

ADVANCE DEVOPS EXPERIMENT 10

Name:Aryan Anil Patankar

Class;D15A

Roll No:34

1) Launch an instance

Launch an ec2 instance.

Select Ubuntu as the os give a meaningful name of the instance.

EC2 > Instances > Launch an instance

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

 [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

Recents

Quick Start

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

SUSE Linux

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

▼ Summary

Number of instances [Info](#)

Software Image (AMI)

Canonical, Ubuntu, 24.04, a
ami-0e86e20dae9224db8

Virtual server type (instance type)

t2.micro

Firewall (security group)

launch-wizard-5

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first 750 hours of t2.micro instances in the Regions in which you have a free tier (not available in all Regions), you can run up to 7 t2.micro instances per month, with a public IPv4 address per instance, 30 GiB of EBS standard storage per instance, 1 million I/Os, 1 GB of S3 transfer per instance, and 100 GB of bandwidth for internet.

Cancel

Select the same security group as given in exp9.

▼ 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

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

Recents Quick Start

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

SUSE Li

Browse more AMIs

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-0e86e20dae9224db8 (64-bit (x86)) / ami-096ea6a12ea24a797 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Architecture

AMI ID

Username

Verified provider

▼ Summary

Number of inst

1

Software Imag

Canonical, Ubu

ami-0e86e20dae

Virtual server t

t2.micro

Firewall (secu

launch-wizard-

Storage (volum

1 volume(s) - 8

Free tie

750 hou

the Reg

unavaila

tier AMI

public I

month,

million

100 GB

internet

Cancel

Make sure to select the same key-pair login used in the exp9 machine.

The screenshot shows the 'Launch instance' wizard in the AWS Management Console. The 'Key pair (login)' step is active, showing a dropdown menu with 'nagios_exp_9' selected and a 'Create new key pair' button. The 'Network settings' step is also visible, showing 'vpc-07b6966cbfba88ee3' selected for the network, 'No preference' for the subnet, and 'Enable' for auto-assigning a public IP. The firewall section shows 'Select existing security group' as the chosen option. On the right, a sidebar shows the 'Free tier' allowance for t2.micro instances, indicating that the current configuration is within the free tier limits.

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*

nagios_exp_9

Create new key pair

Network settings Info

Network Info

vpc-07b6966cbfba88ee3

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

Free tier allowance

Free tier allowance for t2.micro instances. The free tier allowance is 750 hours per month. The free tier allowance is not available for public IP addresses. The free tier allowance is not available for more than one instance per Availability Zone. The free tier allowance is not available for more than one instance per Availability Zone. The free tier allowance is not available for more than one instance per Availability Zone.

Cancel

click on launch instance.

Now connect with this client machine using the ssh through your terminal(open a new terminal in your local machine and we will need both of the terminals open)

The screenshot shows the 'Instances' page in the AWS Management Console. It displays a list of five EC2 instances. The 'exp10client' instance is highlighted, showing its state as 'Initializing'. The other instances are 'Running'. The table includes columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Master	i-0ab175e9c60cc3a23	Running	t2.micro	2/2 checks passed	View alarms	us-east-1b	ec2-3-82-156-160.com...
node-1	i-08ad30b7114767ca2	Running	t2.micro	2/2 checks passed	View alarms	us-east-1b	ec2-3-85-110-80.comp...
node-2	i-03c70d364fb762af5	Running	t2.micro	2/2 checks passed	View alarms	us-east-1b	ec2-54-226-209-38.co...
nagios_host_e...	i-0820376be204a7fcb	Running	t2.micro	2/2 checks passed	View alarms	us-east-1b	ec2-54-224-175-95.co...
exp10client	i-0994ca5a178801a54	Initializing	t2.micro	Initializing	View alarms	us-east-1b	ec2-54-173-58-143.co...

