, illustrating the vital role of automation and monitoring in modern software development practices.

