

Name : Nidhi Rijhwani

Subject : Adv Devops Exp No 10

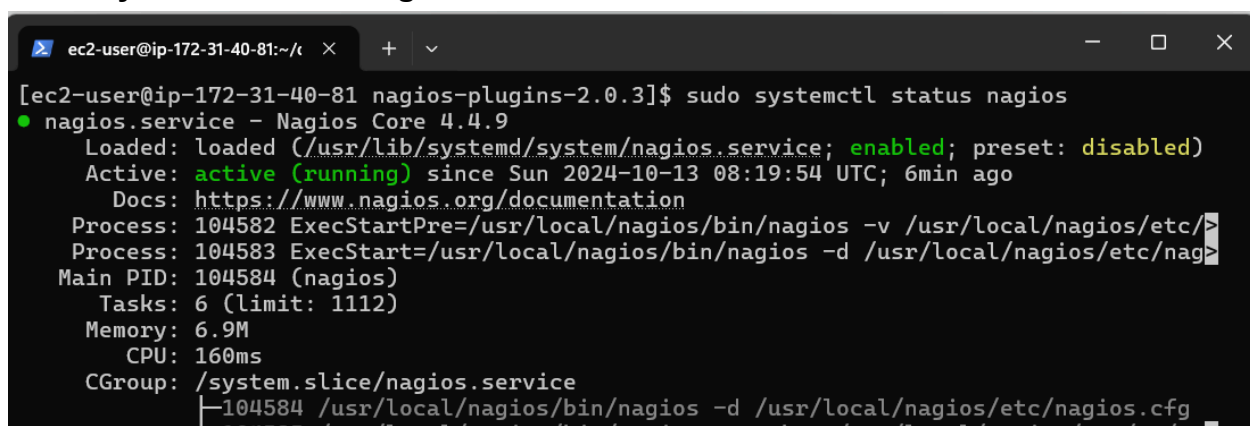
Aim: To perform Port, Service monitoring, Windows/Linux server monitoring using Nagios.

Theory :

1. Port Monitoring
 - Purpose: Ensures that specific ports are open and services are responding.
 - Implementation: Using the `check_tcp` plugin, Nagios can check if a TCP port is accepting connections. This is crucial for monitoring services like HTTP, FTP, or custom applications.
2. Service Monitoring
 - Purpose: Monitors the health and performance of services running on servers.
 - Implementation: Service checks can include HTTP responses, database connection checks, and application health checks. Plugins like `check_http` and `check_mysql` are commonly used.
3. Server Monitoring
 - Windows Monitoring: Uses plugins such as `check_wmi` to gather performance metrics from Windows servers, including CPU usage, memory, disk space, and services status.
 - Linux Monitoring: Utilizes native Linux commands (like `check_load`, `check_disk`, etc.) to monitor system metrics, ensuring the server is operating efficiently.
4. Configuration
 - Monitoring configurations are defined in `.cfg` files, specifying hosts, services, and checks. The Nagios server periodically polls these configurations to perform checks.

Steps :

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

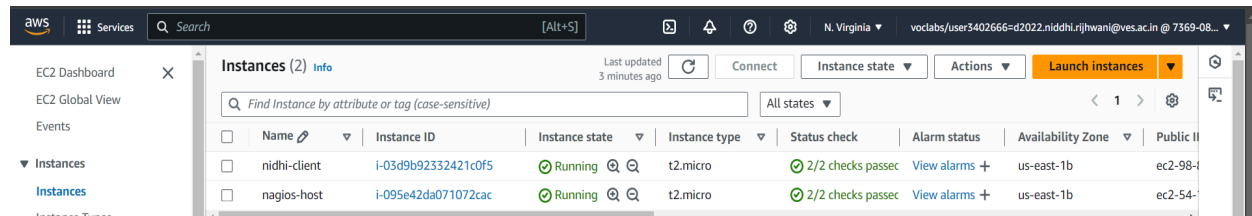


```
ec2-user@ip-172-31-40-81:~/t x + v
[ec2-user@ip-172-31-40-81 nagios-plugins-2.0.3]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.4.9
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Sun 2024-10-13 08:19:54 UTC; 6min ago
     Docs: https://www.nagios.org/documentation
   Process: 104582 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/>
   Process: 104583 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nag>
   Main PID: 104584 (nagios)
    Tasks: 6 (limit: 1112)
   Memory: 6.9M
      CPU: 160ms
   CGroup: /system.slice/nagios.service
           └─104584 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             104585 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
```

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.