EC2 > Instances > i-0994ca5a178801a54 > Connect to instance

Connect to instance Info

Connect to your instance i-0994ca5a178801a54 (exp10client) using any of these options

EC2 Instance Connect

Session Manager

SSH client

EC2 serial console

Instance ID
i-0994ca5a178801a54 (exp10client)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is nagios_exp_9.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 "nagios_exp_9.pem"
4. Connect to your instance using its Public DNS:
ec2-54-173-58-143.compute-1.amazonaws.com

✓ Command copied

ssh -i "nagios_exp_9.pem" ubuntu@ec2-54-173-58-143.compute-1.amazonaws.com

Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel

Note to change the path of the .pem file.

```
Host Client
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Lenovo> ssh -i "C:\Users\Lenovo\Downloads\nagios_exp_9.pem" ubuntu@ec2-54-173-58-143.compute-1.amazonaws.com

The authenticity of host 'ec2-54-173-58-143.compute-1.amazonaws.com (54.173.58.143)' can't be established.
ED25519 key fingerprint is SHA256:IA3XH7f011spK084wDcZFmqRgNn0iJZ7itI2pBMmHP4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-173-58-143.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
 * Support:       https://ubuntu.com/pro

System information as of Sat Sep 28 10:43:28 UTC 2024

System load: 0.01          Processes:            107
Usage of /:  22.8% of 6.71GB Users logged in:          0
Memory usage: 19%         IPv4 address for enx0: 172.31.82.77
```

2) Go to nagios host machine (Host machine)

Perform the following commands

`ps -ef | grep nagios`

```
Host Client
[ec2-user@ip-172-31-80-137 ~]$ ps -ef | grep nagios
nagios 3152 1 0 08:36 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios 3153 3152 0 08:36 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 3154 3152 0 08:36 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 3155 3152 0 08:36 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 3156 3152 0 08:36 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 3160 3152 0 08:36 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user 11528 2972 0 10:44 pts/0 00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-80-137 ~]$
```

`sudo su`

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

```
[root@ip-172-31-80-137 ec2-user]# mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-80-137 ec2-user]# ls
```

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

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

```
[root@ip-172-31-80-137 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
```

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

```
[root@ip-172-31-80-137 ec2-user]# 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) you can get the ip address by clicking on the instance id on the instances section there you will get the public ipv4 address

Instance summary for i-0994ca5a178801a54 (exp10client)

Updated less than a minute ago

Instance ID: i-0994ca5a178801a54 (exp10client)

IPv6 address: -

Hostname type: IP name: ip-172-31-82-77.ec2.internal

Answer private resource DNS name: IPv4 (A)

Auto-assigned IP address

Public IPv4 address copied

54.173.58.143 | open address

Instance state: Running

Private IP DNS name (IPv4 only): ip-172-31-82-77.ec2.internal

Instance type: t2.micro

VPC ID

Private IPv4 addresses: 172.31.82.77

Public IPv4 DNS: ec2-54-173-58-143.compute-1.amazonaws.com | open address

Elastic IP addresses: -

AWS Compute Optimizer finding

```

# 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
                                           ; its values from the template
    host_name          linuxserver             ; Name of the host to monitor
    alias               linuxserver
    address             54.173.58.143
}

```

Change hostgroup_name to linux-servers1

```

# Define an optional hostgroup for Linux machines

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
}

```

Change the occurrences of hostname further in the document from localhost to linuxserver
example like:

```

host_name          localhost
service_description PING

```

changed to

```

define service {

    use                local-service           ; Name of service template
    host_name          linuxserver
    service_description PING
    check_command       check_ping!100.0,20%!500.0,60%
}

```

This is the last one

```

define service {
    use                local-service                ; Name of service template to
    host_name          linuxserver
    service_description HTTP
    check_command       check_http
    notifications_enabled 0

```

now ctrl+O and enter to save and then ctrl+X for exiting.

Open nagios configuration file and add the line shown below

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

```

[root@ip-172-31-80-137 ec2-user]# nano /usr/local/nagios/etc/nagios.cfg

```

##Add this line below the opened nano interface where similar lines are commented.

cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

