Name : Aditya kirtane , Div : D15C , Roll No. 26

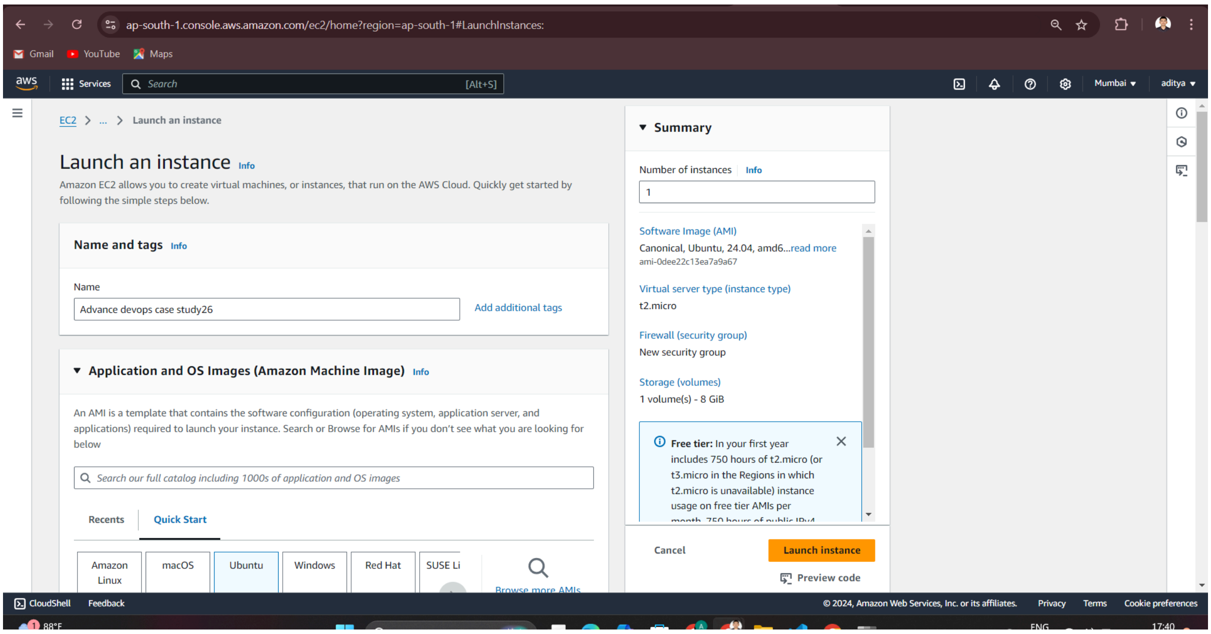
Case study

Topic Name: Automated Deployment with Monitoring ● Concepts Used: Jenkins, EC2, Nagios.

● **Problem Statement**: "Set up a Jenkins CI/CD pipeline to deploy a simple web application on an EC2 instance. Configure Nagios to monitor the deployed application's availability."

● **Tasks:** ○ Create a Jenkins pipeline that builds and deploys a sample web app to an EC2 instance. ○ Install and configure Nagios to monitor the HTTP status of the deployed application. ○ Verify the pipeline by triggering a build and checking the monitoring status in Nagios.

Step 1 : Launch an Ec2 instance



ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:

EC2 > ... > 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: Advance devops case study26

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

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more
ami-0dee22c13ea7a9a67

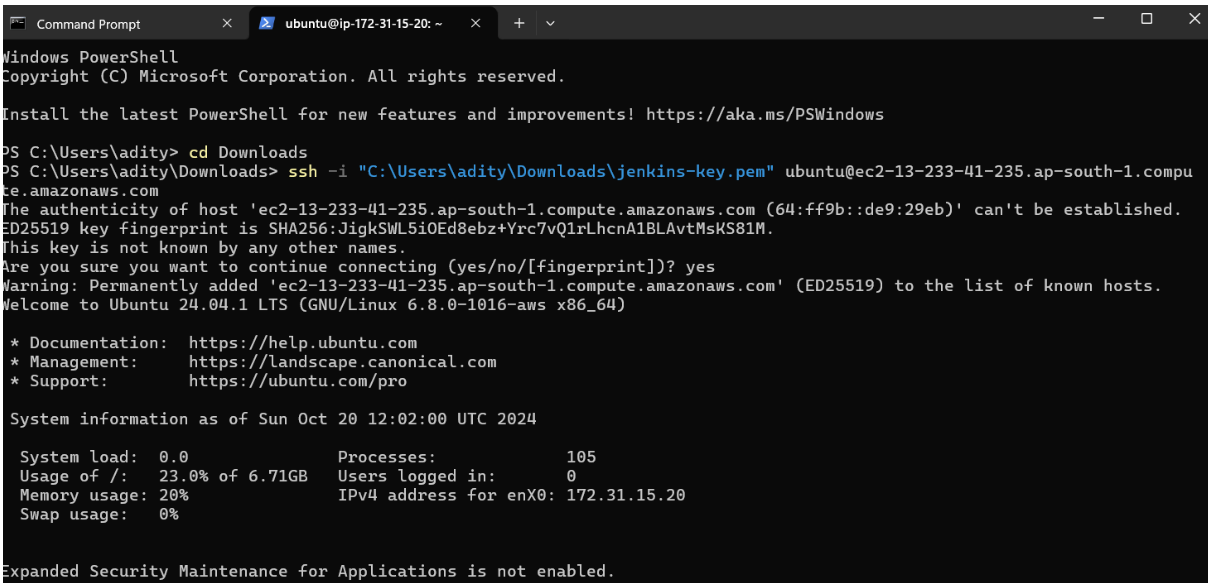
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

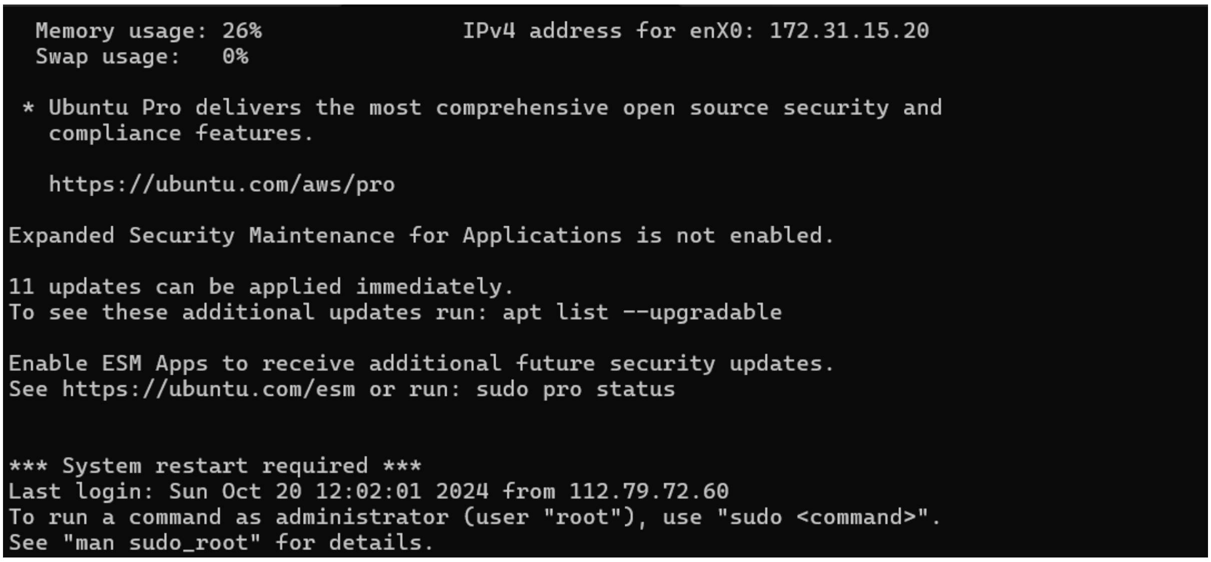
Storage (volumes): 1 volume(s) - 8 GiB

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

Step 2: Connect with ec2 instance

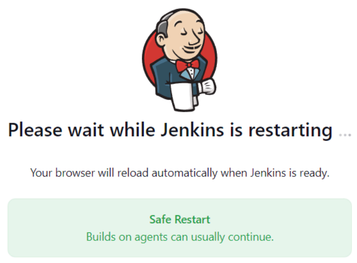


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\Users\adity> cd Downloads
PS C:\Users\adity\Downloads> ssh -i "C:\Users\adity\Downloads\jenkins-key.pem" ubuntu@ec2-13-233-41-235.ap-south-1.compu
te.amazonaws.com
The authenticity of host 'ec2-13-233-41-235.ap-south-1.compute.amazonaws.com (64:ff9b::de9:29eb)' can't be established.
ED25519 key fingerprint is SHA256:JigkSWL5iOEd8ebz+Yrc7vQ1rLhcnA1BLAvtMsKS81M.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-233-41-235.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

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

 System information as of Sun Oct 20 12:02:00 UTC 2024

  System load:  0.0               Processes:             105
  Usage of /:   23.0% of 6.71GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for enX0: 172.31.15.20
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.
```



```
  Memory usage: 26%               IPv4 address for enX0: 172.31.15.20
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

11 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

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


*** System restart required ***
Last login: Sun Oct 20 12:02:01 2024 from 112.79.72.60
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

Step 3 : Restart Jenkins

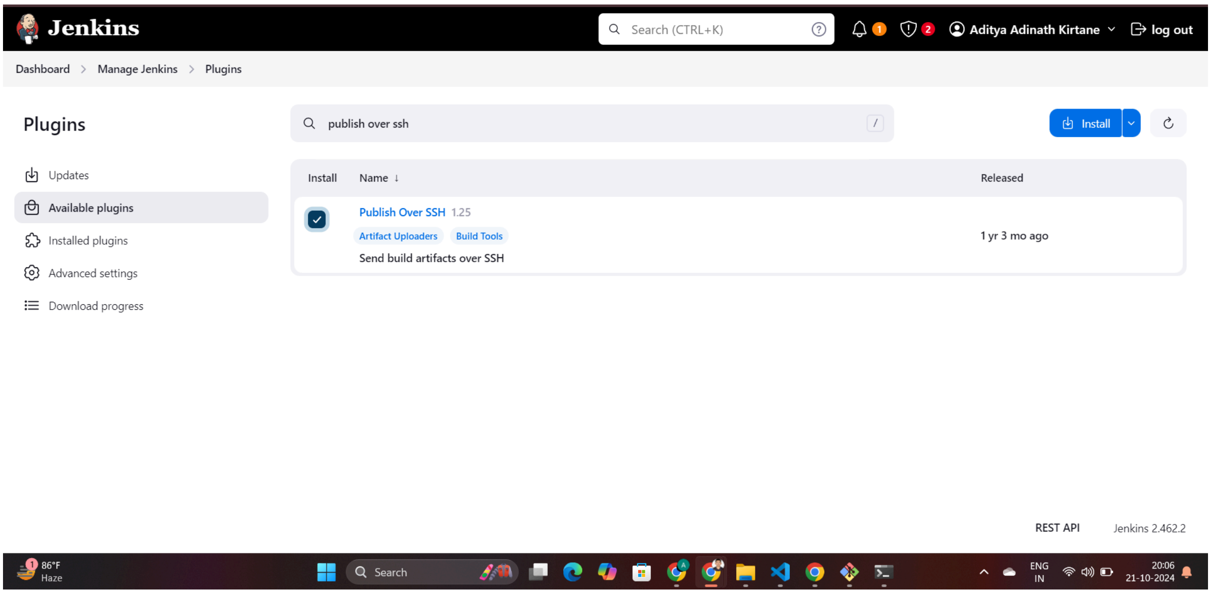


Please wait while Jenkins is restarting ...