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`

```
[ec2-user@ip-172-31-40-81 nagios-plugins-2.0.3]$ ps -ef | grep nagios
nagios 104584 1 0 08:19 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios 104585 104584 0 08:19 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 104586 104584 0 08:19 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 104587 104584 0 08:19 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 104588 104584 0 08:19 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 104589 104584 0 08:19 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user 105254 2401 0 08:30 pts/0 00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-40-81 nagios-plugins-2.0.3]$
```

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`

```
[ec2-user@ip-172-31-40-81 nagios-plugins-2.0.3]$ 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 linux host folder

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

```
[root@ip-172-31-40-81 nagios-plugins-2.0.3]# mkdir /usr/local/nagios/etc/objects/monitorhosts
[root@ip-172-31-40-81 nagios-plugins-2.0.3]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-40-81 nagios-plugins-2.0.3]# 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

- Change the hostname to linuxserver (EVERYWHERE ON THE FILE)
- Change address to the public IP address of your LINUX CLIENT.

```
[root@ip-172-31-40-81 nagios-plugins-2.0.3]# vim /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
```

```
# 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-ser
ver host template definition.
    host_name linuxserver
    alias linuxserver
    address 98.83.233.82
}
```

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

# 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            linuxserver|           ; Comma separated list of hosts th
at belong to this group
}

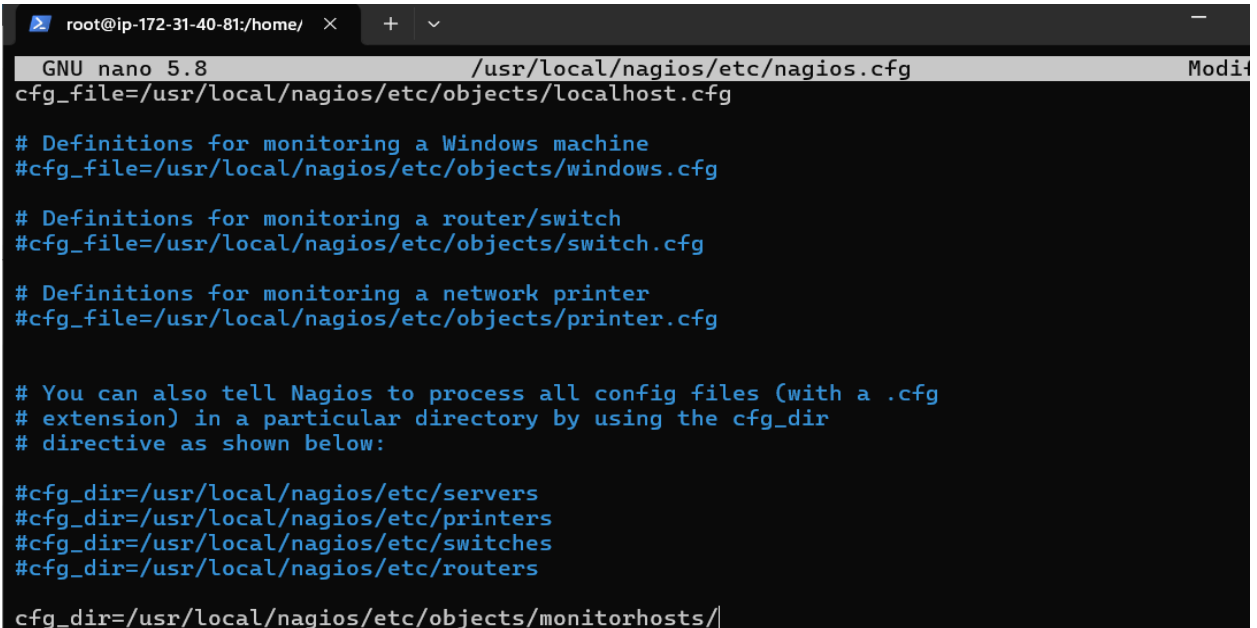
```

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/

```
[root@ip-172-31-40-81 nagios-plugins-2.0.3]# nano /usr/local/nagios/etc/nagios.cfg
[root@ip-172-31-40-81 nagios-plugins-2.0.3]# |

```



```
root@ip-172-31-40-81:/home/  ×  +  ~
GNU nano 5.8 /usr/local/nagios/etc/nagios.cfg Modif
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/|

```

8. Verify the configuration files

