CYBER SECURITY

PRACTICAL GUIDE

Name: Chandana Somanath Khatavakar

Project Title: Monitoring System using Nagios

1. Introduction:

Monitoring is essential to ensure that IT infrastructure runs smoothly and reliably. For businesses, even a small downtime or failure in servers, applications, or network services can lead to productivity loss, missed opportunities, and a poor user experience. Monitoring helps detect issues before they affect end users and provides insights to proactively resolve them. By consistently tracking system health and performance, organizations can reduce downtime, respond faster to incidents, and improve overall security. Monitoring also enables better resource allocation, cost management, and compliance with industry standards.

Nagios was chosen because it is a powerful and flexible open-source monitoring tool that covers a wide range of IT assets, from network devices to applications and databases. It is well-known for its extensive plugin support, allowing monitoring of almost any service or application. With Nagios, alerts are triggered when thresholds are breached, ensuring that administrators are notified instantly of potential issues. It also offers a highly customizable dashboard and reporting tools, giving organizations the ability to create a tailored monitoring solution. Nagios' community and support resources make it an accessible choice, enabling continuous improvement through regular updates and an active user base. Its flexibility, robustness, and wide adoption make Nagios an ideal choice for reliable, real-time monitoring in both small and large IT environments.

2. Scope:

The scope of our monitoring includes servers, network devices, applications, and essential services. Monitoring servers is crucial to track CPU usage, memory, disk space, and overall system health. By observing these metrics, we can identify and prevent performance issues, ensure high availability, and plan for future capacity needs.

For network devices like routers, switches, and firewalls, monitoring ensures reliable data flow, tracks bandwidth usage, and detects connectivity issues. This helps maintain network stability and security, essential for smooth communication and data transfer.

Applications are also closely monitored to track their uptime, response time, and error rates. This ensures that business-critical applications are functioning properly and efficiently, providing users with a seamless experience. Monitoring services such as databases, web servers, and email servers is also key to prevent disruptions and ensure essential functions are always available to users.

Together, these elements allow us to have a holistic view of the IT environment, enabling quick detection and resolution of issues across the infrastructure. This proactive monitoring minimizes downtime and supports continuous, stable operations.

3. Overview of Nagios:

Nagios is an open-source monitoring tool widely used for tracking the health, performance, and availability of IT infrastructure. It allows organizations to monitor servers, network devices, applications, and services to ensure they operate smoothly and detect issues before they impact users. With Nagios, administrators can set up customizable alerts and notifications that trigger when systems go down, reach critical resource thresholds, or experience performance problems.

Nagios is known for its flexibility, as it supports a vast range of plugins, enabling it to monitor various technologies and protocols. It features a user-friendly dashboard for real-time monitoring and detailed reporting, helping teams analyze historical data and make informed decisions about capacity planning and system improvements. With an active community and comprehensive documentation, Nagios is suitable for both small and large-scale deployments, making it a powerful, scalable solution for maintaining a stable IT environment.

4. Features of Nagios:

1. **Plugin Architecture**: Nagios supports a wide range of plugins, allowing it to monitor virtually any service, device, or application.
2. **Alerting System**: It provides instant notifications via email, SMS, or other channels, ensuring that administrators are informed of issues quickly.
3. **Extensibility**: With its open-source nature, Nagios allows for extensive customization, including custom plugins and scripts.
4. **Multi-Platform Support**: Nagios works across various platforms, making it versatile for different operating systems and devices.
5. **Scalability**: Suitable for monitoring small environments to large networks with thousands of nodes.
6. **Dashboard Visualization**: A user-friendly interface presents real-time data and historical performance trends in a centralized dashboard.
7. **Detailed Reporting**: Nagios generates reports for availability, performance, and SLA compliance, supporting analysis and auditing.
8. **Event Handler Support**: Enables automated responses to specific events, reducing the need for manual intervention.
9. **Community Support**: A large, active user community and a wealth of resources contribute to its continuous development.

5. Benefits of Nagios:

1. **Proactive Monitoring**: Detects issues before they affect end users, minimizing downtime and improving reliability.
2. **Custom Alert Configurations**: Set custom alert thresholds and notification options for each monitored component, ensuring timely response to critical incidents.
3. **Dashboard Visualization**: Provides a comprehensive view of the IT infrastructure's status, helping teams track performance and identify trends.
4. **Improved Efficiency**: Automated alerts and event handling reduce manual workload, allowing IT teams to focus on other tasks.
5. **Scalability for Growth**: Easily scalable as the infrastructure grows, making it suitable for both small and large environments.
6. **Enhanced Security**: Continuous monitoring helps identify unusual activity or potential security threats.
7. **Cost-Effectiveness**: Open-source solution with no licensing costs, reducing the overall expense of IT management.

5. Environment Setup:

Software Requirements:

1. **Operating System**:
    - **Ubuntu**: Version 20.04 LTS, 22.04 LTS, or higher
    - **Windows**: Windows Server 2016, 2019, or 2022 for agent-based monitoring (Nagios agents like NSClient++ can be installed on Windows systems for monitoring)
2. **Nagios Version**:
    - **Nagios Core**: Version 4.4.6 or higher for open-source setups
    - **Nagios XI**: For an enterprise-level, more user-friendly experience (requires a subscription)

Network Requirements:

1. **Firewall Settings**: Open relevant ports on both the Nagios server and client machines:
    - **Nagios Plugins and NRPE**: Port 5666 (for Linux clients using NRPE)
    - **NSClient++ on Windows**: Port 12489 (for check_nt) or 5666 (for NRPE)
2. **IP Whitelisting**: Ensure that both the Nagios server and client IPs are allowed in each other's configuration files and firewalls for communication.
3. **Network Stability**: Use a stable network connection to reduce the chance of missed checks and inaccurate alerts.

Installation of nagios on ubantu

Step-by-Step Guide with Explanations

1. **System Update and Initial Package Installation**

```
sudo apt update
sudo apt upgrade
```

Explanation: Updates your package lists and upgrades installed packages,

ensuring the latest versions and security updates.

1. **Installing Required Packages**

```
sudo apt-get install -y autoconf gcc libc6 make wget unzip apache2 php
libapache2-mod-php libgd-dev build-essential
sudo apt-get install openssl libssl-dev
```

Explanation: Installs dependencies required to build and run Nagios, including Apache for the web interface, PHP, and development libraries.

2. **Downloading and Preparing Nagios Core**

```
mkdir tmp
cd tmp
wget-Onagioscore.tar.gz
https://github.com/NagiosEnterprises/nagioscore/archive/nagios-
4.4.14.tar.gz
tar xzf nagioscore.tar.gz
ls
```

Explanation: Creates a temporary directory, downloads Nagios Core, and extracts it.

3. **Configuring and Compiling Nagios Core**

```
cd nagios-4.4.14
sudo ./configure --with-httpd-conf=/etc/apache2/sites-enabled
sudo make all
```

Explanation: Configures Nagios to work with Apache, prepares it for compilation, and compiles the code.
Image: Screenshot of the terminal while running the. /configure and make commands.

4. **Setting Up Users and Installing Nagios Components**

```
sudo make install-groups-users
sudo usermod -a -G nagios www-data
sudo make install
sudo make install-daemoninit
sudo make install-commandmode
sudo make install-config
sudo make install-webconf
```

Explanation: Creates Nagios user groups, modifies permissions, and installs various Nagios components, including configuration and web server integration.