```

GNU nano 5.8 /usr/local/nagios/etc/nagios.cfg
# 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

# 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/

# OBJECT CACHE FILE
# This option determines where object definitions are cached when
# Nagios starts/restarts. The CCTs read object definitions from

```

ctrl+o and enter for saving and ctrl+x to exit nano editor.

Verify configuration files

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

```
[root@ip-172-31-80-137 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 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-172-31-80-137 ec2-user]# |
```

Restart nagios service.

`service nagios restart`

```
Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-80-137 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-80-137 ec2-user]# |
```

3) Go to client machine (ubuntu machine)

Perform the following commands

`sudo apt update -y`

`sudo apt install gcc -y`

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

```
Host Client
ubuntu@ip-172-31-82-77:~$ 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]

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 #1: sshd[990,1101]
ubuntu @ user manager service: systemd[996]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-82-77:~$ |
```

Open the nrpe.cfg file in nano editor

sudo nano /etc/nagios/nrpe.cfg

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

```

# 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

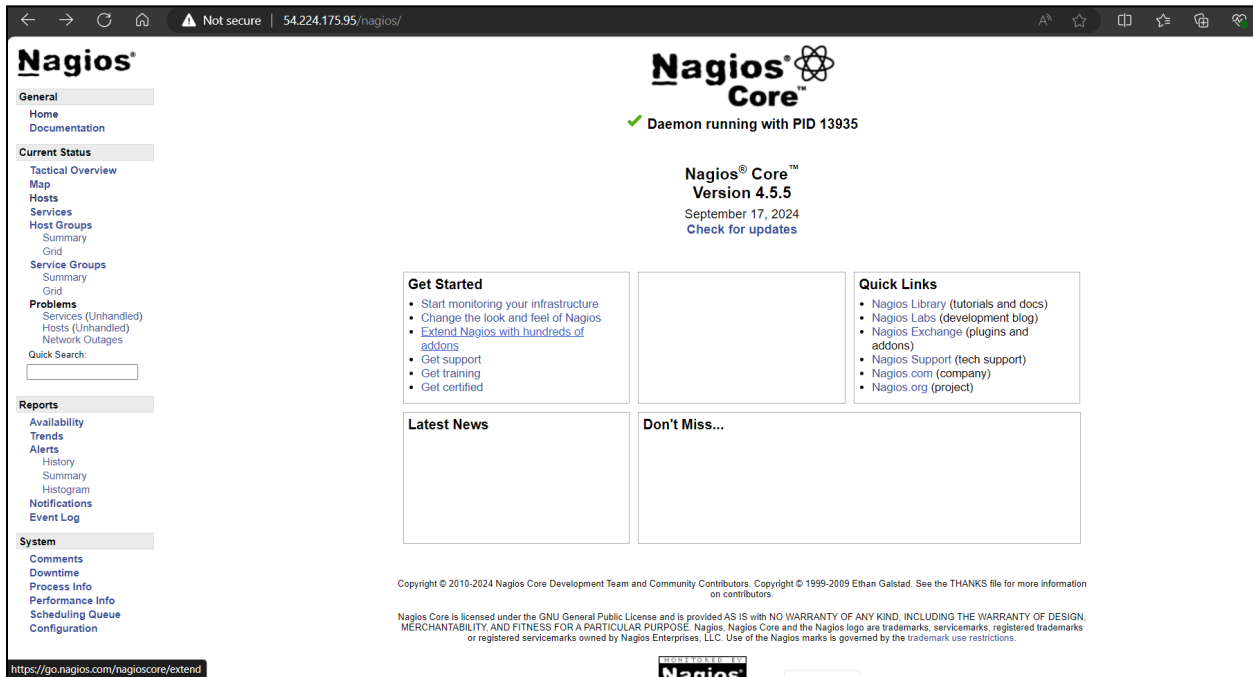
# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostnames
# that are allowed to talk to the NRPE daemon. Network addresses with a bit
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary 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=127.0.0.1,54.224.175.95