```

    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-172-31-40-81 nagios-plugins-2.0.3]# |

```

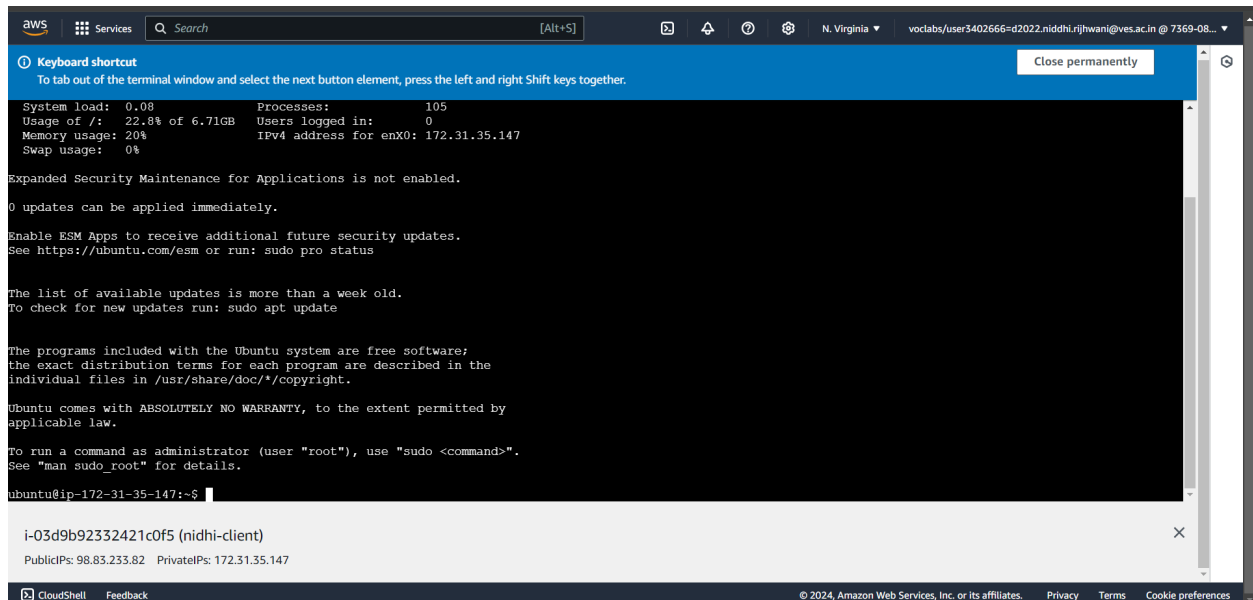
You are good to go if there are no errors.

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.



The screenshot shows the AWS CloudShell interface. At the top, there's a search bar and navigation icons. Below that, a blue banner displays keyboard shortcuts. The terminal window shows the following output:

```
System load: 0.08      Processes:      105
Usage of /:  22.8% of 6.71GB   Users logged in: 0
Memory usage: 20%        IPv4 address for enX0: 172.31.35.147
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

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.

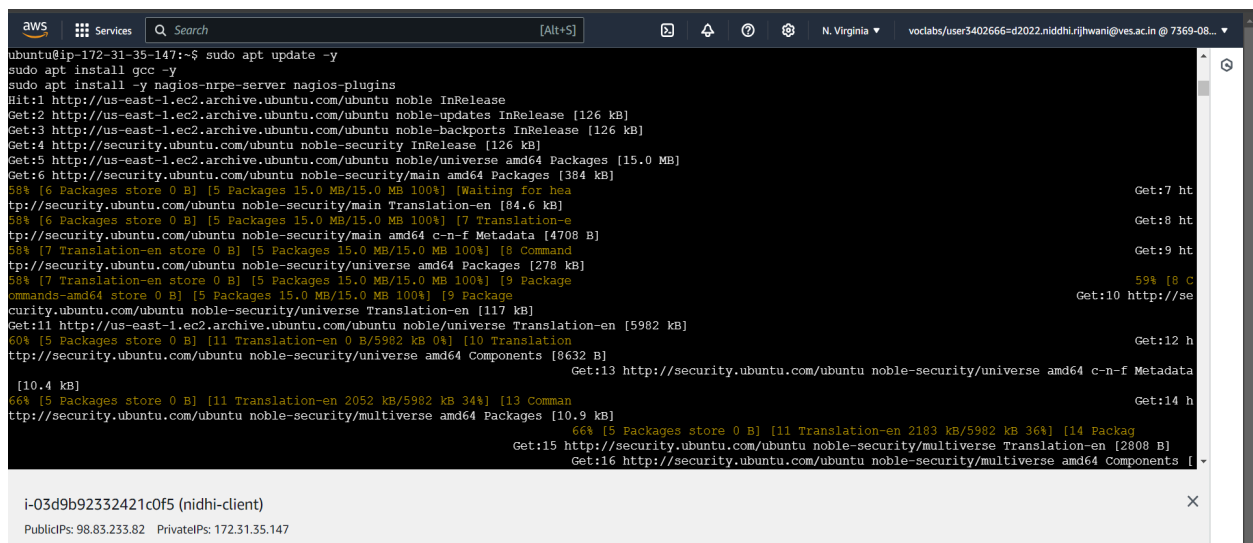
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-35-147:~$
```

At the bottom of the terminal window, the instance ID is shown: i-03d9b92332421c0f5 (nidhi-client). The public IP is 98.83.233.82 and the private IP is 172.31.35.147.

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



The screenshot shows the AWS CloudShell interface with the following output:

```
ubuntu@ip-172-31-35-147:~$ 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://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [384 kB]
58% [6 Packages store 0 B] [5 Packages 15.0 MB/15.0 MB 100%] [Waiting for hea
tp://security.ubuntu.com/ubuntu noble-security/main Translation-en [84.6 kB]
58% [6 Packages store 0 B] [5 Packages 15.0 MB/15.0 MB 100%] [7 Translation-e
tp://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4708 B]
58% [7 Translation-en store 0 B] [5 Packages 15.0 MB/15.0 MB 100%] [8 Command
tp://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [270 kB]
58% [7 Translation-en store 0 B] [5 Packages 15.0 MB/15.0 MB 100%] [9 Package
commands-amd64 store 0 B] [5 Packages 15.0 MB/15.0 MB 100%] [9 Package
curity.ubuntu.com/ubuntu noble-security/universe Translation-en [117 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
60% [5 Packages store 0 B] [11 Translation-en 0 B/5982 kB 0%] [10 Translation
tp://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
[10.4 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata
[10.4 kB]
66% [5 Packages store 0 B] [11 Translation-en 2052 kB/5982 kB 34%] [13 Comman
tp://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.9 kB]
66% [5 Packages store 0 B] [11 Translation-en 2183 kB/5982 kB 36%] [14 Packag
Get:15 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
Get:16 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [10.9 kB]
```

At the bottom of the terminal window, the instance ID is shown: i-03d9b92332421c0f5 (nidhi-client). The public IP is 98.83.233.82 and the private IP is 172.31.35.147.

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

```
ubuntu@ip-172-31-35-147:~$ sudo nano /etc/nagios/nrpe.cfg
```

```
# 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 mas
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currentl
# 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, 54.164.84.20
```

13. Restart the NRPE server

- `sudo systemctl restart nagios-nrpe-server`

```
ubuntu@ip-172-31-35-147:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-35-147:~$
```

i-03d9b92332421c0f5 (nidhi-client)

PublicIPs: 98.83.233.82 PrivateIPs: 172.31.35.147

14. Now, check your nagios dashboard and you'll see a new host being added. Click on Hosts.

The screenshot shows the Nagios web interface. The top navigation bar includes a 'Not secure' warning and the URL '54.164.84.20/nagios/'. The sidebar on the left contains links for 'General' (Home, Documentation), 'Current Status' (Tactical Overview, Map (Legacy), Hosts, Services), 'Host Groups' (Summary, Grid), 'Service Groups' (Summary, Grid), and 'Problems'.

The main content area displays the 'Current Network Status' (Last Updated: Sun Oct 13 09:04:00 UTC 2024, Updated every 90 seconds, Nagios® Core™ 4.4.9 - www.nagios.org, Logged in as nagiosadmin). It also shows 'Host Status Totals' (Up: 2, Down: 0, Unreachable: 0, Pending: 0) and 'Service Status Totals' (Ok: 12, Warning: 1, Unknown: 0, Critical: 3, Pending: 0).

The 'Host Status Details For All Host Groups' section shows a table with the following data:

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	10-13-2024 09:00:12	0d 0h 13m 10s	PING OK - Packet loss = 0%, RTA = 1.13 ms
localhost	UP	10-13-2024 08:59:16	0d 0h 44m 6s	PING OK - Packet loss = 0%, RTA = 0.03 ms

Results 1 - 2 of 2 Matching Hosts