Your browser will reload automatically when Jenkins is ready.

Safe Restart
Builds on agents can usually continue.

Step 4 : install plugins



Search: publish over ssh

Install	Name ↓	Released
☑	Publish Over SSH 1.25 Artifact Uploaders Build Tools Send build artifacts over SSH	1 yr 3 mo ago

Go to manage Jenkins ->credentials create a global credential for ssh.

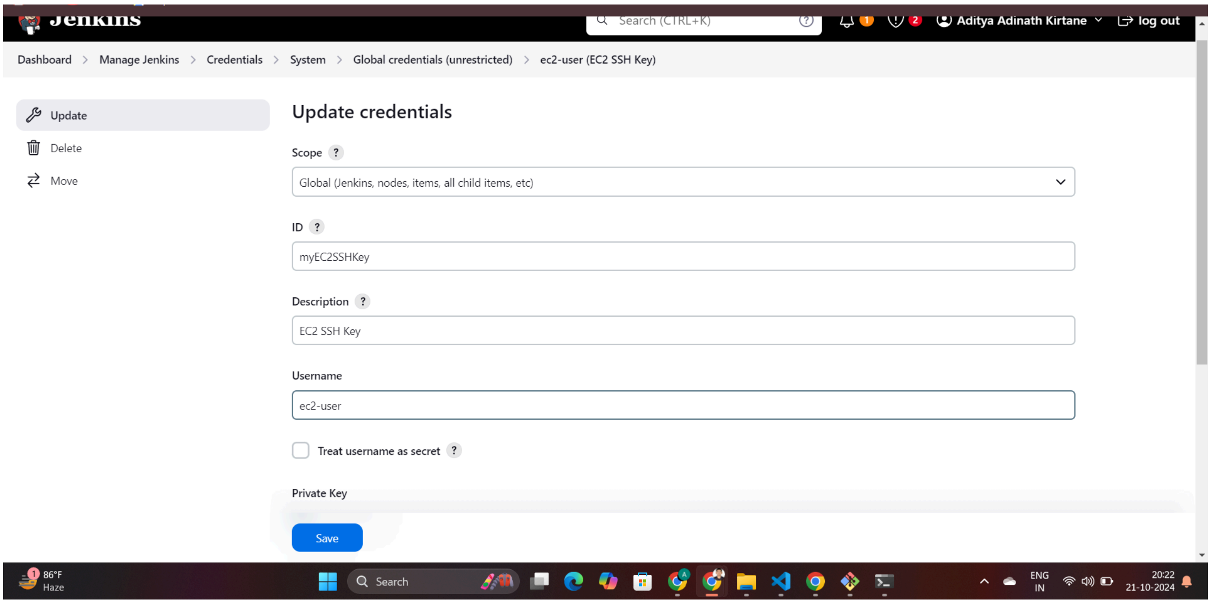
1.Add kind : SSH

2.Scope : Global

3.ID :Assign id

4.Username : ec2 user for amazon Linux

5.Private key


Jenkins
Search (CTRL+K)
Aditya Adinath Kirtane
log out
Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) > ec2-user (EC2 SSH Key)
Update
Delete
Move
Save

Update credentials

Scope: Global (Jenkins, nodes, items, all child items, etc)

ID: myEC2SSHKey

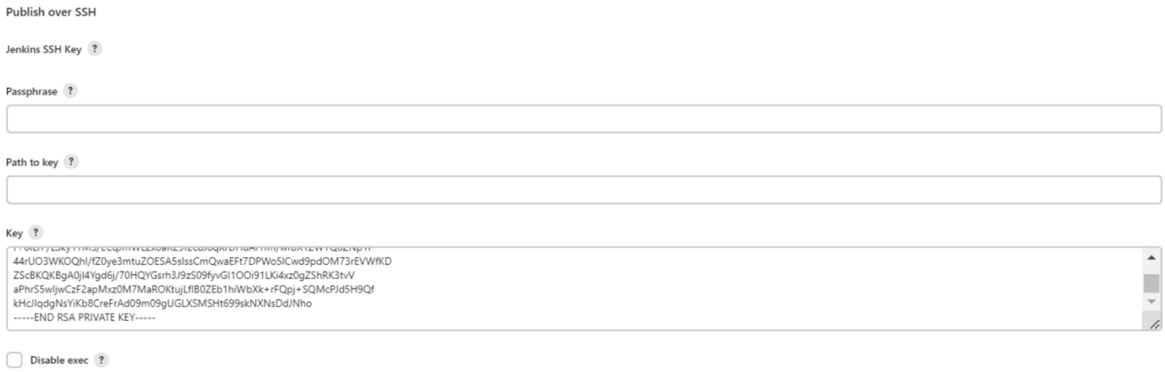
Description: EC2 SSH Key

Username: ec2-user

□ Treat username as secret

Private Key

Step 5 : Go to manage jenkins and scroll down you get publish over ssh section then copy and paste your key.



Publish over SSH

Jenkins SSH Key

Passphrase

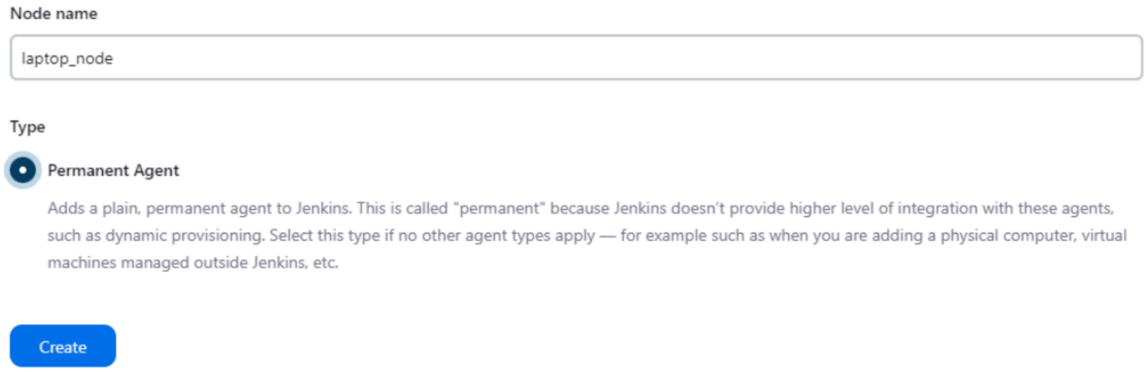
Path to key

Key

```
44rUO3WKOQhl/fZ0ye3mtuZOESA5slssCmQwaEFt7DPWo5ICwd9pdOM73rEVWfKD
ZScBKQKBgA0jI4Ygd6j/70HQYGsrh3J9zS09fyvGI1OOi91LKi4xz0gZShRK3tvV
aPhrS5wljwCzF2apMxz0M7MaROKtujLflB0ZEb1hiWbXk+rFQpj+SQMcPJd5H9Qf
kHcJlqdgNsYiKb8CreFrAd09m09gUGLXSMSHt699skNXNsDdJNho
-----END RSA PRIVATE KEY-----
```

□ Disable exec

Step 6 : As my built-in-node have limited space thus i need to use master-slave architecture. here I have created a node named “latop_node”



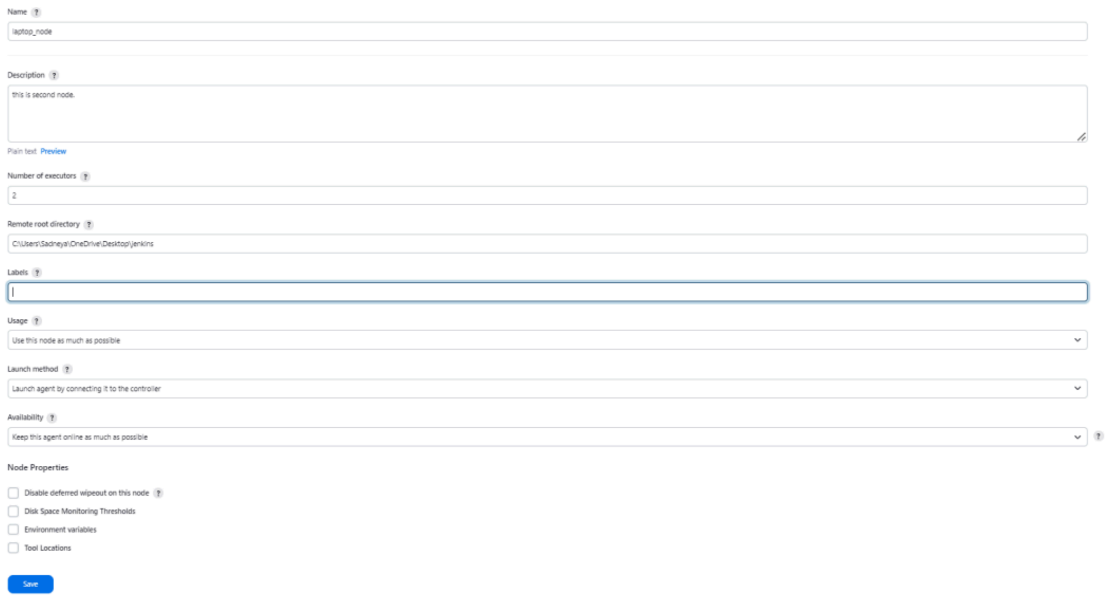
Node name

laptop_node

Type

● Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.



Name

laptop_node

Description

this is second node.

Number of executors

2

Remote root directory

C:\Users\Sadneya\OneDrive\Desktop\jenkins

Labels

Usage

Use this node as much as possible

Launch method

Launch agent by connecting it to the controller

Availability

Keep this agent online as much as possible

Node Properties

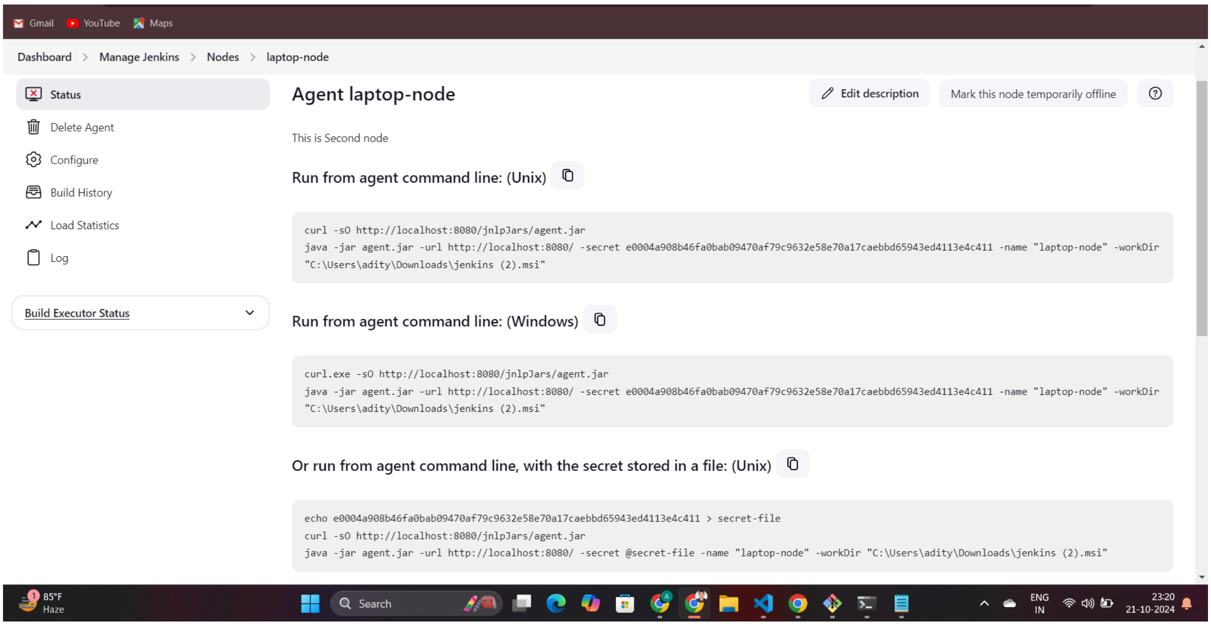
□ Disable deferred wipeout on this node

□ Disk Space Monitoring Thresholds

□ Environment variables

□ Tool Locations

Step 7 : create a node Agent laptop node.



