**Advanced DevOps Lab**

**Experiment 10**

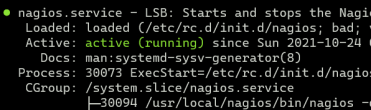
| Roll No. | 19 |
| --- | --- |
| Name | Aditya Gupta |
| Class | D15-B |
| Subject | Advanced DevOps Lab |

Expt No. 10 Advanced DevOps Lab Roll No. 24

**Aim:** To perform Port, Service monitoring, Windows/Linux server monitoring using Nagios. **Steps:**

Prerequisites: AWS Free Tier, Nagios Server running on Amazon Linux Machine.

1. To Confirm that Nagios is running **on the server side**, run this *sudo systemctl status nagios* on the “NAGIOS HOST”.

You can proceed if you get this message.

2. Before we begin,

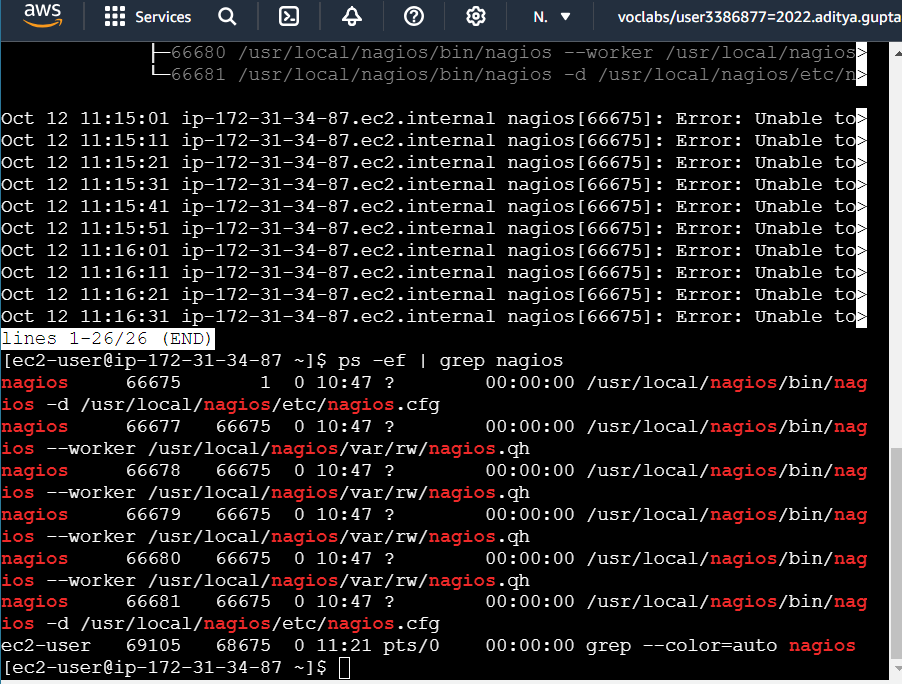
To monitor a Linux machine, create an Ubuntu 20.04 server EC2 Instance in AWS.

Provide it with the same security group as the Nagios Host and name it ‘linux-client’ alongside the host.

