

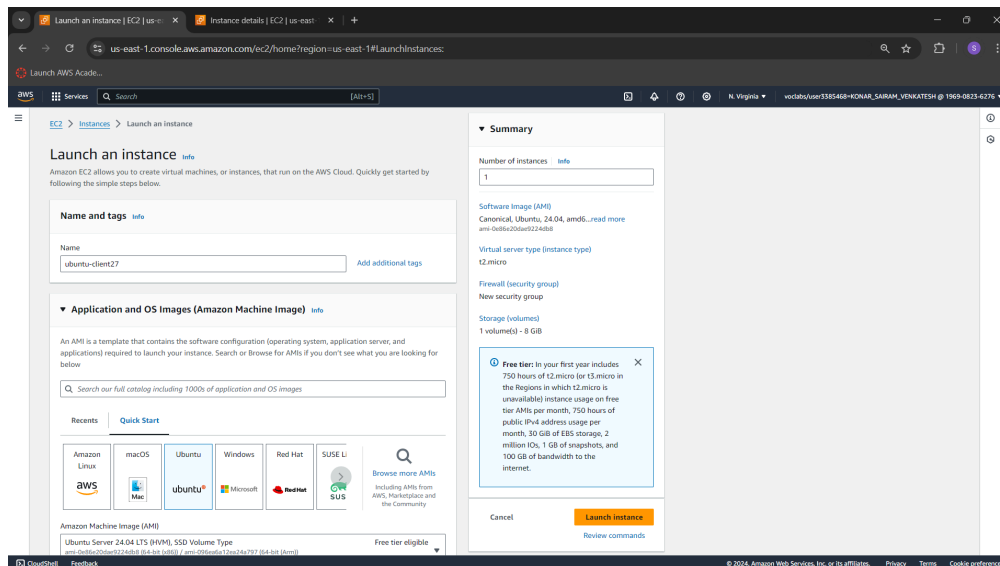
Aim: To perform Port, Service monitoring, Windows/Linux server monitoring using Nagios.

Prerequisites:

- 1) An Amazon Linux instance with nagios already set up.

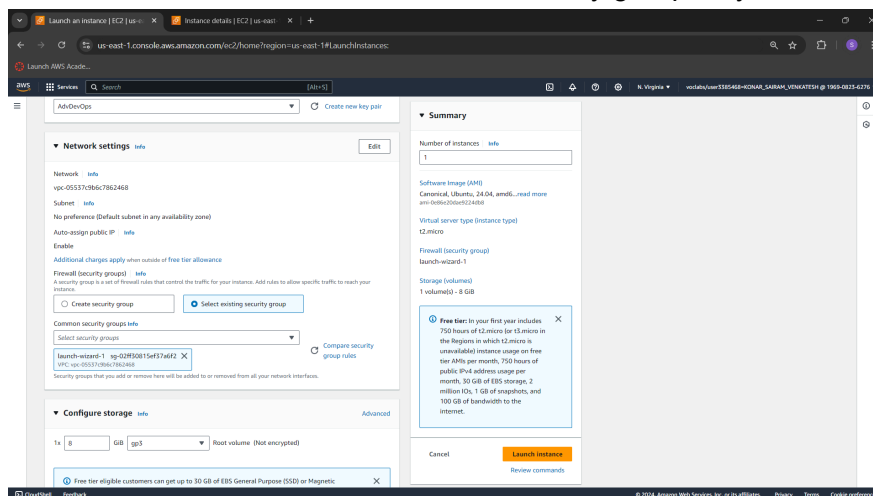
## Step 1: Set up ubuntu instance

- 1) Login to your AWS account. Search for EC2 on services. Open the interface and click on Create Instance.



Select The OS Image as Ubuntu.

- 2) Make sure to select the same private key that you created for the Amazon Linux instance. Also select the same security group as you created for the Linux instance.



- 3) Now come back to the instances screen. Click on the instance ID of your instance. Then click on Connect. Click on SSH client. Copy the example command. Now, we have to connect our local OS terminal to the instance using SSH. For this, open terminal where the private key file is located (.pem). Paste the copied SSH command and run it.

## Step 2: Execute the following on Nagios Host machine (Linux)

- 1) We need to verify whether the nagios service is running or not. For that, run this command.