Image: Screenshot of each make install step for clarity.

5. **Enabling Apache Modules and Firewall Configuration**

```
sudo a2enmod rewrite
sudo a2enmod cgi
sudo ufw allow Apache
sudo ufw reload
```

Explanation: Enables Apache modules required for Nagios and adjusts firewall settings to allow web traffic. *Image:* Firewall rules configuration in terminal or confirmation message.

6. **Setting Up Nagios Web Interface Login**

```
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

Explanation: Creates a user (nagiosadmin) for logging into the Nagios web interface and prompts for a password. *Image:* Password prompt during user creation.

7. **Starting the Nagios Service**

```
sudo systemctl start nagios.service
```

Explanation: Starts the Nagios service, making it active for monitoring.

Image: Status check for the Nagios service using systemctl status nagios.

8. **Testing the Nagios Installation**

Open your web browser and enter http://<your_server_ip>/nagios *Explanation:* Accesses the Nagios web interface for initial testing. *Image:* Screenshot of the Nagios web interface login page.

Configuring NCPA Agent on Windows and Ubuntu

9. **Installing NCPA on Windows**

Explanation: Download and install the NCPA agent on Windows, then test using https://localhost:5693/login.

Image: Screenshot of the NCPA web interface on Windows.

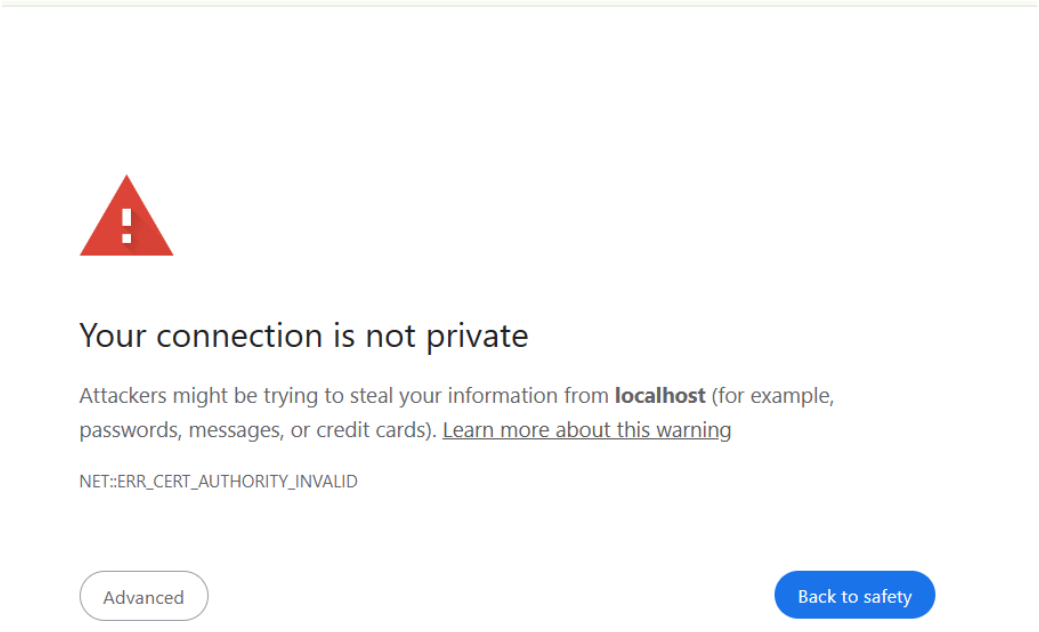


Image 1
Click on Advanced

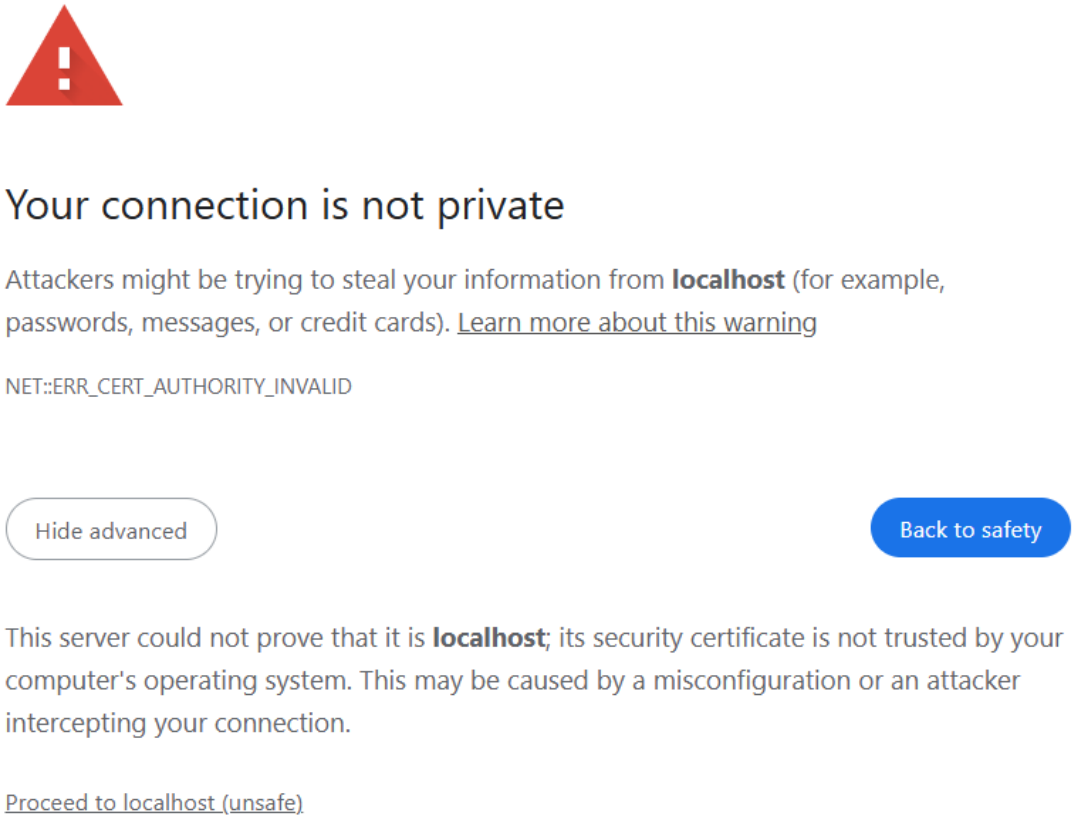


Image 2

Click on proceed to localhost(unsafe)

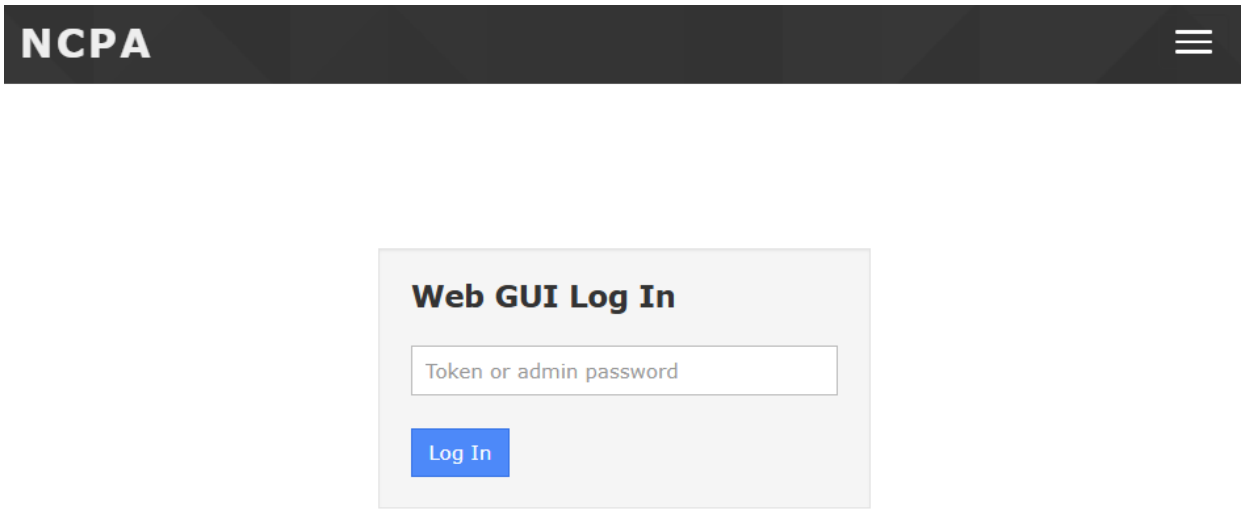


Image 3

Give password as mytoken and click on log In

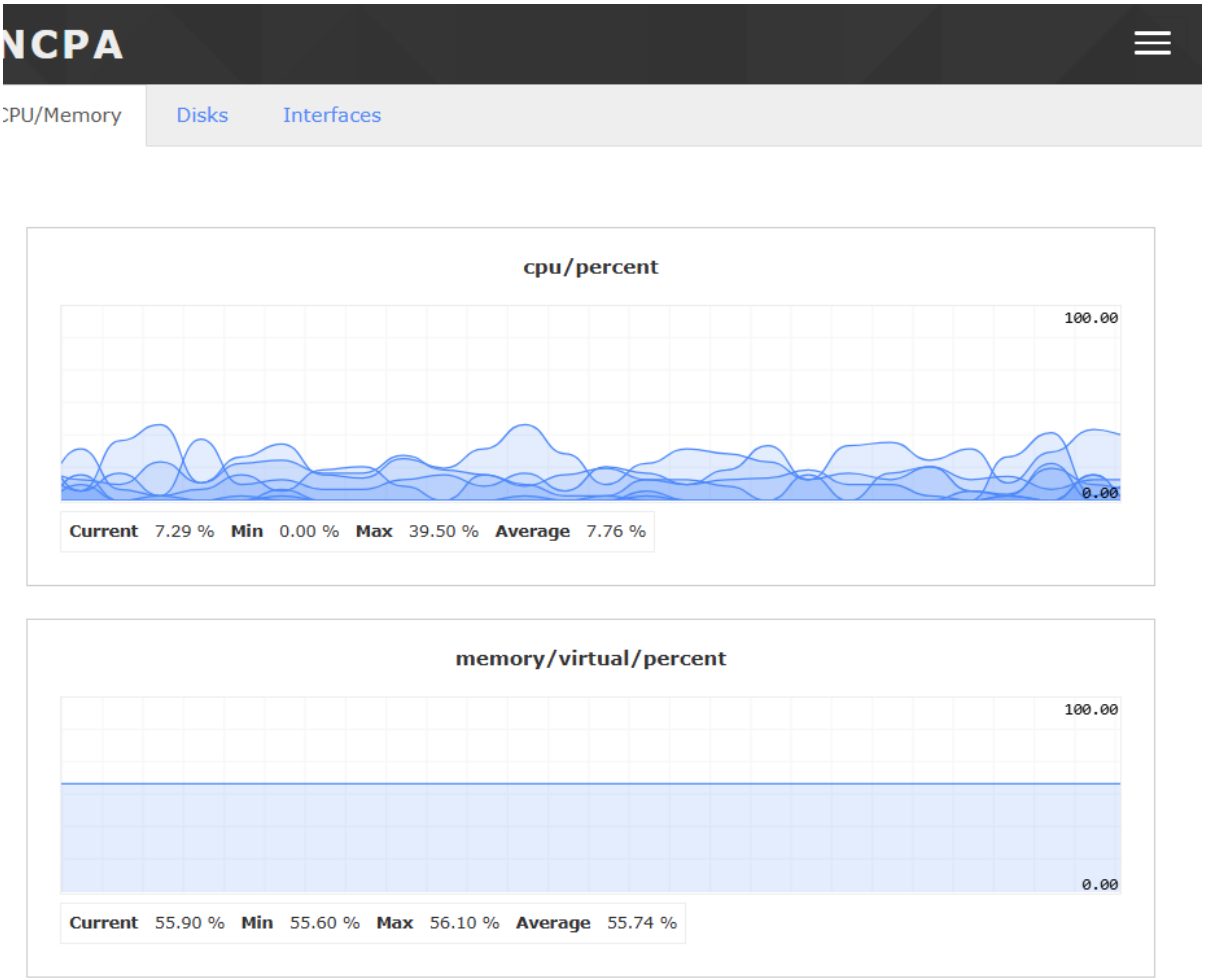


Image 4

Setting Up NCPA on Ubuntu

10. **Download NCPA and copy the check_ncpa.py plugin to Nagios.**

