

# Adv DevOps Practical 10

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

## Theory:

### Port and Service Monitoring

Port and service monitoring in Nagios involves checking the availability and responsiveness of network services running on specific ports. This ensures that critical services (like HTTP, FTP, or SSH) are operational. Nagios uses plugins to ping the ports and verify whether services are up and responding as expected, allowing administrators to be alerted in case of outages.

### Windows/Linux Server Monitoring

Windows/Linux server monitoring with Nagios entails tracking the performance and health of servers running these operating systems. It includes monitoring metrics such as CPU usage, memory consumption, disk space, and system logs. Nagios employs various plugins to gather data, enabling administrators to ensure optimal performance, identify potential issues, and maintain uptime across their server infrastructure.

## Prerequisites:

### AWS Academy or Personal account.

Nagios Server running on Amazon Linux Machine. (Refer Experiment No 9)

## Monitoring Using Nagios:

**Step 1:** To Confirm Nagios is running on the server side Perform the following command on your Amazon Linux Machine (Nagios-host).

**sudo systemctl status nagios**

```
[ec2-user@ip-172-31-91-91 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Sun 2024-09-29 16:18:08 UTC; 21min ago
     Docs: https://www.nagios.org/documentation
   Process: 1942 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Process: 1944 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
  Main PID: 1946 (nagios)
    Tasks: 8 (limit: 1112)
   Memory: 7.7M
      CPU: 387ms
   CGroup: /system.slice/nagios.service
           └─1946 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             └─1947 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
               └─1948 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                 └─1949 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                   └─1950 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                     └─1956 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
                       └─3088 /usr/local/nagios/libexec/check_ping -H 127.0.0.1 -w 3000.0,80% -c 5000.0,100% -p 5
                         └─3089 /usr/bin/ping -n -U -w 30 -c 5 127.0.0.1

Sep 29 16:18:08 ip-172-31-91-91.ec2.internal systemd[1]: Starting nagios.service - Nagios Core 4.5.5...
Sep 29 16:18:08 ip-172-31-91-91.ec2.internal systemd[1]: Started nagios.service - Nagios Core 4.5.5.
Sep 29 16:20:00 ip-172-31-91-91.ec2.internal nagios[1946]: SERVICE FLAPPING ALERT: localhost;HTTP;STARTED; Service appears to have started flapping (20.0% >
Sep 29 16:20:00 ip-172-31-91-91.ec2.internal nagios[1946]: SERVICE ALERT: localhost;HTTP;CRITICAL;HARD;4;connect to address 127.0.0.1 and port 80: Connecti
Sep 29 16:22:30 ip-172-31-91-91.ec2.internal nagios[1946]: SERVICE NOTIFICATION: nagiosadmin;localhost;Swap Usage;CRITICAL;notify-service-by-email;SWAP CRIS
Sep 29 16:22:30 ip-172-31-91-91.ec2.internal nagios[1946]: wproc: NOTIFY job 2 from worker Core Worker 1948 is a non-check helper but exited with return cod
Sep 29 16:22:30 ip-172-31-91-91.ec2.internal nagios[1946]: wproc: host=localhost; service=Swap Usage; contact=nagiosadmin
Sep 29 16:22:30 ip-172-31-91-91.ec2.internal nagios[1946]: wproc: early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
Sep 29 16:22:30 ip-172-31-91-91.ec2.internal nagios[1946]: wproc: stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
Sep 29 16:22:30 ip-172-31-91-91.ec2.internal nagios[1946]: wproc: stderr line 02: /usr/bin/printff: write error: Broken pipe
lines 1=30/30 (END)
```

You can now proceed if you get the above message/output.

**Step 2:** Now Create a new EC2 instance. Name: Nagios-client, AMI: Ubuntu Instance Type: t2.micro.

**Launch an instance** Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags** Info

Name  
nagios-client [Add additional tags](#)

**Application and OS Images (Amazon Machine Image)** Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

Recents | **Quick Start**

Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, [Browse more AMIs](#)

**Summary**

Number of instances Info  
1

Software Image (AMI)  
Canonical, Ubuntu, 24.04, amd64...[read more](#)  
ami-0e86c20dae9224d08

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
New security group

Storage (volumes)  
1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel **Launch instance** [Review commands](#)

**For Key pair :** Click on create key and make key of type RSA with extension .pem . Key will be downloaded to your local machine.  
Now select that key in key pair if you already have key with type RSA and extension .pem no need to create new key but you must have that key downloaded.