# COMMAND ARGUMENT PROCESSING
# This option determines whether or not the NRPE daemon will allow clients

```

again save and exit the nano editor.

4) Go to nagios dashboard and click on hosts



The screenshot shows the Nagios Core dashboard in a web browser. The browser's address bar displays "54.224.175.95/nagios/". The dashboard header includes the Nagios logo and the text "Nagios Core Version 4.5.5" with a status message "Daemon running with PID 13935". A left sidebar contains navigation links under categories: 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), Reports (Availability, Trends, Alerts, History, Summary, Histogram, Notifications, Event Log), and System (Comments, Downtime, Process Info, Performance Info, Scheduling Queue, Configuration). The main content area features a "Get Started" section with links to start monitoring, change the look, extend Nagios with add-ons, get support, get training, and get certified. It also includes "Quick Links" to Nagios Library, Nagios Labs, Nagios Exchange, Nagios Support, Nagios.com, and Nagios.org. Below these are sections for "Latest News" and "Don't Miss...". The footer contains copyright information and a license statement.

Nagios

Nagios Core
✓ Daemon running with PID 13935

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

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)

Latest News

Don't Miss...

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.

<https://go.nagios.com/nagioscore/extend>

Click on hosts



The screenshot shows a close-up of the "Current Status" section of the Nagios Core dashboard. The section is titled "Current Status" and contains a list of links: "Tactical Overview", "Map", "Hosts", "Services", and "Host Groups". The "Hosts" link is highlighted in blue.

Current Status

[Tactical Overview](#)


[Map](#)

[Hosts](#)

[Services](#)

[Host Groups](#)

5) Click on linux server



Current Network Status

Last Updated: Sat Sep 28 11:33:24 UTC 2024
Updated every 30 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as: nagiosadmin

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

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
12	1	0	3	0

All Problems: All Types

4	16
---	----

View Service Status Detail For All Host Groups

View Status Overview For All Host Groups

View Status Summary For All Host Groups

View Status Grid For All Host Groups

Host Status Details For All Host Groups

Limit Results:

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	09-28-2024 11:29:10	0d 0h 6m 36s	PING OK - Packet loss = 0%, RTA = 1.10 ms
localhost	UP	09-28-2024 11:32:16	0d 3h 53m 7s	PING OK - Packet loss = 0%, RTA = 0.83 ms

Results 1 - 2 of 2 Matching Hosts

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
- Downtime
- Flow Info
- Scheduling Queue
- Configuration

Host Information

Last Updated: Sat Sep 28 11:33:39 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as nagiosadmin

View Status Detail For This Host
View Alert History For This Host
View Trends For This Host
View Alert Histogram For This Host
View Availability Report For This Host
View Notifications For This Host

Host
linuxserver
(linuxserver)

Member of
No hostgroups

54.173.58.143

Host State Information

Host Status:	UP! (for 0d 0h 8m 51s)
Status Information:	PING OK - Packet loss = 0%, RTA = 1.18 ms
Performance Data:	rta=1.184000ms;3000.000000;5000.000000;0.000000 p1=0%;80;100;0