Agent laptop-node

This is Second node

Run from agent command line: (Unix)

```
curl -sO http://localhost:8080/jnlpJars/agent.jar
java -jar agent.jar -url http://localhost:8080/ -secret e0004a908b46fa0bab09470af79c9632e58e70a17caebbd65943ed4113e4c411 -name "laptop-node" -workDir
"C:\Users\adity\Downloads\jenkins (2).msi"
```

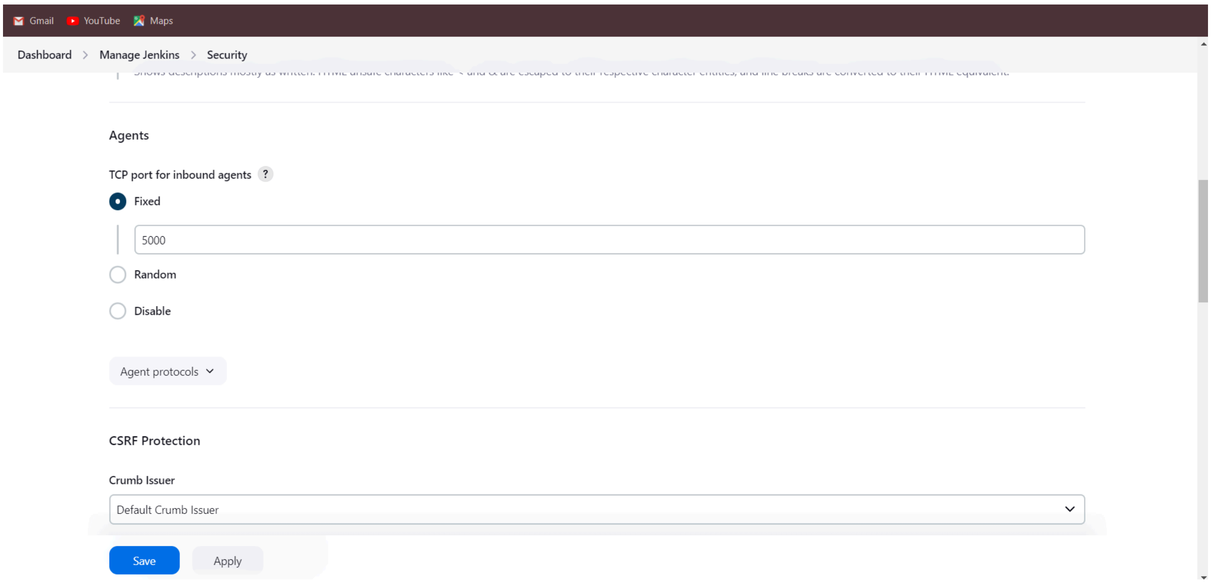
Run from agent command line: (Windows)

```
curl.exe -sO http://localhost:8080/jnlpJars/agent.jar
java -jar agent.jar -url http://localhost:8080/ -secret e0004a908b46fa0bab09470af79c9632e58e70a17caebbd65943ed4113e4c411 -name "laptop-node" -workDir
"C:\Users\adity\Downloads\jenkins (2).msi"
```

Or run from agent command line, with the secret stored in a file: (Unix)

```
echo e0004a908b46fa0bab09470af79c9632e58e70a17caebbd65943ed4113e4c411 > secret-file
curl -sO http://localhost:8080/jnlpJars/agent.jar
java -jar agent.jar -url http://localhost:8080/ -secret @secret-file -name "laptop-node" -workDir "C:\Users\adity\Downloads\jenkins (2).msi"
```

Step 8 :Go to manage jenkins->security and make agent TCP port fixed at 50000.that we before added in security group.



Agents

TCP port for inbound agents

● Fixed

5000

○ Random

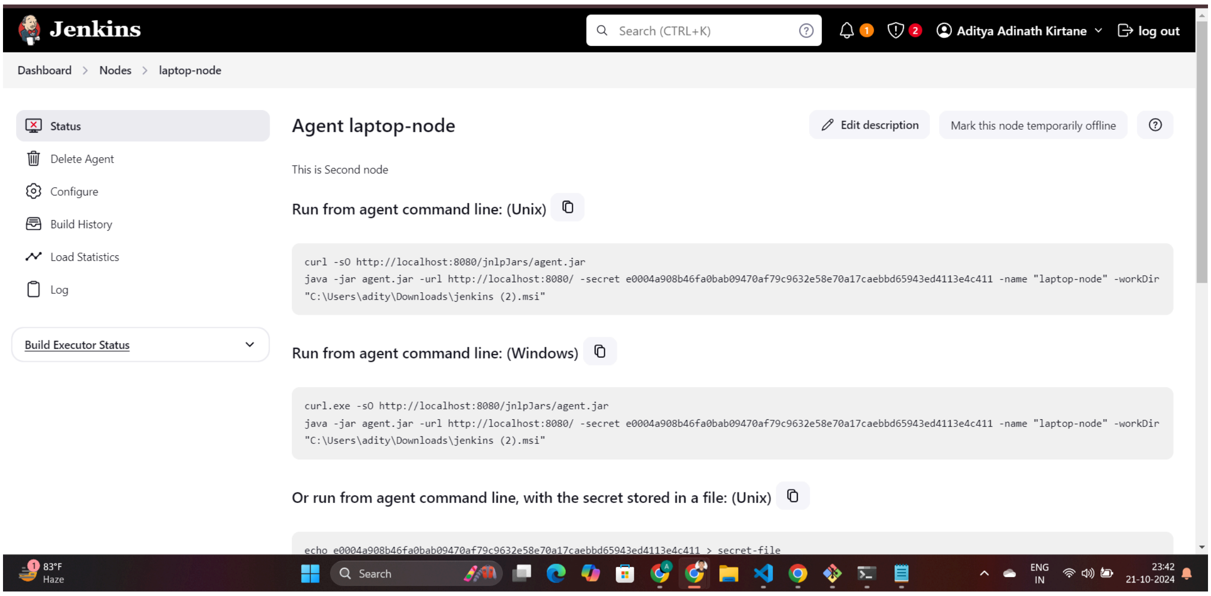
○ Disable

CSRF Protection

Crumb Issuer

Default Crumb Issuer

Step 9 : Now click on node you created. Here you will get the commands then copy paste this commands on your command prompt.



Agent laptop-node

This is Second node

Run from agent command line: (Unix)

```
curl -sO http://localhost:8080/jnlpJars/agent.jar
java -jar agent.jar -url http://localhost:8080/ -secret e0004a908b46fa0bab09470af79c9632e58e70a17caebbd65943ed4113e4c411 -name "laptop-node" -workDir
"C:\Users\adity\Downloads\jenkins (2).msi"
```

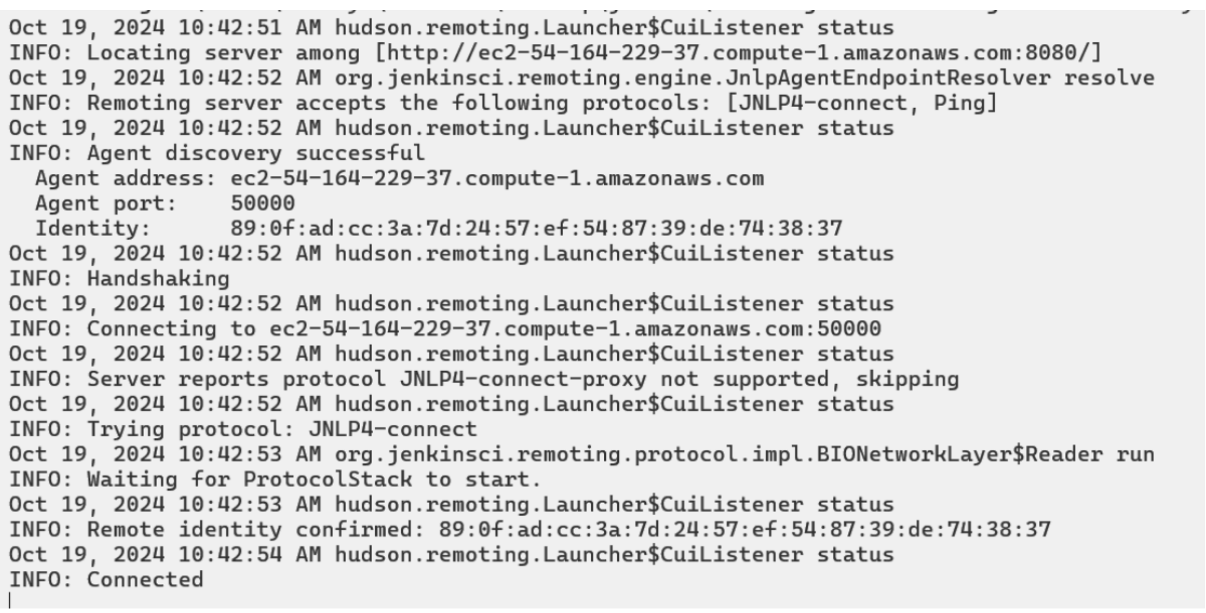
Run from agent command line: (Windows)

```
curl.exe -sO http://localhost:8080/jnlpJars/agent.jar
java -jar agent.jar -url http://localhost:8080/ -secret e0004a908b46fa0bab09470af79c9632e58e70a17caebbd65943ed4113e4c411 -name "laptop-node" -workDir
"C:\Users\adity\Downloads\jenkins (2).msi"
```

Or run from agent command line, with the secret stored in a file: (Unix)

```
echo e0004a908b46fa0bab09470af79c9632e58e70a17caebbd65943ed4113e4c411 > secret-file
```