**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  
exp\_09 [Create new key pair](#)

**Summary**

Firewall (security group)  
New security group

Storage (volumes)  
1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Select the Existing Security Group and select the Security Group that we have created in Experiment no 9 or the same one you have used for the Nagios server (Nagios-host).

**Network settings** Info [Edit](#)

Network Info  
vpc-0d4c0d8f48c2e4508

Subnet Info  
No preference (Default subnet in any availability zone)

Auto-assign public IP Info  
Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group ☒ Select existing security group

Common security groups Info  
Select security groups  
launch-wizard-3 sg-09d51590eb1851b46 X  
VPC: vpc-0d4c0d8f48c2e4508

[Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

**Summary**

Number of instances Info  
1

Software Image (AMI)  
Canonical, Ubuntu, 24.04, amd64...[read more](#)  
ami-0e86c20dae9224d08

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
launch-wizard-3

Storage (volumes)  
1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

**Step 3:** Now After creating the EC2 Instance click on connect and then copy the command which is given as example in the SSH Client section .  
Now open the terminal in the folder where your key(RSA key with .pem) is located. and paste that copied command.

```
PS C:\Users\Ayush Maurya> ssh -i "Downloads/exp_09.pem" ubuntu@ec2-44-206-245-149.compute-1.amazonaws.com
The authenticity of host 'ec2-44-206-245-149.compute-1.amazonaws.com (44.206.245.149)' can't be established.
ED25519 key fingerprint is SHA256:DT+AA+mkcydh3k0J2vEpm4ZsA6FL+LM4m1QSIImddAHg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-206-245-149.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
```

```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-92-146:~$ |
```

**Now perform all the commands on the Nagios-host till step 10**

**Step 4:** Now on the server Nagios-host run the following command.

**ps -ef | grep nagios**

```
[ec2-user@ip-172-31-91-91 ~]$ ps -ef | grep nagios
nagios 1946 1 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios 1947 1946 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1948 1946 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1949 1946 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1950 1946 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1956 1946 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
root 3090 3055 0 16:40 pts/0 00:00:00 sudo systemctl status nagios
root 3092 3090 0 16:40 pts/1 00:00:00 sudo systemctl status nagios
root 3093 3092 0 16:40 pts/1 00:00:00 systemctl status nagios
ec2-user 3914 3890 0 16:59 pts/2 00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-91-91 ~]$ |
```

**Step 5:** Now Become root user and create root directories.

**sudo su**

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

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

```
[ec2-user@ip-172-31-91-91 ~]$ sudo su
[root@ip-172-31-91-91 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-91-91 ec2-user]# |
```

**Step 6:** Copy the sample localhost.cfg to linuxhost.cfg by running the following command.(Below command should come in one line see screenshot below)

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

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

```
[root@ip-172-31-91-91 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-91-91 ec2-user]# |
```

**Step 7:** Open linuxserver.cfg using nano and make the following changes in all positions?everywhere in file.

**> nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg**

Change hostname to **linuxserver**.

Change address to the **public IP of your Linux client**.

Set hostgroup\_name to **linux-servers1**.

```
#####
#
# HOST DEFINITION
#
#####

# Define a host for the local machine
define host {
    use                linux-server          ; Name of host template to use
                                           ; This host definition will inherit all variables that are defined
                                           ; in (or inherited by) the linux-server host template definition.

    host_name          linuxserver
    alias              localhost
    address            172.31.92.146
}

#####
#
# HOST GROUP DEFINITION
#
#####

# Define an optional hostgroup for Linux machines
define hostgroup {
    hostgroup_name     linux-servers[]      ; The name of the hostgroup
    alias              Linux Servers        ; Long name of the group
    members            localhost            ; Comma separated list of hosts that belong to this group
}

```

**Step 8:** Now update the Nagios config file .Add the following line in the file. Line to add :

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

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

```
#
# NAGIOS.CFG - Sample Main Config File for Nagios 4.5.5
#
# Read the documentation for more information on this configuration
# file. I've provided some comments here, but things may not be so
# clear without further explanation.
#
#
#####

# LOG FILE
# This is the main log file where service and host events are logged
# for historical purposes. This should be the first option specified
# in the config file!!!
log_file=/usr/local/nagios/var/nagios.log

# OBJECT CONFIGURATION FILE(S)
# These are the object configuration files in which you define hosts,
# host groups, contacts, contact groups, services, etc.
# You can split your object definitions across several config files
# if you wish (as shown below), or keep them all in a single config file.

# You can specify individual object config files as shown below:
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/
# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg

```

**Step 9:** Now Verify the configuration files by running the following commands.

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

```
Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-91-91 ec2-user]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
License: GPL

```

```

Running pre-flight check on configuration data...

Checking objects...
  Checked 8 services.
  Checked 2 hosts.
  Checked 2 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.
Checking for circular paths...
  Checked 2 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check

```

**Step 10:** Now restart the services of nagios by running the following command.

**service nagios restart**

```

Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-91-91 ec2-user]# service nagios restart
Restarting nagios (via systemctl): [ OK ]
[root@ip-172-31-91-91 ec2-user]#

```

**Step 11:** Now Go to the Nagios-client ssh terminal and update and install the packages by running the following command.

**sudo apt update -y**

**sudo apt install gcc -y**

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

```

ubuntu@ip-172-31-92-146:~$ sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins

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://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [380 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [82.9 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4560 B]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [272 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [115 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.3 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [353 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [68.1 kB]
Get:17 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [428 B]
Get:18 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.9 kB]
Get:19 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
Get:20 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:21 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]

```

```

Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart getty@tty1.service
systemctl restart networkd-dispatcher.service
systemctl restart serial-getty@ttyS0.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service

```

No containers need to be restarted.

```

User sessions running outdated binaries:
ubuntu @ session #2: sshd[992,1102]
ubuntu @ session #7: sshd[1190,1248]

```

```

ubuntu@ip-172-31-92-146:~$

```

**Step 12:** Open nrpe.cfg file to make changes. Under allowed\_hosts, add your nagios host IP address.

**sudo nano /etc/nagios/nrpe.cfg**

```
# NRPE GROUP
# This determines the effective group that the NRPE daemon should run as.
# You can either supply a group name or a GID.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

nrpe_group=nagios

# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostnames
# that are allowed to talk to the NRPE daemon. Network addresses with a bit mask
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address. I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

allowed_hosts=127.0.0.1,:134.207.68.187

# COMMAND ARGUMENT PROCESSING
# This option determines whether or not the NRPE daemon will allow clients
# to specify arguments to commands that are executed. This option only works
# if the daemon was configured with the --enable-command-args configure script
# option.
#
# *** ENABLING THIS OPTION IS A SECURITY RISK! ***
```

**Step 13:** Now restart the NRPE server by this command.

**sudo systemctl restart nagios-nrpe-server**

```
monitoring-plugins is already the newest version (2.15.0-1ubuntu5).
0 upgraded, 0 newly installed, 0 to remove and 139 not upgraded.
ubuntu@ip-172-31-92-146:~$ sudo nano /etc/nagios/nrpe.cfg
ubuntu@ip-172-31-92-146:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-92-146:~$
```

**Step 14:** Now again check the status of Nagios by running this command on Nagios-host and also check httpd is active and run the command to active it.

**sudo systemctl status nagios**

```
lec2-user@ip-172-31-91-91:~$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Sun 2024-09-29 17:20:07 UTC; 12min ago
     Docs: https://www.nagios.org/documentation
   Process: 4761 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Process: 4762 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 4763 (nagios)
    Tasks: 6 (limit: 1112)
   Memory: 4.1M
      CPU: 234ms
   CGroup: /system.slice/nagios.service
           └─4763 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             └─4764 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
               └─4765 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                 └─4766 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                   └─4767 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                     └─4768 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 29 17:20:07 ip-172-31-91-91.ec2.internal nagios[4763]: Warning: Duplicate definition found for service 'Current Users' on host 'localhost' (config file
Sep 29 17:20:07 ip-172-31-91-91.ec2.internal nagios[4763]: Warning: Duplicate definition found for service 'Root Partition' on host 'localhost' (config fil
Sep 29 17:20:07 ip-172-31-91-91.ec2.internal nagios[4763]: Warning: Duplicate definition found for service 'PING' on host 'localhost' (config file '/usr/lo
Sep 29 17:20:07 ip-172-31-91-91.ec2.internal nagios[4763]: Successfully launched command file worker with pid 4768
Sep 29 17:22:30 ip-172-31-91-91.ec2.internal nagios[4763]: SERVICE NOTIFICATION: nagiosadmin;localhost;Swap Usage;CRITICAL;notify-service-by-email;SWAP CRI
Sep 29 17:22:30 ip-172-31-91-91.ec2.internal nagios[4763]: wproc: NOTIFY job 1 from worker Core Worker 4766 is a non-check helper but exited with return co
Sep 29 17:22:30 ip-172-31-91-91.ec2.internal nagios[4763]: wproc: host=localhost; service=Swap Usage; contact=nagiosadmin
Sep 29 17:22:30 ip-172-31-91-91.ec2.internal nagios[4763]: wproc: early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
Sep 29 17:22:30 ip-172-31-91-91.ec2.internal nagios[4763]: wproc: stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
Sep 29 17:22:30 ip-172-31-91-91.ec2.internal nagios[4763]: wproc: stderr line 02: /usr/bin/printf: write error: Broken pipe
[lines 1-28/28 (END)]
```