Current Attempt:	1/10 (HARD state)
Last Check Time:	09-28-2024 11:29:10
Check Type:	ACTIVE
Check Latency / Duration:	0.000 / 4.066 seconds
Next Scheduled Active Check:	09-28-2024 11:34:10
Last State Change:	09-28-2024 11:24:48
Last Notification:	N/A (notification 0)
Is This Host Flapping?	NO (0.00% state change)
In Scheduled Downtime?	NO
Last Update:	09-28-2024 11:33:37 (0d 0h 0m 2s ago)
Active Checks:	ENABLED
Passive Checks:	ENABLED
Obsessing:	ENABLED
Notifications:	ENABLED
Event Handler:	ENABLED
Flap Detection:	ENABLED

Host Commands

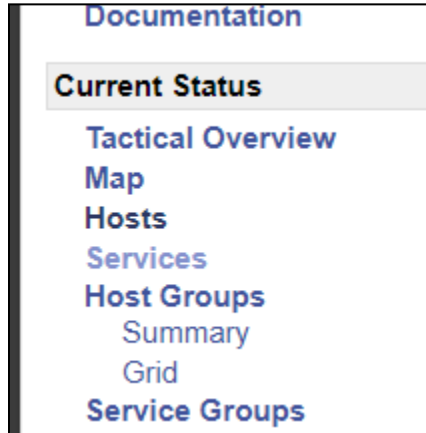
- Locate host on map
- Disable active checks of this host
- Re-schedule the next check of this host
- Submit passive check result for this host
- Stop accepting passive checks for this host
- Stop obsessing over this host
- Disable notifications for this host
- Send custom host notification
- Schedule 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.							

6) Click on nagios services



Nagios® Current Network Status
Last Updated: Sat Sep 28 11:33:58 UTC 2024
Updated every 50 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as nagiosadmin

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

Service Status Totals
Ok: 12, Warning: 1, Unknown: 0, Critical: 3, Pending: 0
All Problems: 4, All Types: 16

Service Status Details For All Hosts

Host	Service	Status	Last Check	Duration	Attempt	Status Information
linuxserver	Current Load	OK	09-28-2024 11:30:25	0d 0h 8m 33s	1/4	OK - load average: 0.01, 0.00, 0.00
	Current Users	OK	09-28-2024 11:31:03	0d 0h 7m 55s	1/4	USERS OK - 2 users currently logged in
	HTTP	CRITICAL	09-28-2024 11:29:40	0d 0h 4m 18s	4/4	connect to address 54.173.58.143 and port 80: Connection refused
	PING	OK	09-28-2024 11:32:18	0d 0h 6m 40s	1/4	PING OK - Packet loss = 0%, RTA = 1.03 ms
	Root Partition	OK	09-28-2024 11:32:55	0d 0h 6m 3s	1/4	DISK OK - free space: / 6105 MiB (75.23% inode=96%)
	SSH	OK	09-28-2024 11:33:33	0d 0h 5m 25s	1/4	SSH OK - OpenSSH_9.6p1 Ubuntu-3ubuntu13.4 (protocol 2.0)
	Swap Usage	CRITICAL	09-28-2024 11:32:10	0d 0h 1m 48s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
localhost	Total Processes	OK	09-28-2024 11:29:48	0d 0h 9m 10s+	1/4	PROCS OK: 37 processes with STATE = RSZDT
	Current Load	OK	09-28-2024 11:29:39	0d 3h 53m 5s	1/4	OK - load average: 0.02, 0.01, 0.00
	Current Users	OK	09-28-2024 11:30:17	0d 3h 52m 27s	1/4	USERS OK - 2 users currently logged in
	HTTP	WARNING	09-28-2024 11:29:46	0d 2h 49m 12s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.001 second response time
	PING	OK	09-28-2024 11:31:32	0d 3h 51m 12s	1/4	PING OK - Packet loss = 0%, RTA = 0.93 ms
	Root Partition	OK	09-28-2024 11:32:09	0d 3h 50m 35s	1/4	DISK OK - free space: / 6105 MiB (75.23% inode=96%)
	SSH	OK	09-28-2024 11:32:47	0d 3h 49m 57s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)
	Swap Usage	CRITICAL	09-28-2024 11:31:24	0d 3h 12m 34s	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 11:29:02	0d 3h 14m 56s	1/4	PROCS OK: 37 processes with STATE = RSZDT

Results 1 - 16 of 16 Matching Services

Conclusion:

In this lab, we successfully configured a monitoring setup between a Nagios host machine (referred to as "exp9 machine") and a client machine (created specifically for this experiment). The goal was to set up Nagios to monitor a remote Linux server, which involved configuring both the Nagios host and client machine (Ubuntu instance) in an EC2 environment.