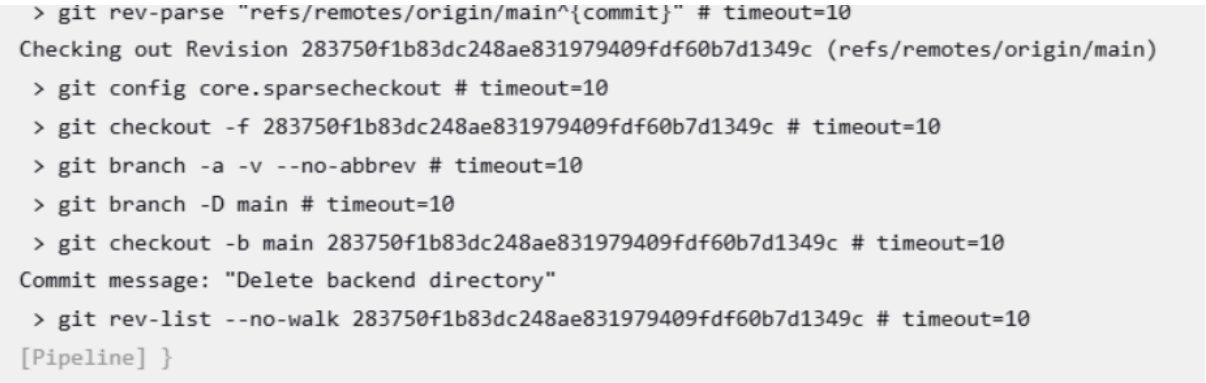
Step 10 : connected output



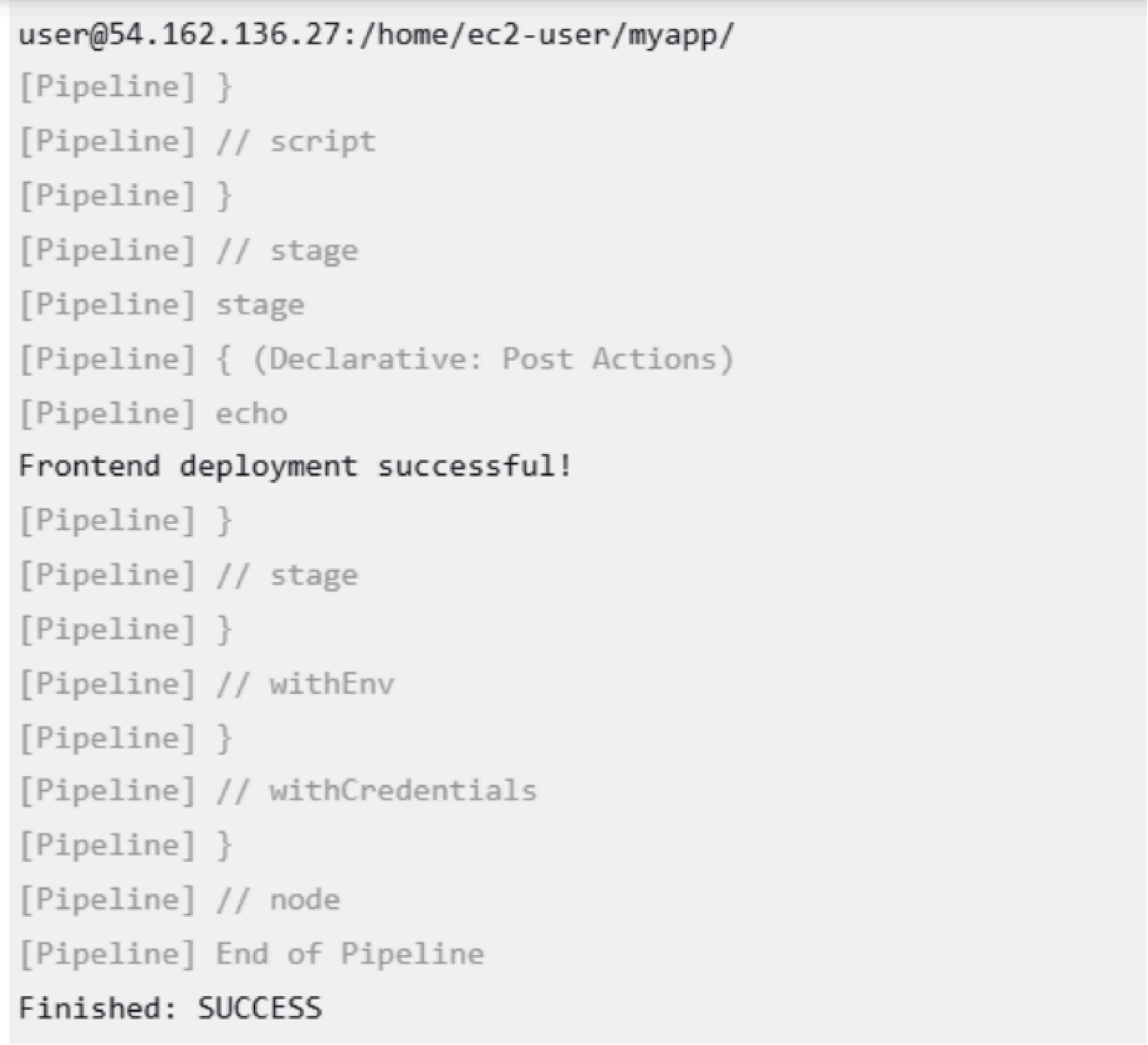
```
Oct 19, 2024 10:42:51 AM hudson.remoting.Launcher$CuiListener status
INFO: Locating server among [http://ec2-54-164-229-37.compute-1.amazonaws.com:8080/]
Oct 19, 2024 10:42:52 AM org.jenkinsci.remoting.engine.JnlpAgentEndpointResolver resolve
INFO: Remoting server accepts the following protocols: [JNLP4-connect, Ping]
Oct 19, 2024 10:42:52 AM hudson.remoting.Launcher$CuiListener status
INFO: Agent discovery successful
  Agent address: ec2-54-164-229-37.compute-1.amazonaws.com
  Agent port:    50000
  Identity:      89:0f:ad:cc:3a:7d:24:57:ef:54:87:39:de:74:38:37
Oct 19, 2024 10:42:52 AM hudson.remoting.Launcher$CuiListener status
INFO: Handshaking
Oct 19, 2024 10:42:52 AM hudson.remoting.Launcher$CuiListener status
INFO: Connecting to ec2-54-164-229-37.compute-1.amazonaws.com:50000
Oct 19, 2024 10:42:52 AM hudson.remoting.Launcher$CuiListener status
INFO: Server reports protocol JNLP4-connect-proxy not supported, skipping
Oct 19, 2024 10:42:52 AM hudson.remoting.Launcher$CuiListener status
INFO: Trying protocol: JNLP4-connect
Oct 19, 2024 10:42:53 AM org.jenkinsci.remoting.protocol.impl.BIONetworkLayer$Reader run
INFO: Waiting for ProtocolStack to start.
Oct 19, 2024 10:42:53 AM hudson.remoting.Launcher$CuiListener status
INFO: Remote identity confirmed: 89:0f:ad:cc:3a:7d:24:57:ef:54:87:39:de:74:38:37
Oct 19, 2024 10:42:54 AM hudson.remoting.Launcher$CuiListener status
INFO: Connected
```

1 . Deploy your code in the pipeline you created

```
pipeline {
    agent any
    environment {
        EC2_USER = 'ec2-user' // Your EC2 user
        EC2_IP = '54.162.136.27' // Your EC2 public IP address
without trailing slash
        SSH_KEY = credentials('myEC2SSHKey') // Your Jenkins
credential ID for SSH
    }
    stages {
        // Stage 1: Checkout code from GitHub
        stage('Checkout') {
            steps {
                script {
                    git branch: 'main', url: 'https://github.com/sadneya145/Leafling-copy.git'
                }
            }
        }

        // Stage 2: Clean previous installations
        stage('Clean Previous Installations') {
            steps {
                script {
                    dir('frontend') {
                        if (fileExists('node_modules')) {
                            bat 'rmdir /s /q node_modules'  // Windows-style command
                        }
                        if (fileExists('package-lock.json')) {
                            bat 'del package-lock.json'  // Windows-style delete
                        }
                    }
                }
            }
        }

        // Stage 3: Install frontend dependencies
        stage('Install Frontend Dependencies') {
            steps {
                script {
                    try {
                        dir('frontend') {
                            bat 'npm install'  // Replace `bat` with `sh` if using
a Linux node
                        }
                    } catch (Exception e) {
                        error "Dependency installation failed: ${e.message}"
                    }
                }
            }
        }

        // Stage 4: Build the frontend
        stage('Build Frontend') {
            steps {
                script {
                    dir('frontend') {
                        bat 'npm run build'  // Replace `bat` with `sh` for
Linux
                    }
                }
            }
        }

        // Stage 5: Deploy to EC2 instance
        stage('Deploy to EC2') {
            steps {
                script {
                    sshagent(['myEC2SSHKey']) {
                        sh """
                        ssh -o StrictHostKeyChecking=no
${EC2_USER}@${EC2_IP} 'mkdir -p /var/www/frontend'
                        scp -o StrictHostKeyChecking=no -r frontend/build/*
${EC2_USER}@${EC2_IP}:/var/www/frontend/
                        """
                    }
                }
            }
        }
    }

    // Post-stage cleanup or notifications
    post {
        success {
            echo 'Deployment successful!'
        }
        failure {
            echo 'Deployment failed!'
        }
    }
}
```

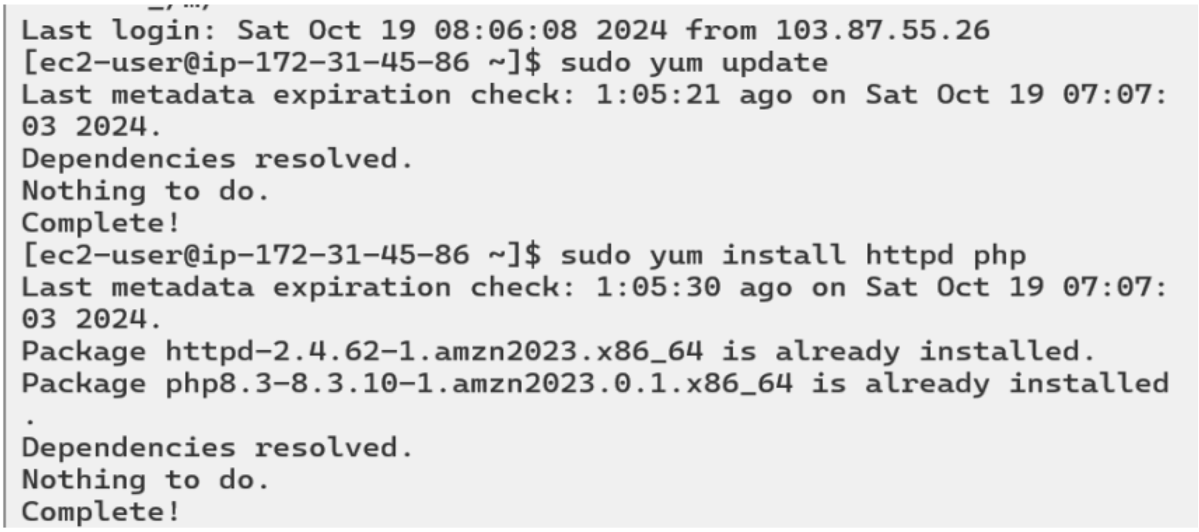


```
 > git rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision 283750f1b83dc248ae831979409fdf60b7d1349c (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 283750f1b83dc248ae831979409fdf60b7d1349c # timeout=10
 > git branch -a -v --no-abbrev # timeout=10
 > git branch -D main # timeout=10
 > git checkout -b main 283750f1b83dc248ae831979409fdf60b7d1349c # timeout=10
Commit message: "Delete backend directory"
 > git rev-list --no-walk 283750f1b83dc248ae831979409fdf60b7d1349c # timeout=10
[Pipeline] }
```