**ps -ef | grep nagios**

```
[ec2-user@ip-172-31-83-157 ~]$ ps -ef | grep nagios
nagios 66054 1 0 04:18 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios 66055 66054 0 04:18 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 66056 66054 0 04:18 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 66057 66054 0 04:18 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 66058 66054 0 04:18 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 66059 66054 0 04:18 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user 66758 66657 0 04:29 pts/0 00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-83-157 ~]$
```

- 2) Now, make yourself as the root user, and create a folder with the path '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts'

**sudo su**

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

```
[ec2-user@ip-172-31-83-157 ~]$ sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-83-157 ec2-user]#
```

- 3) We need to create a config file in this folder. So, copy the contents of the existing localhost config to the new file 'linuxserver.cfg'.

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

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

- 4) We need to make some changes in this config file. Open it using nano editor.  
nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

Change **hostname** and **alias** to **linuxserver**

Change address to **public ip address of client instance** (Ubuntu instance)

```
# Define a host for the local machine

define host{
    use                linux-server          ; Name of host template
                                ; This host definition
                                ; in (or inherited)
    host_name          linuxserver
    alias               linuxserver
    address             32.226.136.73
}
```

Change `hostgroup_name` to **linux-servers1**

```
define hostgroup{
    hostgroup_name     linux-servers1 ; The name of the hostgroup
    alias               Linux Servers ; Long name of the group
    members             localhost      ; Comma separated list of hosts that
}
```

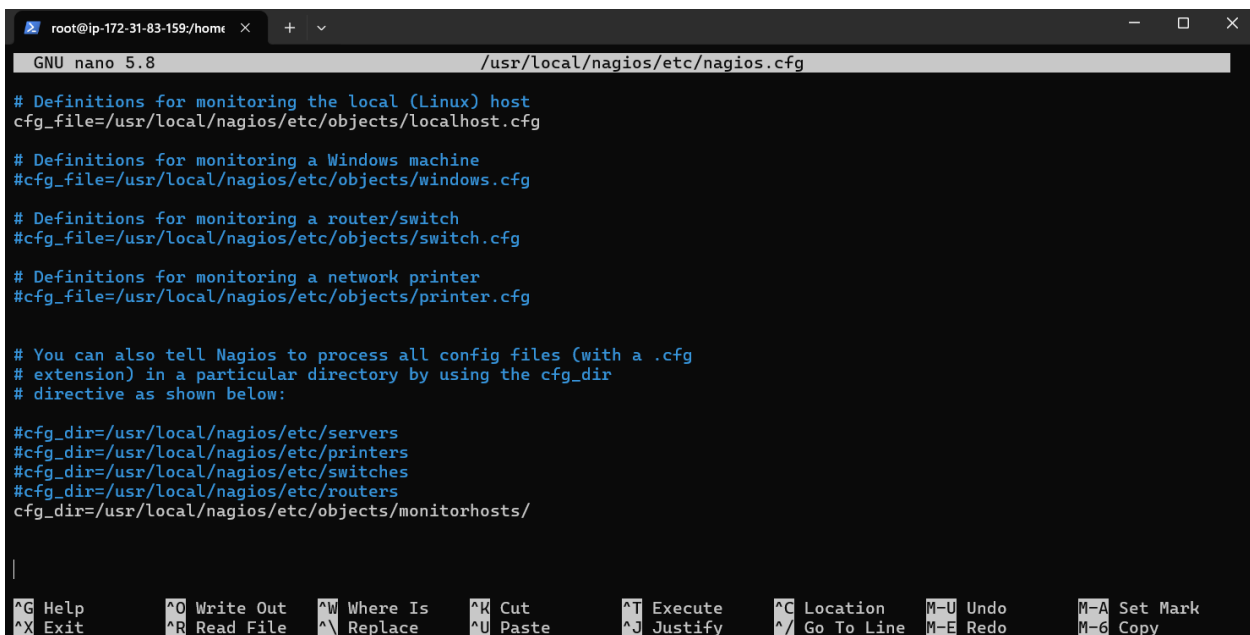
Change the **occurrences of hostname** further in the document from **localhost** to **linuxserver**

5) Now, we need to edit the nagios configuration file to add this directory.

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

Run this command and add the following line

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



```
root@ip-172-31-83-159:/home x + v
GNU nano 5.8 /usr/local/nagios/etc/nagios.cfg

# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg

# Definitions for monitoring a Windows machine
#cfg_file=/usr/local/nagios/etc/objects/windows.cfg

# Definitions for monitoring a router/switch
#cfg_file=/usr/local/nagios/etc/objects/switch.cfg

# Definitions for monitoring a network printer
#cfg_file=/usr/local/nagios/etc/objects/printer.cfg

# You can also tell Nagios to process all config files (with a .cfg
# extension) in a particular directory by using the cfg_dir
# directive as shown below:

#cfg_dir=/usr/local/nagios/etc/servers
#cfg_dir=/usr/local/nagios/etc/printers
#cfg_dir=/usr/local/nagios/etc/switches
#cfg_dir=/usr/local/nagios/etc/routers
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo     M-A Set Mark
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line M-E Redo     M-G Copy
```

6) Now we verify the configuration files.