```
sudo cp Downloads/check_ncpa/check_ncpa.py /usr/local/nagios/libexec/
```

Explanation: Configures the NCPA plugin on Ubuntu so Nagios can use it to monitor remote clients.

Image: Screenshot of plugin location.

A. Configuring Hosts and Services in Nagios

11. **Defining a Windows Host and Host Group**

Use sudo nano windows11.cfg to define a host with name Chandu.

```
define host {
use windows-server
host_name Chandu
alias Chandu
address 192.168.10.11
check_command check_ping!100.0,20%!500.0,50%
}
define hostgroup{
hostgroup_name windows-servers
alias windows servers
members Chandu
}
```

Explanation*:* Configures a host (a Windows server) to be monitored by Nagios.

Image: Configuration file with host details.

12. **Testing the Configuration**

```
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

Explanation: Validates Nagios configuration files for errors.

Image: Output showing successful configuration validation.

13. **Defining Services for Monitoring**

Open chandu.cfg to define various services, such as PING and disk usage.

```
define service {
use     local-service
host_name Chandu
service_description     PING
check_command   check_ping!100.0,20%!500.0,60%
}

define service {
use     local-service
host_name Chandu
service_description     ROOT Partition
check_command   check_local_disk!13%!20%!/
}

define service {
use     local-service
host_name Chandu service_description
Current Users
check_command   check_local_users!20!50
}
```

Control+O enter Control+X

Explanation*:* Configures services on the host, like checking ping response and disk usage.
Image: Screenshot of chandu.cfg with service definitions.

14. **Restarting the Nagios Service**

```
sudo systemctl restart nagios.service
```

Explanation: Restarts Nagios to apply new configurations. */*

mage: Confirmation message in terminal after successful restart.

B. Final Testing and Validation

15. **Testing Host and Service Monitoring**

Go to the Nagios web interface and check for the status of BhavaniGowda and its services.

Explanation: Verify that the new host and services are being monitored as expected.