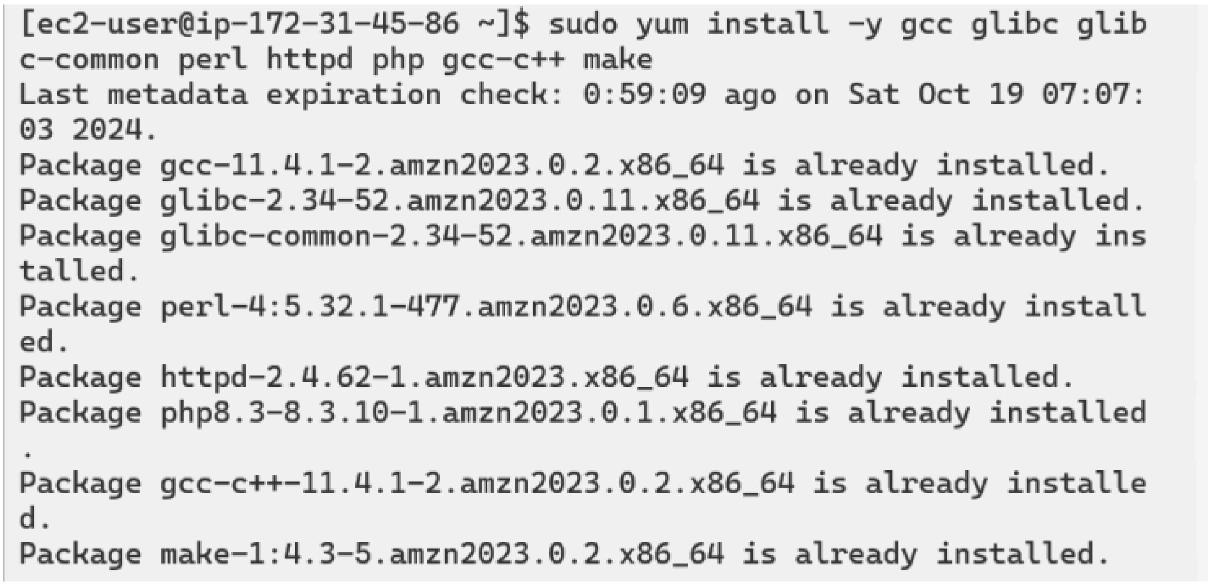
```
user@54.162.136.27:/home/ec2-user/myapp/
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Frontend deployment successful!
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

2.Installation Of Nagios For Monitoring:



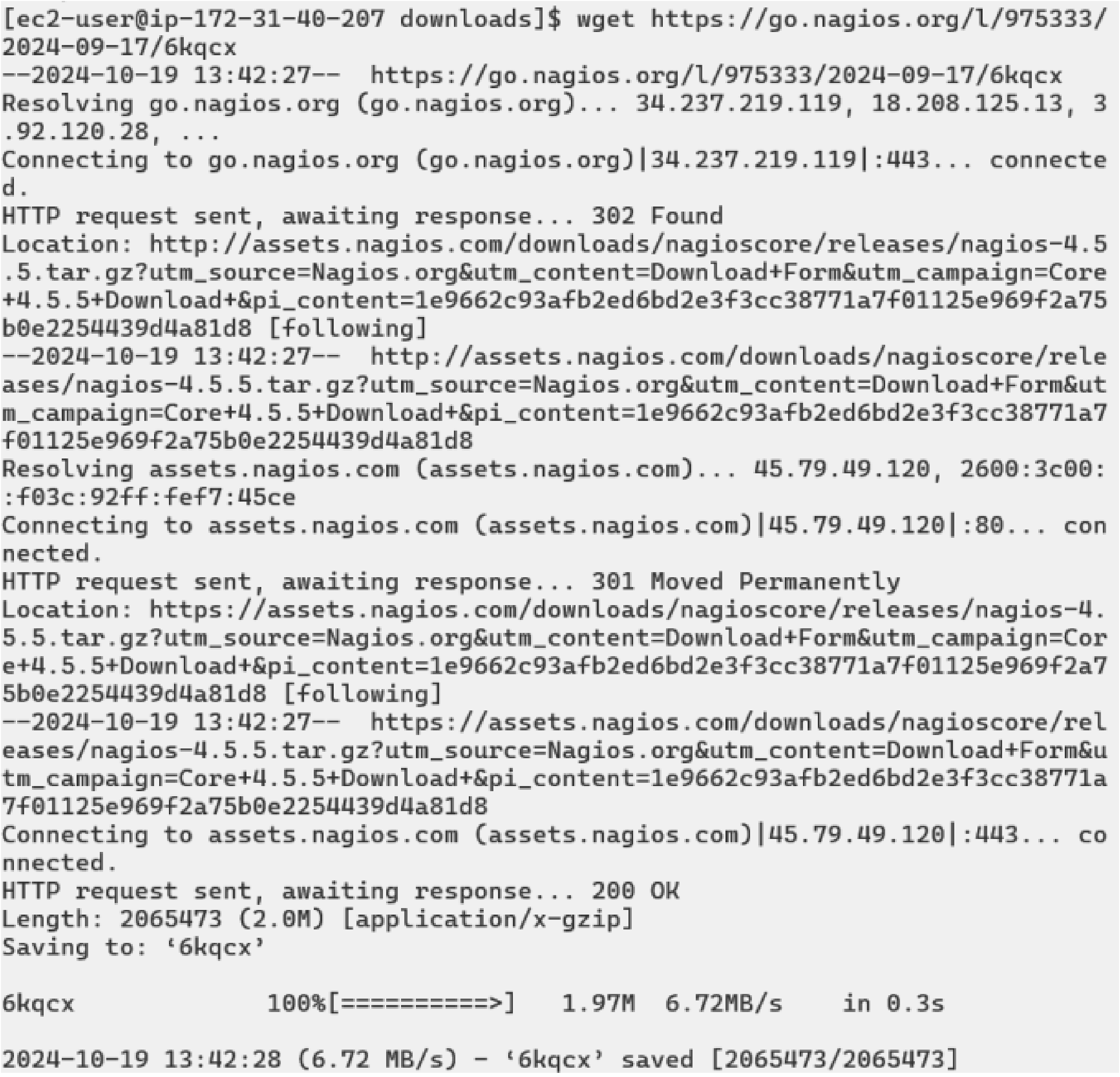
```
Last login: Sat Oct 19 08:06:08 2024 from 103.87.55.26
[ec2-user@ip-172-31-45-86 ~]$ sudo yum update
Last metadata expiration check: 1:05:21 ago on Sat Oct 19 07:07:
03 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-45-86 ~]$ sudo yum install httpd php
Last metadata expiration check: 1:05:30 ago on Sat Oct 19 07:07:
03 2024.
Package httpd-2.4.62-1.amzn2023.x86_64 is already installed.
Package php8.3-8.3.10-1.amzn2023.0.1.x86_64 is already installed
.
Dependencies resolved.
Nothing to do.
Complete!
```

3.Install Required Packages: sudo yum install gcc glibc glibc-common perl httpd php gcc-c++ make



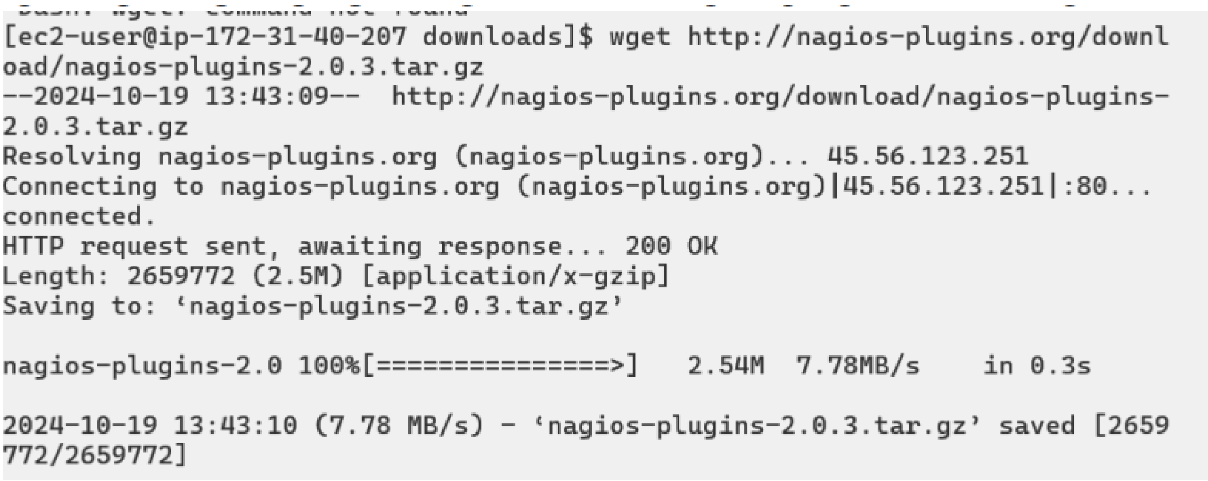
```
[ec2-user@ip-172-31-45-86 ~]$ sudo yum install -y gcc glibc glib
c-common perl httpd php gcc-c++ make
Last metadata expiration check: 0:59:09 ago on Sat Oct 19 07:07:
03 2024.
Package gcc-11.4.1-2.amzn2023.0.2.x86_64 is already installed.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already ins
talled.
Package perl-4:5.32.1-477.amzn2023.0.6.x86_64 is already install
ed.
Package httpd-2.4.62-1.amzn2023.x86_64 is already installed.
Package php8.3-8.3.10-1.amzn2023.0.1.x86_64 is already installed
.
Package gcc-c++-11.4.1-2.amzn2023.0.2.x86_64 is already installe
d.
Package make-1:4.3-5.amzn2023.0.2.x86_64 is already installed.
```

4.Download Nagios Source Files: wget https://go.nagios.org/l/975333/2024-09-17/6kqcx



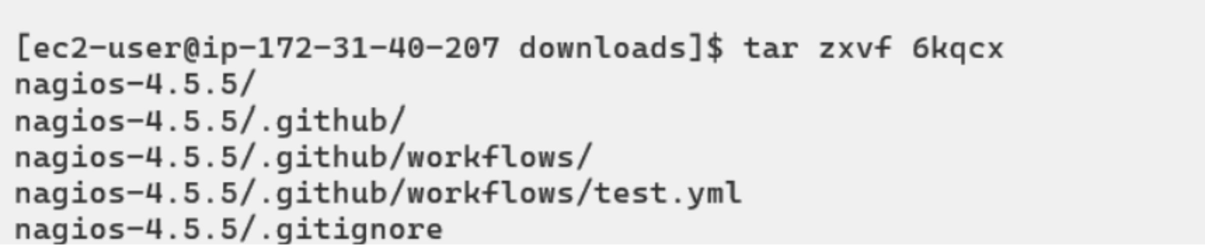
```
[ec2-user@ip-172-31-40-207 downloads]$ wget https://go.nagios.org/l/975333/
2024-09-17/6kqcx
--2024-10-19 13:42:27--  https://go.nagios.org/l/975333/2024-09-17/6kqcx
Resolving go.nagios.org (go.nagios.org)... 34.237.219.119, 18.208.125.13, 3
.92.120.28, ...
Connecting to go.nagios.org (go.nagios.org)|34.237.219.119|:443... connecte
d.
HTTP request sent, awaiting response... 302 Found
Location: http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5
.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core
+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75
b0e2254439d4a81d8 [following]
--2024-10-19 13:42:27--  http://assets.nagios.com/downloads/nagioscore/rele
ases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&ut
m_campaign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7
f01125e969f2a75b0e2254439d4a81d8
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00:
:f03c:92ff:fef7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:80... con
nected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.
5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Cor
e+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a7
5b0e2254439d4a81d8 [following]
--2024-10-19 13:42:27--  https://assets.nagios.com/downloads/nagioscore/rel
eases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&u
tm_campaign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a
7f01125e969f2a75b0e2254439d4a81d8
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... co
nnected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: ‘6kqcx’