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

```
[root@ip-172-31-83-157 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

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 16 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
[root@ip-172-31-83-157 ec2-user]#
```

7) Once the files are verified, we need to restart the server.

**service nagios restart**

```
[root@ip-172-31-83-159 nagios-plugins-2.0.3]# service nagios restart
Restarting nagios (via systemctl): [ OK ]
[root@ip-172-31-83-159 nagios-plugins-2.0.3]#
```

Step 3: Execute the following on Nagios Client machine (Ubuntu)

- 1) First, we check for any new updates, then we install gcc, nagios nrpe server and nagios plugins.

**sudo apt update -y**

**sudo apt install gcc -y**

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

```
ubuntu@ip-172-31-81-89:~$ 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://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
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://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]
```

Running kernel seems to be up-to-date.

Restarting services...

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 #4: sshd[1495,1569]
ubuntu @ user manager service: systemd[1500]
```

No VM guests are running outdated hypervisor (qemu) binaries on this host.

```
ubuntu@ip-172-31-81-89:~$ |
```

- 2) We need to add the public IP address of our host Nagios machine (Linux) to the nrpe configuration file.

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

Under `allowed_hosts`, add the nagios host ip address (public)

```
GNU nano 7.2 /etc/nagios/nrpe.cfg *
# You can either supply a username or a UID.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
nrpe_user=nagios