Image: Screenshot of the Nagios web interface showing the host and service statuses.

IMAGES:

Image 5

Nagios interface

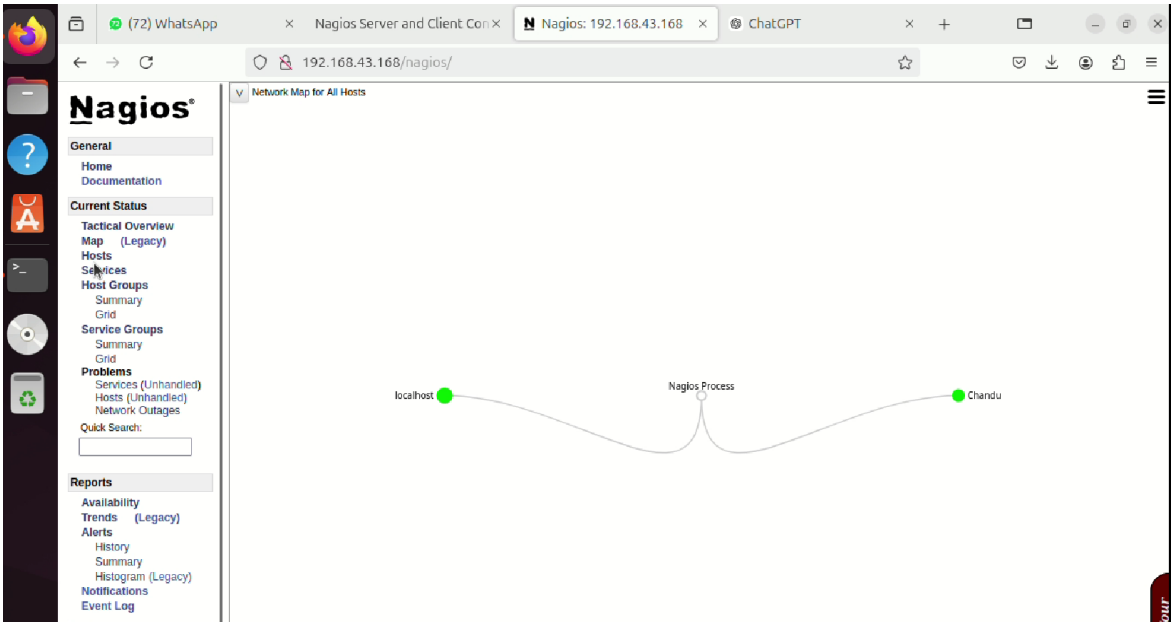


image 6

This image shows the Nagios web interface displaying a **Network Map**. Here's a breakdown of the key elements in the image:

1. **Nagios Web Interface** (Left Sidebar)
    - The left sidebar contains links to various sections within Nagios, such as:
        - **Home**: The main dashboard.
        - **Current Status**: Allows you to view the real-time status of hosts and services being monitored.
        - **Host Groups, Service Groups, and Network Map**: Organizes hosts and services into groups and displays their relationships.
        - **Reports and Systems**: Provides options for generating reports on system performance, availability, etc.
2. **Network Map (Main Display Area)**
    - This network map visualizes the hosts and connections monitored by Nagios:
        - **localhost**: Represents the Nagios server itself (running on the local machine).
        - **Nagios Process**: This node is a central hub, representing the Nagios monitoring engine.
        - **BhavaniGowda**: This is a monitored client host connected to the Nagios server.
    - The green dots indicate that both localhost and BhavaniGowda are in an "up" (healthy) state, meaning they are reachable and currently functioning as expected.

This map view is helpful for a quick overview of the monitored network and to visualize the relationships and statuses of hosts. If any host were down, it would likely show a different color (such as red) to indicate an issue.

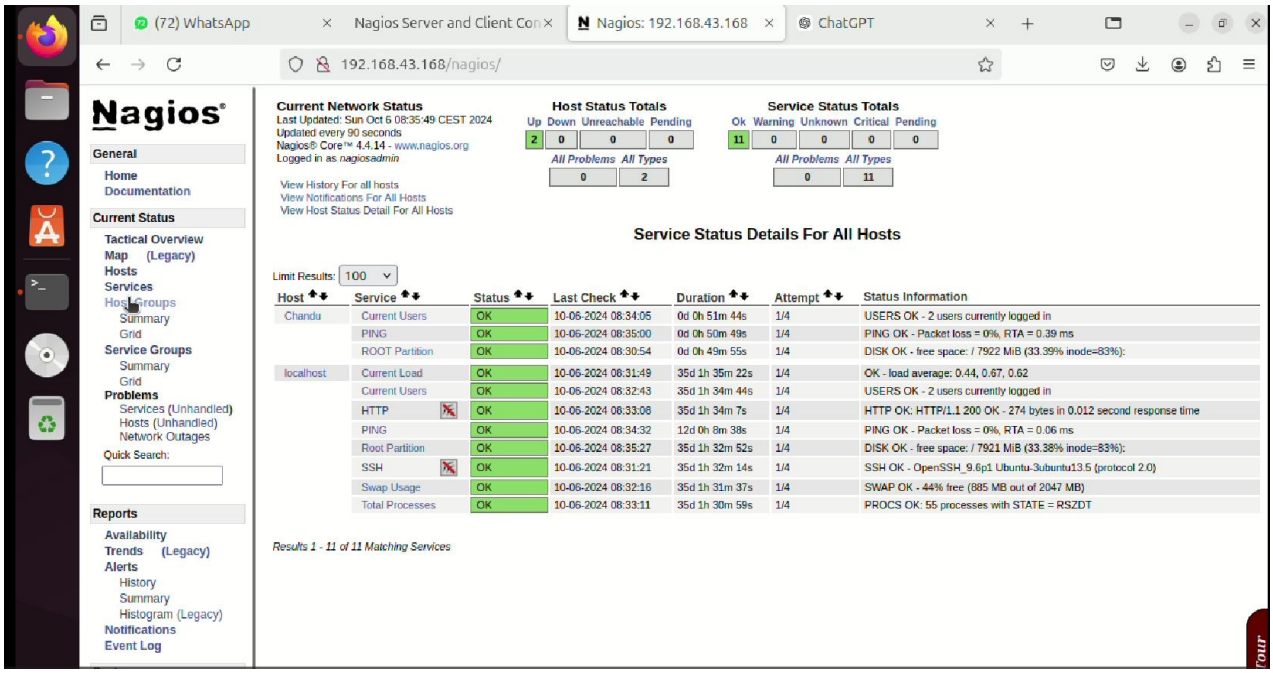


Image 7

This image shows the **Nagios Service Status Dashboard** for monitored hosts. Here's a simple breakdown:

1. **Host and Service Status Overview (Top Section)**:
    - This section gives a summary of the **Host Status Totals** and **Service Status Totals**.
    - It shows the number of hosts and services that are **up (OK)**, **down**, or **in a warning state**.
    - Currently, it indicates that all hosts and services are in the "OK" state.
2. **Service Status Details for All Hosts (Main Table)**:
    - This table lists the **services being monitored** for each host.
    - Each row shows details for a specific service on a host. For example:

    Host: Shows the name of the host, such as localhost or Chandu.