6kqcx              100%[==========>]   1.97M  6.72MB/s    in 0.3s

2024-10-19 13:42:28 (6.72 MB/s) - ‘6kqcx’ saved [2065473/2065473]
```

8. Download Nagios Plugins: wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz



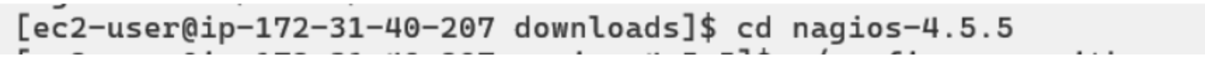
```
bash: wget: command not found
[ec2-user@ip-172-31-40-207 downloads]$ wget http://nagios-plugins.org/downl
oad/nagios-plugins-2.0.3.tar.gz
--2024-10-19 13:43:09--  http://nagios-plugins.org/download/nagios-plugins-
2.0.3.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:80...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 2659772 (2.5M) [application/x-gzip]
Saving to: ‘nagios-plugins-2.0.3.tar.gz’

nagios-plugins-2.0 100%[===============>]   2.54M  7.78MB/s    in 0.3s

2024-10-19 13:43:10 (7.78 MB/s) - ‘nagios-plugins-2.0.3.tar.gz’ saved [2659
772/2659772]
```

9. Unzip the Nagios Source Files: tar zxvf 6kqcx



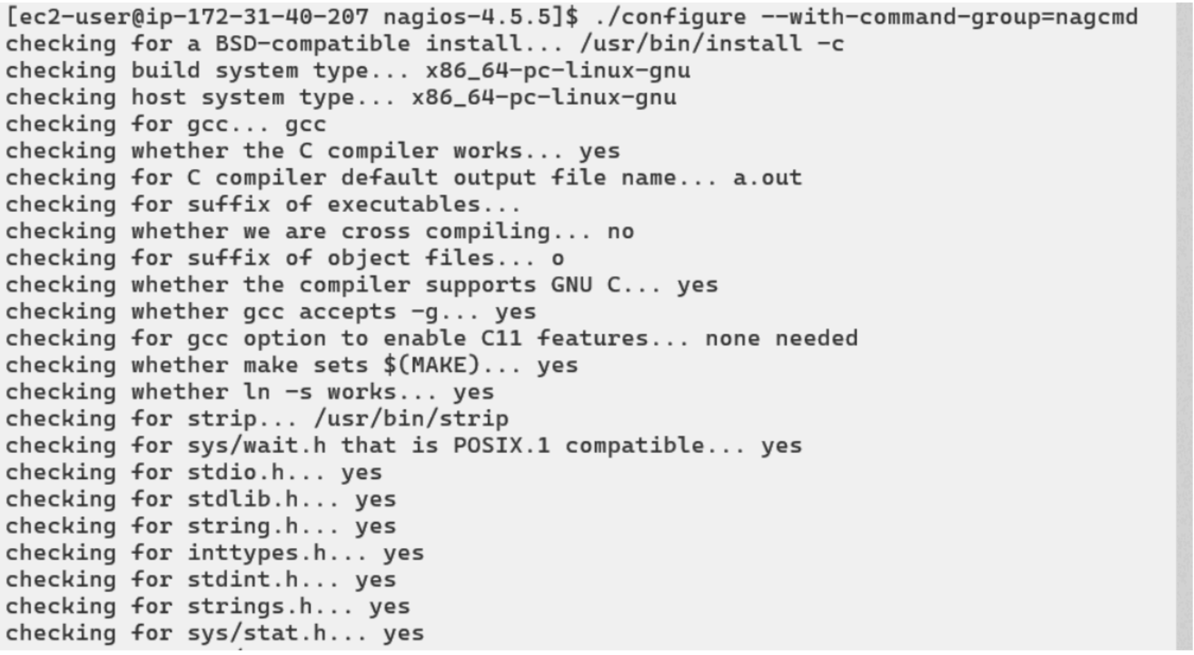
```
[ec2-user@ip-172-31-40-207 downloads]$ tar zxvf 6kqcx
nagios-4.5.5/
nagios-4.5.5/.github/
nagios-4.5.5/.github/workflows/
nagios-4.5.5/.github/workflows/test.yml
nagios-4.5.5/.gitignore
```

cd nagios-4.5.5

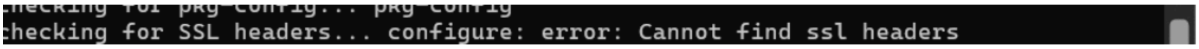


```
[ec2-user@ip-172-31-40-207 downloads]$ cd nagios-4.5.5
```

10. Run Configuration Script: ./configure --with-command-group=nagcmd

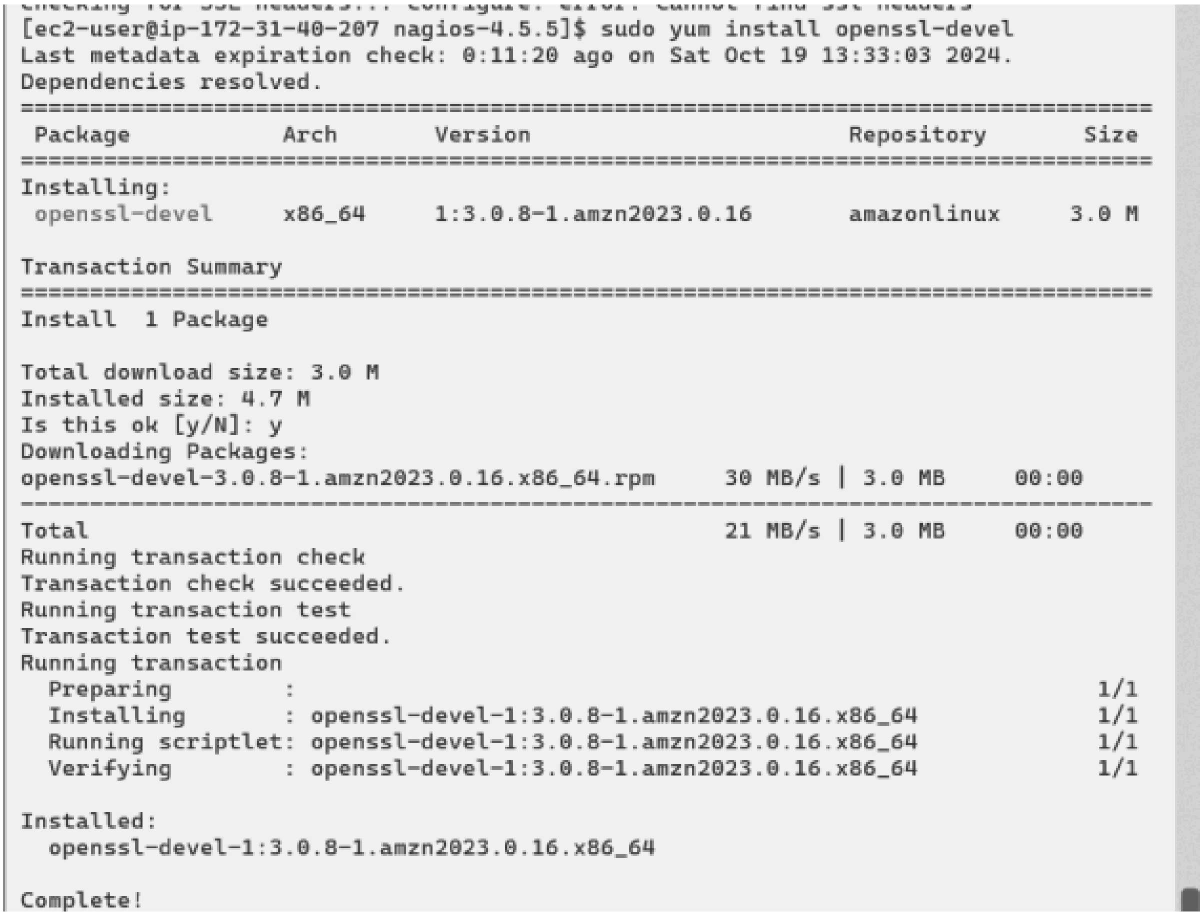


```
[ec2-user@ip-172-31-40-207 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for strings.h... yes
checking for sys/stat.h... yes
```



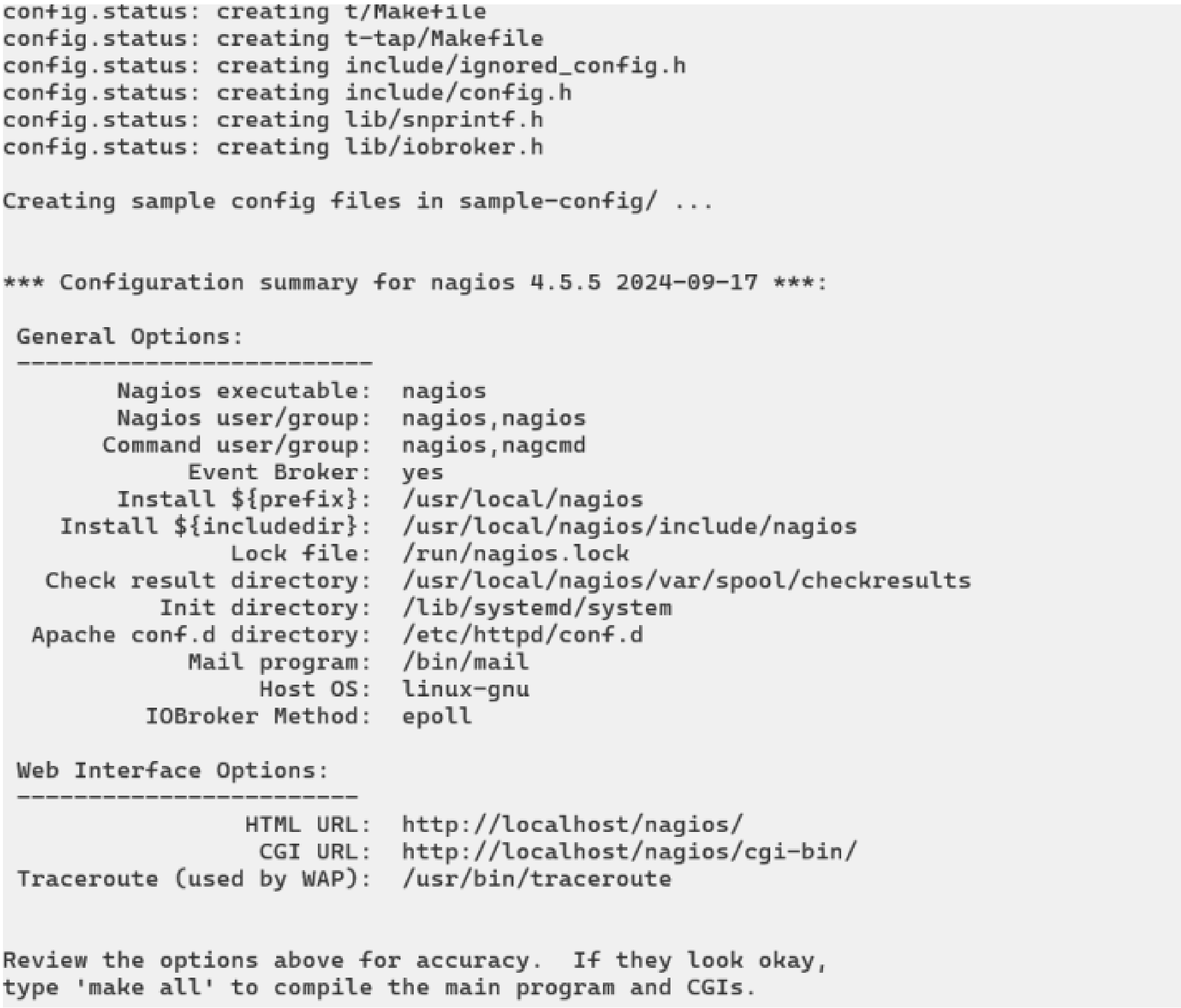
```
checking for pkg-config... pkg-config
checking for SSL headers... configure: error: Cannot find ssl headers
```

11.Install SSL Development Package:



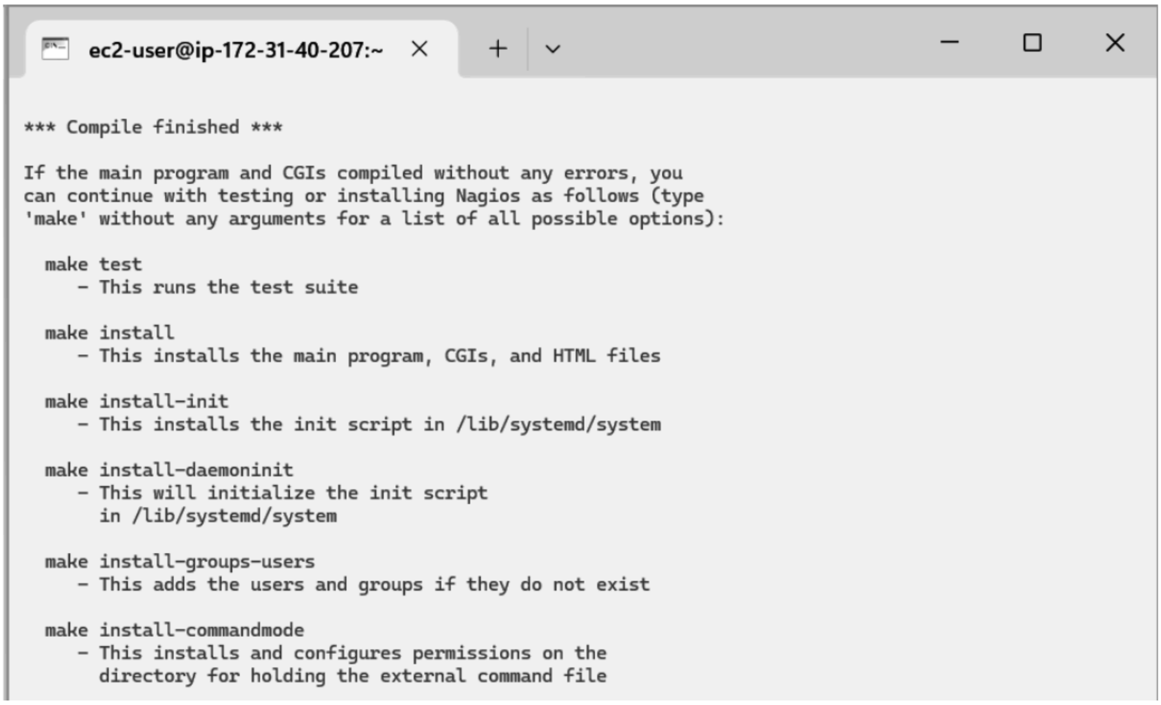
```
Checking for SSL headers...: configure: error: Cannot find ssl headers
[ec2-user@ip-172-31-40-207 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:11:20 ago on Sat Oct 19 13:33:03 2024.
Dependencies resolved.
==================================================================================
 Package            Arch       Version                       Repository       Size
==================================================================================
Installing:
 openssl-devel      x86_64     1:3.0.8-1.amzn2023.0.16       amazonlinux     3.0 M