**For now, leave this machine as is, and go back to your nagios HOST machine**.

3. On the server, run this command

ps -ef | grep nagios



4. Become a root user and create 2 folders

Sudo su

mkdir /usr/local/nagios/etc/objects/monitorhosts

mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts

5. Copy the sample localhost.cfg file to linuxhost folder

cp /usr/local/nagios/etc/objects/localhost.cfg

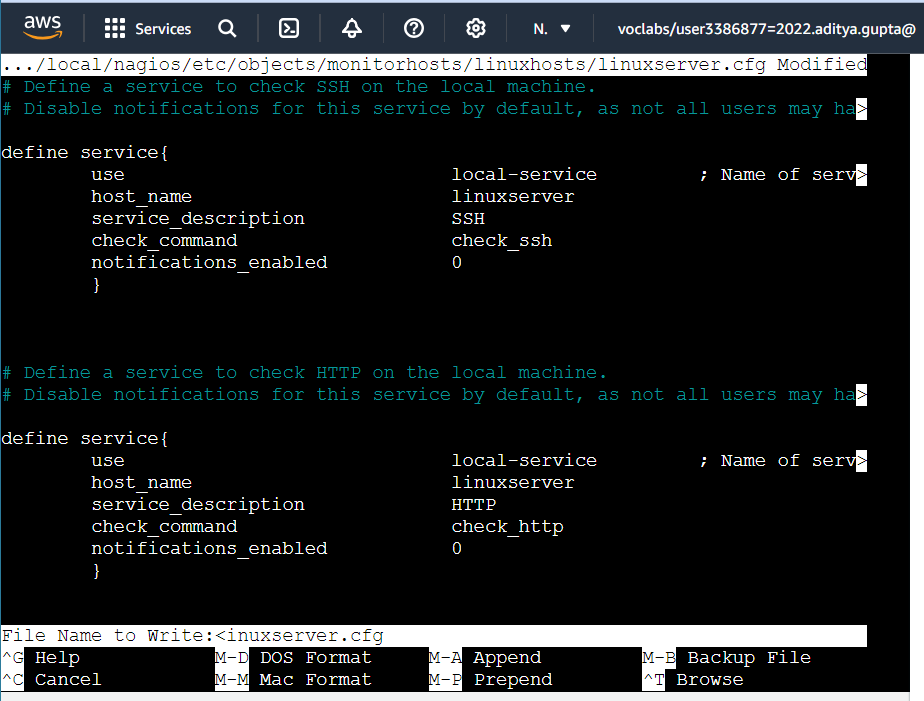
/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg 6. Open linuxserver.cfg using nano and make the following changes

nano

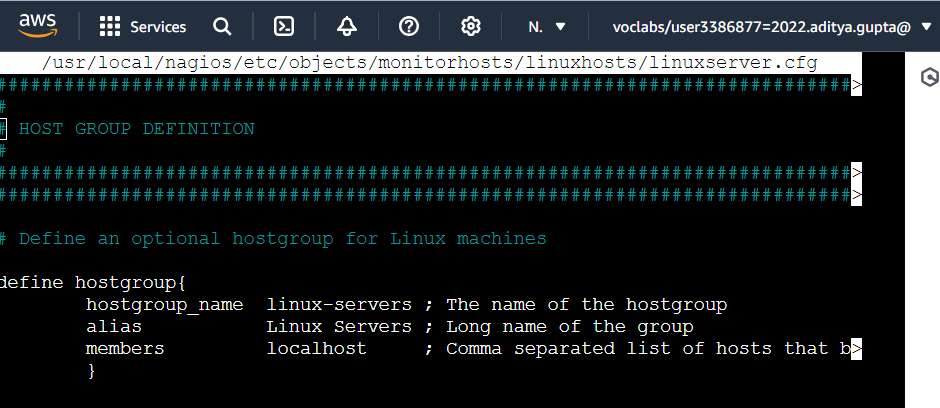
/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

Change the hostname to linuxserver (EVERYWHERE ON THE FILE)

Change address to the public IP address of your **LINUX CLIENT**.



Change hostgroup\_name under hostgroup to linux-servers1



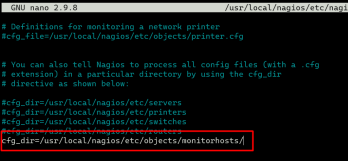
Everywhere else on the file, change the hostname to linuxserver instead of localhost.

7. Open the Nagios Config file and add the following line

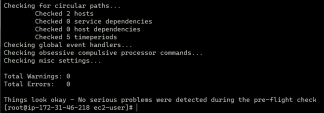
nano /usr/local/nagios/etc/nagios.cfg

##Add this line

cfg\_dir=/usr/local/nagios/etc/objects/monitorhosts/



8. Verify the configuration files

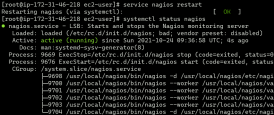


You are good to go if there are no errors.

Expt No. 10 Advanced DevOps Lab Roll No. 24

9. Restart the nagios service

service nagios restart



Now it is time to switch to the client machine.