3. **Service**: Lists the specific services being monitored, like **PING**, **HTTP**, **CPU**, **Disk Usage** (Root Partition), and others.
4. **Status**: Indicates if the service is working correctly (OK) or if there's an issue. Here, all services are marked "OK" in green.
5. **Last Check**: Shows the last time Nagios checked that service.
6. **Duration**: Indicates how long the service has been in its current state.

7. **Status Information**: Provides specific details about the service's condition. For example:
    - **PING**: Shows packet loss and response time.
    - **CPU**: Indicates the number of users logged in.
    - **Disk Usage (Root Partition)**: Displays disk space usage.
    - **HTTP** and **SMTP**: Reports the service's response time.

This dashboard provides a quick overview of the health of various services across hosts. In this case, everything is functioning normally, as shown by the "OK" status for each service.

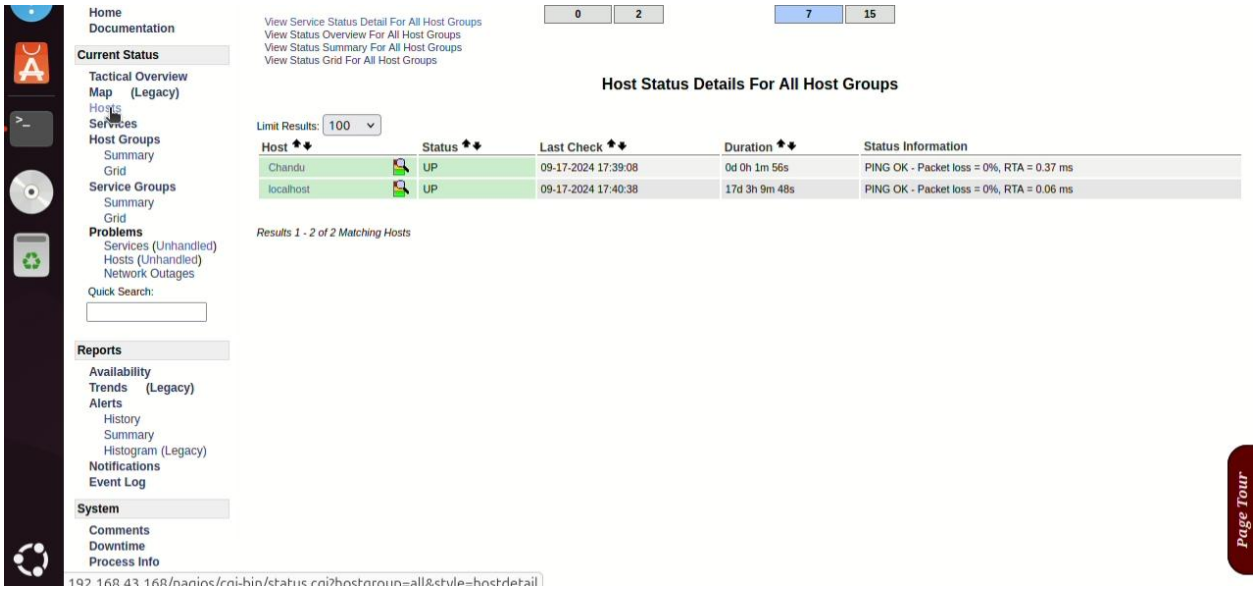


Image 8

It displays the current status of the network.

Here's a breakdown of what you see:

Top Section:

- **Current Network Status:** Shows the overall health of the network.
- **Host Status Totals:** Shows the number of hosts (devices) that are up, down, or have unknown status.
- **Service Status Totals:** Shows the number of services (like web servers, databases) that are up, down, or have unknown status.

Host Status Details:

- **Host Groups:** Lists different groups of hosts.
- **Summary:** Provides a summary of the host's status.
- **Last Check:** Shows the last time the host was checked.
- **Status:** Shows the current status of the host (up, down, unknown).
- **Duration:** Shows how long the host has been in the current state.
- **Status Information:** Provides more details about the host's status, like network latency or disk usage.

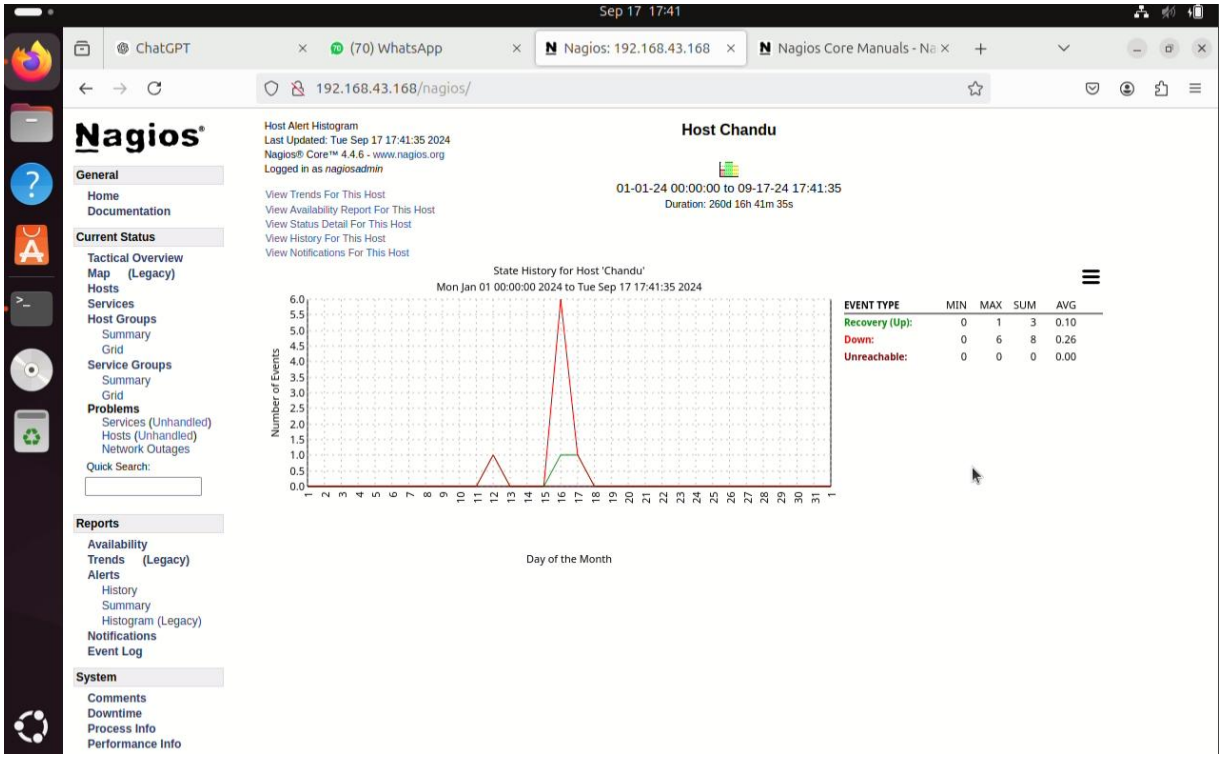


Image 9

The image shows a Nagios monitoring dashboard displaying the **Host Alert Histogram** for a host named "Chandu." This histogram provides a graphical representation of the host's status history, indicating the host's state (up, down, or unreachable) over a period.

1.**Graph:**

- The graph on the right represents the "State History for Host 'Chandu'" by day of the month.
- **Y-Axis:** Number of events.
- **X-Axis:** Days of the month.