Transaction Summary
==================================================================================
Install  1 Package

Total download size: 3.0 M
Installed size: 4.7 M
Is this ok [y/N]: y
Downloading Packages:
openssl-devel-3.0.8-1.amzn2023.0.16.x86_64.rpm     30 MB/s | 3.0 MB     00:00
----------------------------------------------------------------------------------
Total                                              21 MB/s | 3.0 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                         1/1
  Installing       : openssl-devel-1:3.0.8-1.amzn2023.0.16.x86_64            1/1
  Running scriptlet: openssl-devel-1:3.0.8-1.amzn2023.0.16.x86_64            1/1
  Verifying        : openssl-devel-1:3.0.8-1.amzn2023.0.16.x86_64            1/1

Installed:
  openssl-devel-1:3.0.8-1.amzn2023.0.16.x86_64

Complete!
```

12. Rerun Configuration Script: You will get final output like this ./configure --with-command-group=nagcmd

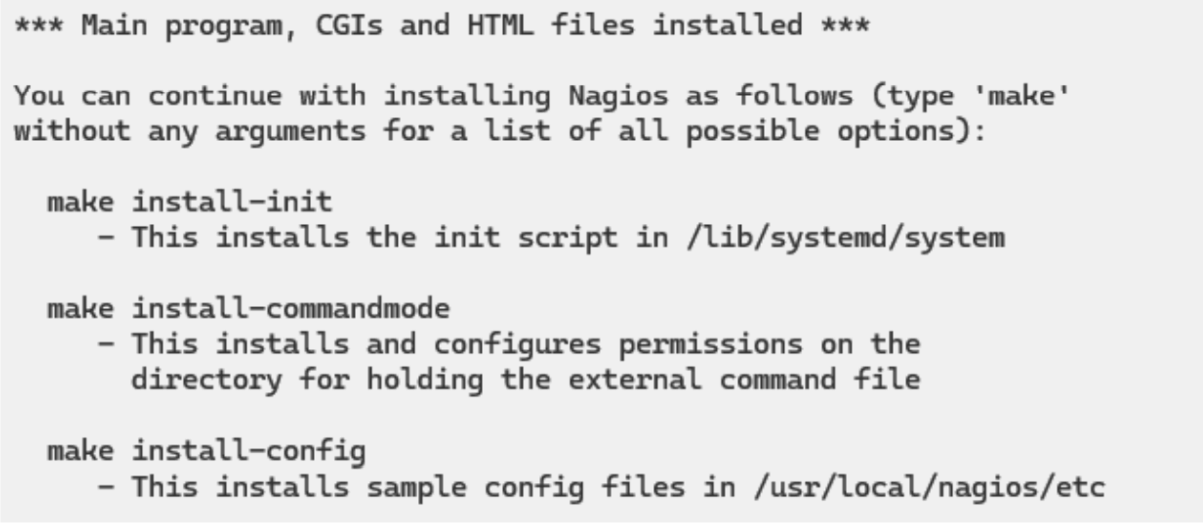


```
config.status: creating t/Makefile
config.status: creating t-tap/Makefile
config.status: creating include/ignored_config.h
config.status: creating include/config.h
config.status: creating lib/snprintf.h
config.status: creating lib/iobroker.h

Creating sample config files in sample-config/ ...


*** Configuration summary for nagios 4.5.5 2024-09-17 ***:

 General Options:
 -------------------------
        Nagios executable:  nagios
        Nagios user/group:  nagios,nagios
       Command user/group:  nagios,nagcmd
             Event Broker:  yes
        Install ${prefix}:  /usr/local/nagios
    Install ${includedir}:  /usr/local/nagios/include/nagios
                Lock file:  /run/nagios.lock
   Check result directory:  /usr/local/nagios/var/spool/checkresults
           Init directory:  /lib/systemd/system
  Apache conf.d directory:  /etc/httpd/conf.d
             Mail program:  /bin/mail
                  Host OS:  linux-gnu
          IOBroker Method:  epoll

 Web Interface Options:
 ------------------------
                 HTML URL:  http://localhost/nagios/
                  CGI URL:  http://localhost/nagios/cgi-bin/
 Traceroute (used by WAP):  /usr/bin/traceroute


Review the options above for accuracy.  If they look okay,
type 'make all' to compile the main program and CGIs.
```

13. Install Nagios: sudo make install

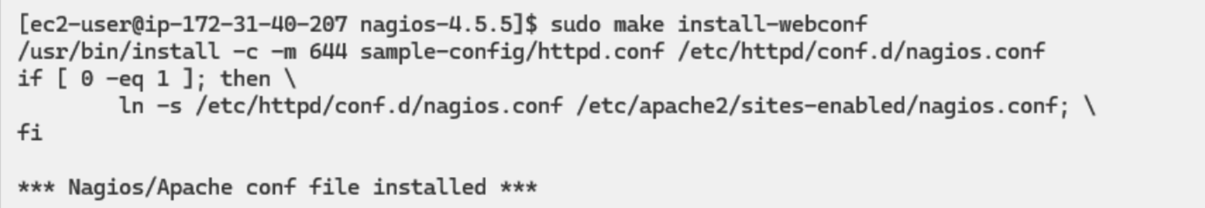


```
*** Compile finished ***

If the main program and CGIs compiled without any errors, you
can continue with testing or installing Nagios as follows (type
'make' without any arguments for a list of all possible options):

  make test
     - This runs the test suite

  make install
     - This installs the main program, CGIs, and HTML files

  make install-init
     - This installs the init script in /lib/systemd/system

  make install-daemoninit
     - This will initialize the init script
       in /lib/systemd/system

  make install-groups-users
     - This adds the users and groups if they do not exist

  make install-commandmode
     - This installs and configures permissions on the
       directory for holding the external command file
```



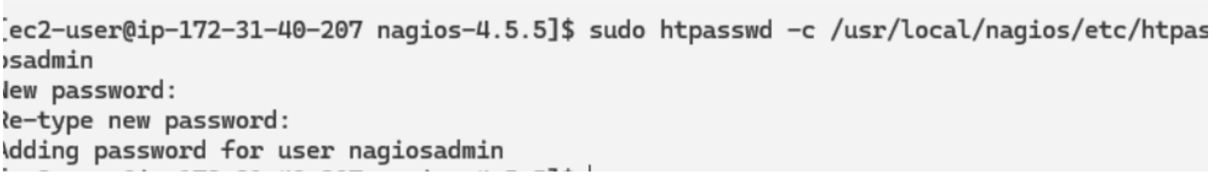
```
*** Main program, CGIs and HTML files installed ***

You can continue with installing Nagios as follows (type 'make'
without any arguments for a list of all possible options):

  make install-init
     - This installs the init script in /lib/systemd/system

  make install-commandmode
     - This installs and configures permissions on the
       directory for holding the external command file

  make install-config
     - This installs sample config files in /usr/local/nagios/etc
```

14. Configure Nagios Web Interface: sudo make install-webconf



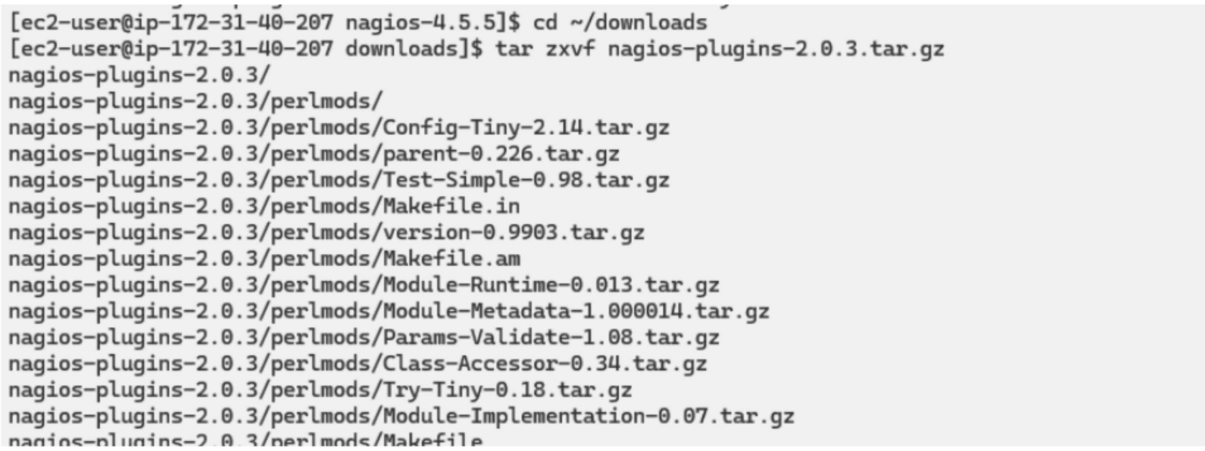
```
[ec2-user@ip-172-31-40-207 nagios-4.5.5]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
        ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