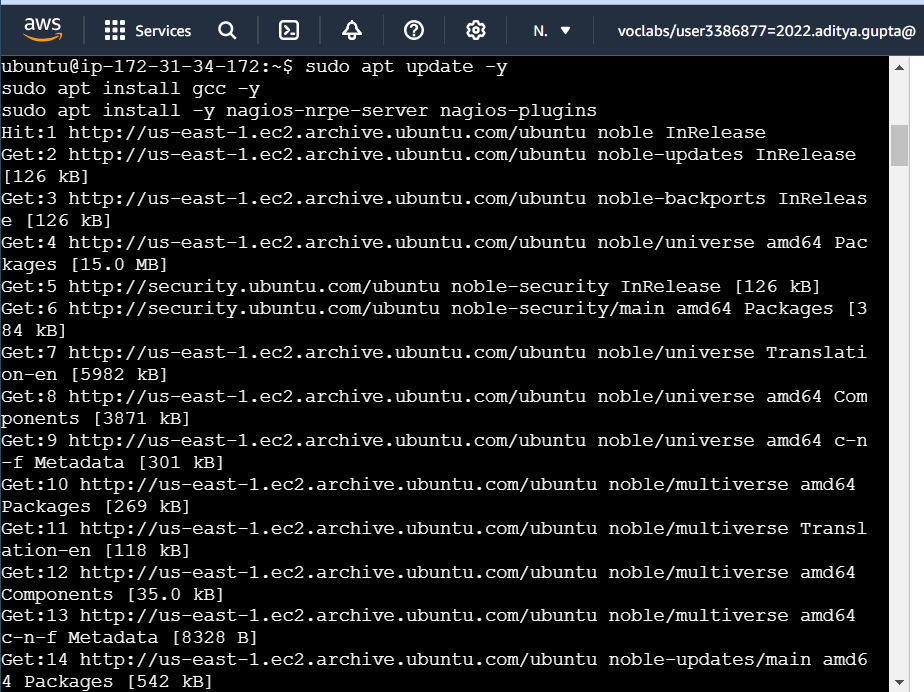
10. SSH into the machine or simply use the EC2 Instance Connect feature.

11. Make a package index update and install gcc, nagios-nrpe-server and the plugins.

sudo apt update -y

sudo apt install gcc -y

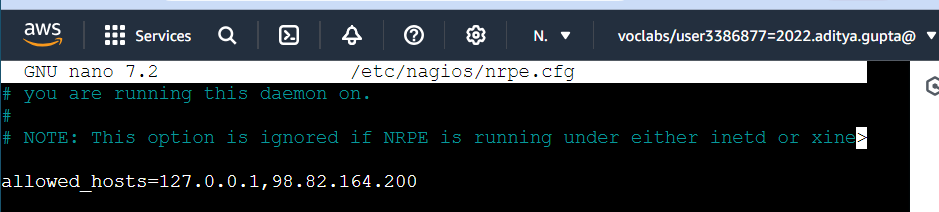
sudo apt install -y nagios-nrpe-server nagios-plugins



12. Open nrpe.cfg file to make changes.

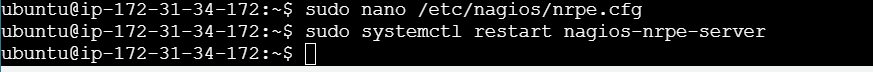
sudo nano /etc/nagios/nrpe.cfg

Under allowed\_hosts, add your nagios host IP address like so



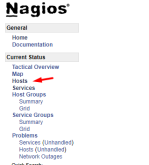
13. Restart the NRPE server

sudo systemctl restart nagios-nrpe-server



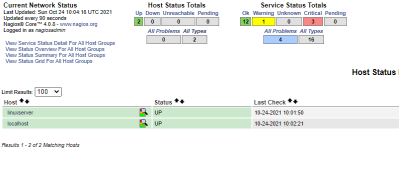
14. Now, check your nagios dashboard and you’ll see a new host being added.

Click on Hosts.

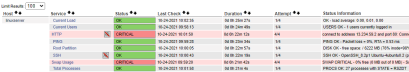




Click on linuxserver to see the host details



You can click Services to see all services and ports being monitored.



As you can see, we have our linuxserver up and running. It is showing critical status on HTTP due to permission errors and swap because there is no partition created.

**In this case, we have monitored -**

**Servers: 1 linux server**

**Services: swap**

**Ports: 22, 80 (ssh, http)**

**Processes: User status, Current load, total processes, root partition, etc.**

**Recommended Cleanup**

● Terminate both of your EC-2 instances to avoid charges.

● Delete the security group if you created a new one (it won’t affect your bill, you may avoid it)

**Conclusion:**

Thus, we learned about service monitoring using Nagios and successfully monitored a Linux Server and monitored its different ports and services using Nagios and NRPE.