- **Lines:**
    - **Green line** represents recovery (up) events.
    - **Red line** represents down events.

In the graph, there is a noticeable spike around the middle of the month, indicating multiple down events.

2. **Event Type Table (on the right):**

- Shows the minimum, maximum, sum, and average values for each event type:
    - **Recovery (Up):** Min 0, Max 1, Sum 3, Avg 0.10
    - **Down:** Min 0, Max 6, Sum 8, Avg 0.26
    - **Unreachable:** No events (all values are 0)

3. **Menu on the Left:**

- Various Nagios options and sections are listed in the sidebar, such as "Home," "Tactical Overview," "Hosts," "Host Groups," "Problems," and "Reports." This allows the user to navigate through different aspects of the Nagios monitoring system.

Nagios email Alert Notification

- Configure SMTP server for generating the email notification
- Here I am using Postfix Server to send an email
- Before doing this email alert notification I will tell you how to use email as a Smarthost
- SmartHost means : A **Smarthost** is an intermediary mail server that acts as a relay for outbound emails. In a Smarthost configuration, your internal email server forwards emails to the Smarthost, which then handles the process of delivering those emails to their destination (other email servers on the internet).
- The primary use of a Smarthost is to provide a reliable, authenticated channel for sending emails, particularly in environments where the internal email server may not have direct access to the internet or where email delivery management needs to be outsourced.

1. Change some parameters in your gmail account go to your Gmail and then go to App password and create one custom application software before doing this step make a sure your configured the two step authentication on your gmail.

2. create an application password write a postfix to the name and click create or generate it will generate an application password and take a screenshot of this because this password I am going to use in my next doing postfix con figuration on Ubuntu.

3. Install Postfix:

```
sudo apt install postfix
```

Explanation: Installs Postfix, a mail transfer agent, to handle email delivery on your system.

4. Edit the SASL Password File:
   ```
   sudo nano /etc/postfix/sasl/sasl_passwd
   ```

Explanation: Opens the sasl_passwd file in a text editor. Add your SMTP relay credentials in the following format:

```
[smtp.example.com]:587 username:password
```

5. Set File Permissions for sasl_passwd:
   ```
   sudo chmod 600 /etc/postfix/sasl/sasl_passwd
   ```

Explanation: Restricts access to sasl_passwd to ensure only the root user can read it, enhancing security.

6. Generate a Postfix Lookup Table for SASL:

   sudo postmap /etc/postfix/sasl/sasl_passwd

Explanation: Creates a hashed version of sasl_passwd as sasl_passwd.db, which Postfix uses for authentication.

7. Verify the Contents of the SASL Directory:

   ls -l /etc/postfix/sasl/

Explanation: Lists the contents and permissions of /etc/postfix/sasl/, allowing you to verify that both sasl_passwd and sasl_passwd.db exist and have the correct permissions.

8. Verify Permissions on the Postfix Main Configuration File:

   ls -l /etc/postfix/main.cf

Explanation: Checks the permissions of the main.cf configuration file to ensure its properly secured.

9. Backup the Postfix Configuration File:

   sudo cp -a /etc/postfix/main.cf /etc/postfix/main.cf.bak

Explanation: Creates a backup of main.cf in case you need to revert any changes.

10. Edit the Postfix Main Configuration File:

    sudo nano /etc/postfix/main.cf

Explanation: Opens the main configuration file in a text editor, where you can add settings for SASL authentication. Common settings include defining the relay host and enabling SASL.

11. Check Postfix Configuration:

    sudo postfix check

Explanation: Checks the Postfix configuration for syntax errors or other issues.

12. Restart the Postfix Service:

    sudo systemctl restart postfix.service

Explanation: Restarts Postfix to apply the new settings.

13. Send a Test Email:

```
echo "This is a test email." | mail -s "Test2" chandanask567@gmail.com
```

Explanation: Sends a test email with the subject "Test2" to verify that Postfix is working correctly.

14. View the Mail Log:

```
less /var/log/mail.log
```

Explanation: Opens the Postfix mail log for review. This log shows details of email deliveries and any errors that occurred during the process.