# 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=127.0.0.1,54.210.81.106

# 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.

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   ^U Undo       ^M Set Mark   ^_ To Bracket
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify    ^/_ Go To Line ^E Redo       ^G Copy       ^O Where Was
```

## Step 4: Check the Nagios Dashboard

- 1) Go to Nagios dashboard, click on hosts.

Here, we can see that the linuxserver is also added as a host.

The screenshot shows the Nagios web dashboard interface. The top navigation bar includes links for Home, Documentation, Current Status, Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, Problems, Reports, and System. The main content area displays the 'Host Status Details For All Host Groups' section, which lists the status of various hosts. The 'linuxserver' host is highlighted in green, indicating it is UP. The status bar shows 0 DOWN, 0 UNREACHABLE, and 0 PENDING hosts. The service status bar shows 1 OK, 0 WARNING, 0 CRITICAL, and 2 PENDING services. The host status details table shows the 'linuxserver' host is UP, last checked on 09-28-2024 at 04:42:16, with a duration of 0d 0h 2m 36s. The status information for the host is PING OK - Packet loss = 0%, RTA = 1.15 ms.

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	09-28-2024 04:42:16	0d 0h 2m 36s	PING OK - Packet loss = 0%, RTA = 1.15 ms

2) Click on linuxserver. Here, we can check all the information about linuxserver host.

The screenshot shows the Nagios web interface for host 'linuxserver (linuxserver)'. The interface includes a sidebar with navigation links like General, Home, Documentation, Current Status, Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, Problems, Reports, System, Availability, Trends, Alerts, History, Summary, Histogram, Notifications, Event Log, Comments, Downtime, Process Info, Performance Info, Scheduling Queue, and Configuration.

**Host Information:**  
Last Updated: Sat Sep 28 04:43:37 UTC 2024  
Updated every 90 seconds  
Nagios® Core™ 4.5.5 - www.nagios.org  
Logged in as nagiosadmin

**Host State Information:**  
Host Status: **UP** (for 0d 0h 3m 13s)  
Status Information: PING OK - Packet loss = 0%, RTA = 1.15 ms  
Performance Data: rta=1.151000ms,3000.000000,5000.000000,0.000000 p1=0%,80,100,0  
Current Attempt: 1/10 (HARD state)  
Last Check Time: 09-28-2024 04:42:16  
Check Type: ACTIVE  
Check Latency / Duration: 0.000 / 4.033 seconds  
Next Scheduled Active Check: 09-28-2024 04:47:16  
Last State Change: 09-28-2024 04:40:24  
Last Notification: N/A (notification 0)  
Is This Host Flapping? **NO** (0.00% state change)  
In Scheduled Downtime? **NO**  
Last Update: 09-28-2024 04:43:33 ( 0d 0h 0m 4s ago)

**Active Checks:** ENABLED  
**Passive Checks:** ENABLED  
**Obsessing:** ENABLED  
**Notifications:** ENABLED  
**Event Handler:** ENABLED  
**Flap Detection:** ENABLED

**Host Commands:**  
Locate host on map  
Disable active checks of this host  
Re-schedule the next check of this host  
Submit passive check result for this host  
Stop accepting passive checks for this host  
Stop obsessing over this host  
Disable notifications for this host  
Send custom host notification  
Schedule downtime for this host  
Schedule downtime for all services on this host  
Disable notifications for all services on this host  
Enable notifications for all services on this host  
Schedule a check of all services on this host  
Disable checks of all services on this host  
Enable checks of all services on this host  
Disable event handler for this host  
Disable flap detection for this host  
Clear flapping state for this host

**Host Comments:**  
Add a new comment Delete all comments  
Entry Time Author Comment Comment ID Persistent Type Expires Actions  
This host has no comments associated with it

3) Click on services. Here we can see all the services that are being monitored by linuxserver.

The screenshot shows the Nagios web interface for service status details. The interface includes a sidebar with navigation links like General, Home, Documentation, Current Status, Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, Problems, Reports, System, Availability, Trends, Alerts, History, Summary, Histogram, Notifications, Event Log, Comments, Downtime, Process Info, Performance Info, Scheduling Queue, and Configuration.

**Current Network Status:**  
Last Updated: Sat Sep 28 04:44:10 UTC 2024  
Updated every 90 seconds  
Nagios® Core™ 4.5.5 - www.nagios.org  
Logged in as nagiosadmin

**Host Status Totals:**  
Up Down Unreachable Pending  
2 0 0 0  
All Problems All Types  
0 2

**Service Status Totals:**  
Ok Warning Unknown Critical Pending  
10 1 0 2 3  
All Problems All Types  
3 16

**Service Status Details For All Hosts:**

Host	Service	Status	Last Check	Duration	Attempt	Status Information	
linuxserver	Current Load	OK	09-28-2024 04:41:31	0d 0h 3m 46s+	1/4	OK - load average: 0.00, 0.01, 0.00	
	Current Users	OK	09-28-2024 04:41:39	0d 0h 3m 46s+	1/4	USERS OK - 2 users currently logged in	
	HTTP	CRITICAL	09-28-2024 04:43:16	0d 0h 1m 54s	2/4	connect to address 107.22.153.120 and port 80: Connection refused	
	PING	OK	09-28-2024 04:42:54	0d 0h 3m 46s+	1/4	PING OK - Packet loss = 0%, RTA = 1.11 ms	
	Root Partition	OK	09-28-2024 04:43:31	0d 0h 3m 46s+	1/4	DISK OK - free space / 6116 MB (75.36% inode=98%)	
	SSH	PENDING	N/A	0d 0h 3m 46s+	1/4	Service check scheduled for Sat Sep 28 04:44:09 UTC 2024	
	Swap Usage	PENDING	N/A	0d 0h 3m 46s+	1/4	Service check scheduled for Sat Sep 28 04:44:46 UTC 2024	
	Total Processes	PENDING	N/A	0d 0h 3m 46s+	1/4	Service check scheduled for Sat Sep 28 04:45:24 UTC 2024	
	localhost	Current Load	OK	09-28-2024 04:39:36	0d 0h 24m 34s	1/4	OK - load average: 0.00, 0.02, 0.00
	Current Users	OK	09-28-2024 04:40:14	0d 0h 23m 56s	1/4	USERS OK - 2 users currently logged in	
HTTP	WARNING	09-28-2024 04:43:51	0d 0h 20m 19s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.000 second response time		
PING	OK	09-28-2024 04:41:29	0d 0h 22m 41s	1/4	PING OK - Packet loss = 0%, RTA = 0.03 ms		
Root Partition	OK	09-28-2024 04:42:06	0d 0h 22m 4s	1/4	DISK OK - free space / 6116 MB (75.36% inode=98%)		
SSH	OK	09-28-2024 04:42:44	0d 0h 21m 26s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)		
Swap Usage	CRITICAL	09-28-2024 04:41:21	0d 0h 17m 49s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.		
Total Processes	OK	09-28-2024 04:43:59	0d 0h 20m 11s	1/4	PROCS OK: 37 processes with STATE = RSZDT		

Results 1 - 16 of 16 Matching Services

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.

### **Conclusion:**

In this experiment, we learned to perform port service monitoring and server monitoring using Nagios. For this, we need the Linux instance used to host the Nagios dashboard and server.

Also, we would need an Ubuntu instance which would be linked to a second host. We need to set up some configurations on the Linux instance and add the IP address of the Ubuntu instance. After that, we need to make the same initial setup on the ubuntu instance as the linux instance. Add the IP address of linux instance in allowed hosts. After restarting the NRPE server, we can see the 'linuxserver' host added.