## sudo systemctl status httpd

```
[ec2-user@ip-172-31-91-91 ~]$ sudo systemctl status httpd
o httpd.service - The Apache HTTP Server
Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
Drop-In: /usr/lib/systemd/system/httpd.service.d
└─php-fpm.conf
Active: inactive (dead)
Docs: man:httpd.service(8)
[ec2-user@ip-172-31-91-91 ~]$
```

## sudo systemctl start httpd

## sudo systemctl enable httpd

```
[ec2-user@ip-172-31-91-91 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-91-91 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-91-91 ~]$
```

Step 15: Now to check Nagios dashboard go to `http://<nagios host ip>/nagios`

Eg. `http://34.207.68.187/nagios`

Enter username as `nagiosadmin` and password which you set in Exp 9.

**Nagios®**

General

- Home
- Documentation

**Current Status**

- Tactical Overview
- Map
- Hosts
- Services
- Host Groups
- Summary
- Grid
- Service Groups
- Summary
- Grid
- Problems
- Services (Unhandled)
- Hosts (Unhandled)
- Network Outages
- Quick Search:

**Reports**

- Availability
- Trends
- Alerts
- History
- Summary
- Histogram
- Notifications
- Event Log

**System**

- Comments
- Down time
- Process Info
- Performance Info
- Scheduling Queue

**Nagios® Core™**

✓ Daemon running with PID 4763

**Nagios® Core™**  
Version 4.5.5  
September 17, 2024  
[Check for updates](#)

**Get Started**

- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Extend Nagios with hundreds of addons
- Get support
- Get training
- Get certified

**Latest News**

**Don't Miss...**

**Quick Links**

- Nagios Library (tutorials and docs)
- Nagios Labs (development blog)
- Nagios Exchange (plugins and addons)
- Nagios Support (tech support)
- Nagios.com (company)
- Nagios.org (project)

Copyright © 2010-2024 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

Nagios Core is licensed under the GNU General Public License and is provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Nagios, Nagios Core and the Nagios logo are trademarks, servicemarks, registered trademarks or registered servicemarks owned by Nagios Enterprises, LLC. Use of the Nagios marks is governed by the trademark use restrictions.

Now Click on Hosts from left side panel

**Nagios®**

General

- Home
- Documentation

**Current Status**

- Tactical Overview
- Map
- Hosts
- Services
- Host Groups
- Summary
- Grid
- Service Groups
- Summary
- Grid
- Problems
- Services (Unhandled)
- Hosts (Unhandled)
- Network Outages
- Quick Search:

**Host Status Details For All Host Groups**

Limit Results: 100

| Host        | Status | Last Check          | Duration      | Status Information                        |
|-------------|--------|---------------------|---------------|---|
| linuxserver | UP     | 09-29-2024 17:40:07 | 0d 0h 22m 43s | PING OK - Packet loss = 0%, RTA = 0.56 ms |
| localhost   | UP     | 09-29-2024 17:40:00 | 0d 9h 37m 43s | PING OK - Packet loss = 0%, RTA = 0.04 ms |

Results 1 - 2 of 2 Matching Hosts

**Conclusion:**

In this practical, we set up a Nagios host and client to monitor services and server performance on both Linux and Windows servers. We configured Nagios on an Amazon Linux machine to monitor critical services like HTTP, SSH, and system resources, ensuring their availability and health. By creating and configuring a new EC2 instance as the Nagios client, we enabled seamless communication between the client and host for efficient service monitoring. This setup helps ensure uptime and quick detection of issues across the infrastructure.