IMAGES

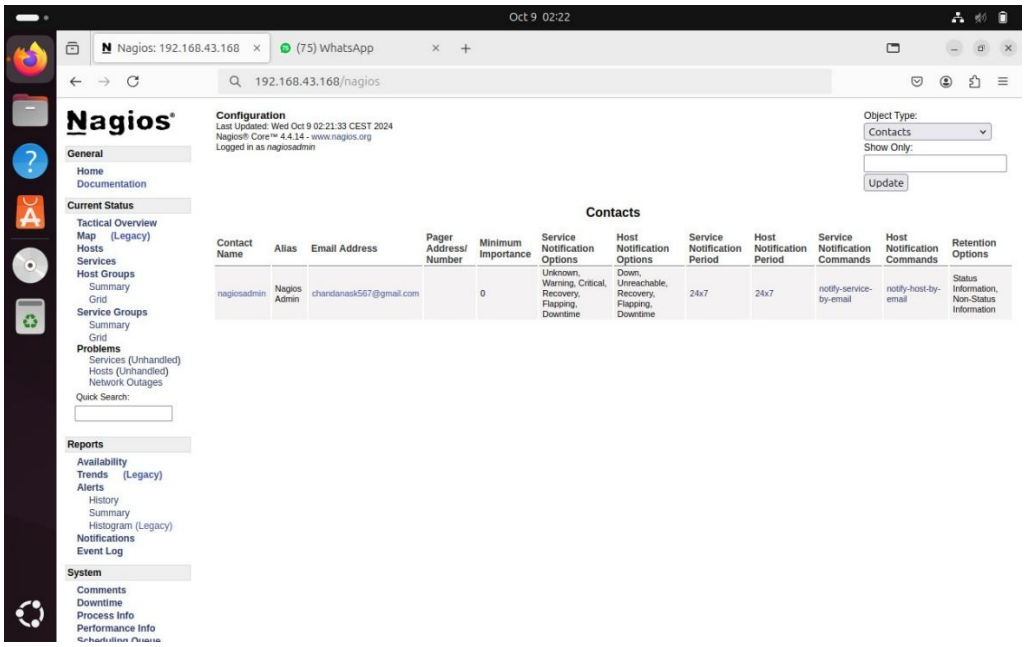


Image 10

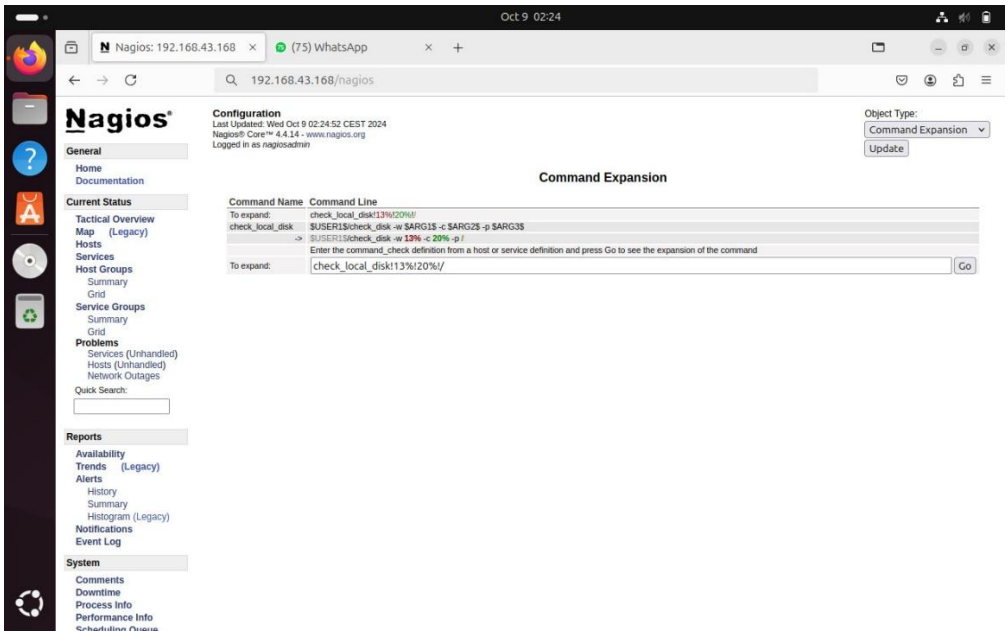


Image 11

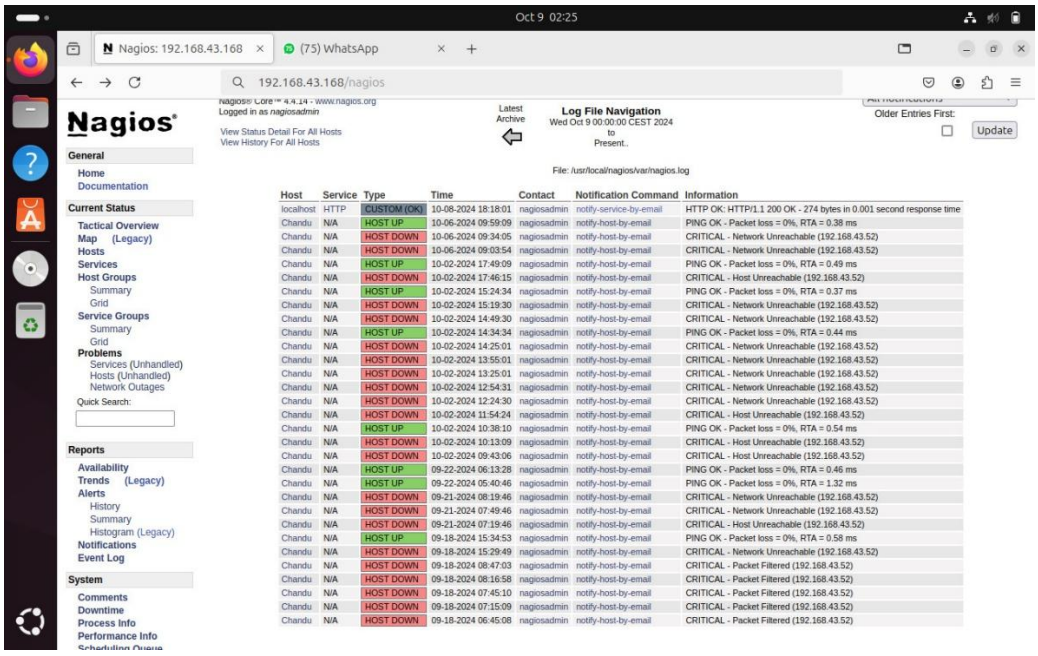


Image 12

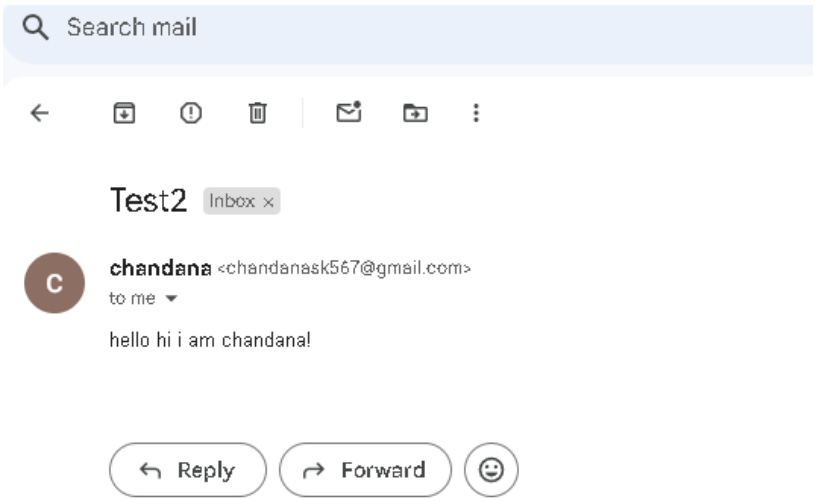


Image 13

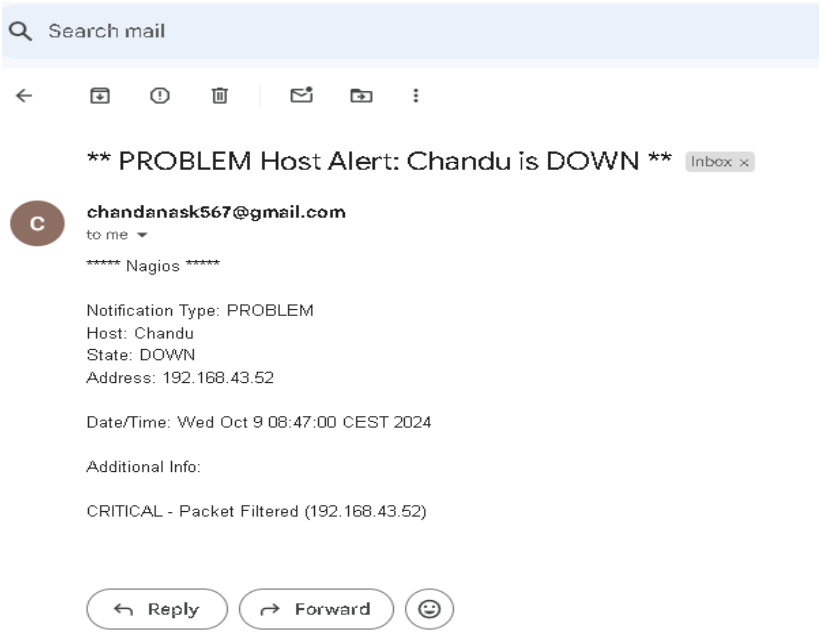
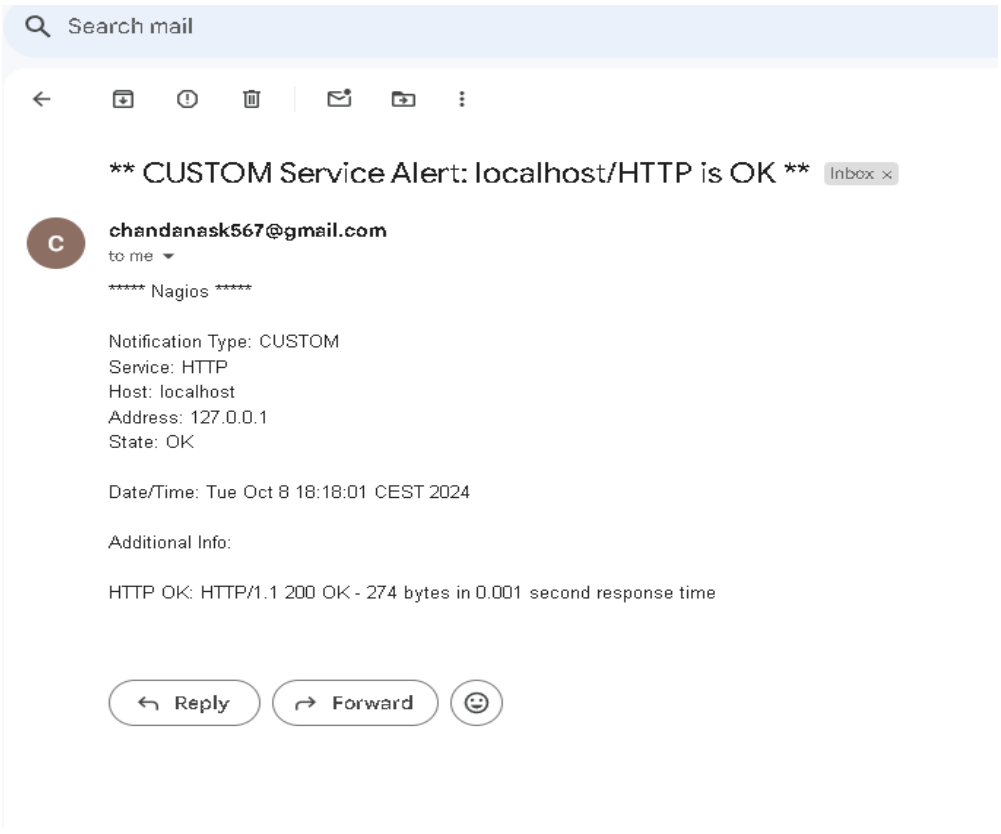


Image 14


Search mail
** CUSTOM Service Alert: localhost/HTTP is OK **
Inbox ×
C
chandanask567@gmail.com
to me
***** Nagios *****
Notification Type: CUSTOM
Service: HTTP
Host: localhost
Address: 127.0.0.1
State: OK
Date/Time: Tue Oct 8 18:18:01 CEST 2024
Additional Info:
HTTP OK: HTTP/1.1 200 OK - 274 bytes in 0.001 second response time
Reply
Forward

Challenges and Solutions

Here’s a simpler version of the challenges and solutions:

1. Network Problems (Slow Connection)

Challenge:
The network might be slow, causing delays.

Solution:

- Check for network blockages (firewall or DNS issues).
- Use tools like ping to test the network speed.
- Place servers closer to users for faster response.

2. Wrong Configuration or Software Versions

Challenge:
Different software versions might not work well together.

Solution:

- Make sure both the server and client use compatible versions.
- Use tools like Docker to manage versions easily.

3. High CPU or Memory Use

Challenge:
Too much CPU or memory usage can slow things down.

Solution:

- Use monitoring tools to check usage.
- Restart services or offload tasks to other servers if needed.

4. Permissions and Security Issues

Challenge:
Incorrect permissions can block access to files or cause security risks.

Solution:

- Ensure permissions follow the "least privilege" rule.
- Use tools to enforce security (like firewalls).
- Regularly check security logs.

5. Missing Dependencies (Required Software)

Challenge:
Some required software might be missing or incompatible.

Solution:

- Install missing software using package managers (e.g., apt or pip).
- Use virtual environments to avoid conflicts.

6. Service Failures

Challenge:
Services might crash or stop working.

Solution:

- Check logs for the cause.
- Set services to restart automatically.
- Use backup servers to reduce downtime.

7. Wrong or Missing Configuration Files

Challenge:
Incorrect or missing settings can cause problems.

Solution:

- Backup configuration files before changing them.
- Track changes using Git or other tools.
- Use tools like Ansible to manage configurations.

8. Not Monitoring the System

Challenge:
If you're not watching system health, issues might go unnoticed.

Solution:

- Set up monitoring tools (like Nagios or Prometheus).
- Set up alerts to get notified of problems early.

9. Database Connection Issues

Challenge:
The system might not be able to connect to the database.

Solution:

- Check if the database is set up to allow connections.
- Use connection pooling to manage connections more efficiently.
- Monitor database logs for errors.

10. Lack of Documentation

Challenge:
It’s hard to solve problems without clear instructions.

Solution:

- Keep good, up-to-date documentation.
- Share knowledge with the team through shared resources.

This should help address common issues in a more straightforward way!

Project Summary:

The Nagios monitoring system has been successfully implemented to track the health, performance, and availability of critical IT infrastructure, including servers, network devices, applications, and services. The system's real-time alerting and comprehensive reporting have allowed proactive issue detection and resolution, significantly reduced downtime and improving system reliability. The user-friendly dashboard and custom alert configurations have made monitoring more efficient, while the plugin architecture has provided flexibility in monitoring diverse systems. The project has demonstrated the effectiveness of Nagios in maintaining a stable and secure IT environment.

Future Enhancements:

1. **Advanced Plugins**: As the infrastructure grows, incorporating more advanced Nagios plugins for specialized monitoring (e.g., cloud services, containerized environments, or advanced databases) will further enhance its capabilities.
2. **Integration with Other Systems**: Integrating Nagios with IT Service Management (ITSM) tools, ticketing systems, or automation platforms (like Ansible or Puppet) can streamline incident management and resolution processes.
3. **Scalability for Larger Environments**: To support larger, more complex environments, consider implementing Nagios in a distributed setup, where multiple Nagios instances work together, ensuring scalability without compromising performance.
4. **Artificial Intelligence and Machine Learning**: Introducing AI-based anomaly detection could help identify patterns and predict potential failures before they occur, reducing reliance on manual configurations.
5. **Cloud and Hybrid Environments**: Expanding monitoring capabilities to include cloud platforms (e.g., AWS, Azure) and hybrid infrastructures can ensure comprehensive visibility across on-premises and cloud-based resources.
6. **Enhanced Reporting and Analytics**: Future work could include integrating more advanced reporting tools and data visualization dashboards to provide deeper insights into trends and performance over time.

References:

1. **Nagios Documentation**:
   Official Nagios documentation is a primary source for setup and configuration:
   - Nagios Core User Guide: https://nagios.org/documentation/
   - Nagios XI Documentation: https://assets.nagios.com/downloads/nagiosxi/docs/
2. **Nagios Community**:
   The Nagios community forum and mailing list were helpful for troubleshooting common issues.
   - Nagios Forum: https://support.nagios.com/forum/
   - Nagios Exchange (plugins and add-ons): https://exchange.nagios.org/
3. **Ubuntu Community**:
   Official guides and forums for installing and configuring Nagios on Ubuntu.
   - Ubuntu Wiki: https://help.ubuntu.com/community/Nagios
   - Ask Ubuntu: https://askubuntu.com/
4. **NSClient++ Documentation**:
   For installing and configuring the Nagios agent on Windows clients.
   - NSClient++ Documentation: https://nsclient.org/
5. **Nagios Tutorials and Blogs**:
   Many practical guides and tutorials were found on various websites.
   - Nagios Setup Tutorials: https://www.digitalocean.com/community/tutorials

These resources helped in setting up Nagios on both Ubuntu and Windows systems, as well as in troubleshooting any encountered issues during configuration and installation