*** Nagios/Apache conf file installed ***
```

15. Create Nagios Admin Account: sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin



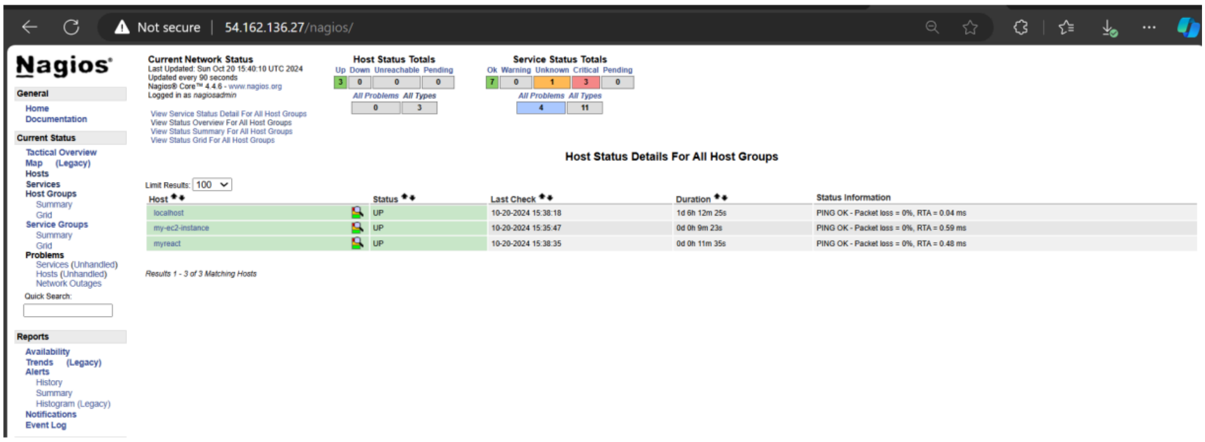
```
ec2-user@ip-172-31-40-207 nagios-4.5.5]$ sudo htpasswd -c /usr/local/nagios/etc/htpas
sadmin
lew password:
Re-type new password:
Adding password for user nagiosadmin
```

17. Unzip Nagios Plugins: cd ~/downloads



```
[ec2-user@ip-172-31-40-207 nagios-4.5.5]$ cd ~/downloads
[ec2-user@ip-172-31-40-207 downloads]$ tar zxvf nagios-plugins-2.0.3.tar.gz
nagios-plugins-2.0.3/
nagios-plugins-2.0.3/perlmods/
nagios-plugins-2.0.3/perlmods/Config-Tiny-2.14.tar.gz
nagios-plugins-2.0.3/perlmods/parent-0.226.tar.gz
nagios-plugins-2.0.3/perlmods/Test-Simple-0.98.tar.gz
nagios-plugins-2.0.3/perlmods/Makefile.in
nagios-plugins-2.0.3/perlmods/version-0.9903.tar.gz
nagios-plugins-2.0.3/perlmods/Makefile.am
nagios-plugins-2.0.3/perlmods/Module-Runtime-0.013.tar.gz
nagios-plugins-2.0.3/perlmods/Module-Metadata-1.000014.tar.gz
nagios-plugins-2.0.3/perlmods/Params-Validate-1.08.tar.gz
nagios-plugins-2.0.3/perlmods/Class-Accessor-0.34.tar.gz
nagios-plugins-2.0.3/perlmods/Try-Tiny-0.18.tar.gz
nagios-plugins-2.0.3/perlmods/Module-Implementation-0.07.tar.gz
nagios-plugins-2.0.3/perlmods/Makefile
```

6. Nagios page: Go back to your nagios page you will se output.here you will see my-ec2-instance.



Current Network Status
Last Updated: Sun Oct 20 15:40:10 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.4.6 - www.nagios.org
Logged in as *nagiosadmin*

Host Status Totals

Up	Down	Unreachable	Pending
3	0	0	0

All Problems	All Types
0	3

Service Status Totals

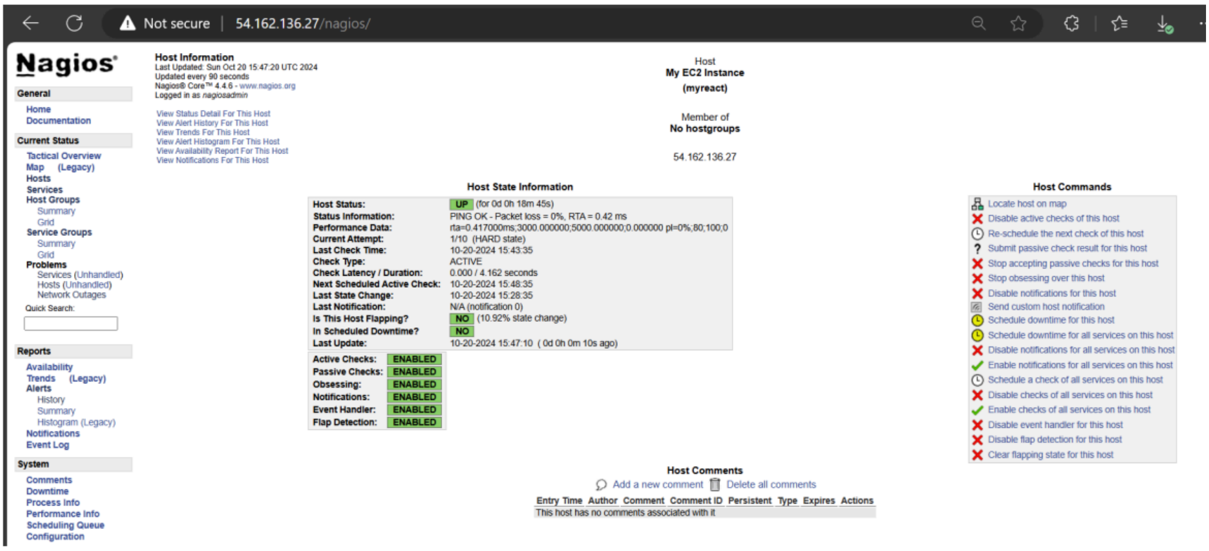
Ok	Warning	Unknown	Critical	Pending
7	0	1	3	0

All Problems	All Types
4	11

Host Status Details For All Host Groups

Host	Status	Last Check	Duration	Status Information
localhost	UP	10-20-2024 15:38:18	1d 6h 12m 25s	PING OK - Packet loss = 0%, RTA = 0.04 ms
my-ec2-instance	UP	10-20-2024 15:35:47	0d 0h 9m 23s	PING OK - Packet loss = 0%, RTA = 0.59 ms
myreact	UP	10-20-2024 15:38:35	0d 0h 11m 35s	PING OK - Packet loss = 0%, RTA = 0.48 ms

Results 1 - 3 of 3 Matching Hosts



Host Information
Last Updated: Sun Oct 20 15:47:20 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.4.6 - www.nagios.org
Logged in as *nagiosadmin*

Host
My EC2 Instance
(myreact)

Member of
No hostgroups

54.162.136.27

Host State Information

Host Status:	UP (for 0d 0h 18m 45s)
Status Information:	PING OK - Packet loss = 0%, RTA = 0.42 ms
Performance Data:	rta=0.417000ms;3000.000000;5000.000000;0.000000 pl=0%;80;100;0
Current Attempt:	1/10 (HARD state)
Last Check Time:	10-20-2024 15:43:35
Check Type:	ACTIVE
Check Latency / Duration:	0.000 / 4.162 seconds
Next Scheduled Active Check:	10-20-2024 15:48:35
Last State Change:	10-20-2024 15:28:35
Last Notification:	N/A (notification 0)
Is This Host Flapping?	NO (10.92% state change)
In Scheduled Downtime?	NO
Last Update:	10-20-2024 15:47:10 (0d 0h 0m 10s ago)

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

Host Commands

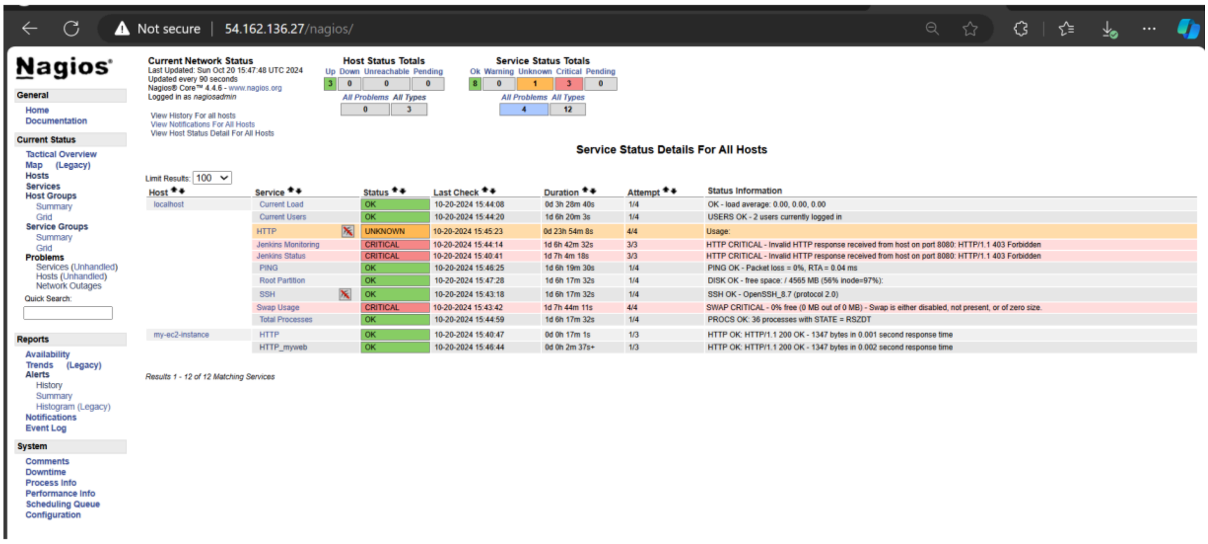
- Locate host on map
- ✗ Disable active checks of this host
- Re-schedule the next check of this host
- ? Submit passive check result for this host
- ✗ Stop accepting passive checks for this host
- ✗ Stop obsessing over this host
- ✗ Disable notifications for this host
- Send custom host notification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

Now click on services on left sidebar you will get detailed information about network status.



Current Network Status
Last Updated: Sun Oct 20 15:47:48 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.4.6 - www.nagios.org
Logged in as *nagiosadmin*

Host Status Totals

Up	Down	Unreachable	Pending
3	0	0	0

All Problems	All Types
0	3

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
8	0	1	3	0

All Problems	All Types
4	12

Service Status Details For All Hosts

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	10-20-2024 15:44:08	0d 3h 28m 40s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	10-20-2024 15:44:20	1d 6h 20m 3s	1/4	USERS OK - 2 users currently logged in
	HTTP	UNKNOWN	10-20-2024 15:45:23	0d 23h 54m 8s	4/4	Usage:
	Jenkins Monitoring	CRITICAL	10-20-2024 15:44:14	1d 6h 42m 32s	3/3	HTTP CRITICAL - Invalid HTTP response received from host on port 8080: HTTP/1.1 403 Forbidden
	Jenkins Status	CRITICAL	10-20-2024 15:40:41	1d 7h 4m 18s	3/3	HTTP CRITICAL - Invalid HTTP response received from host on port 8080: HTTP/1.1 403 Forbidden
	PING	OK	10-20-2024 15:46:25	1d 6h 19m 30s	1/4	PING OK - Packet loss = 0%, RTA = 0.04 ms
	Root Partition	OK	10-20-2024 15:47:28	1d 6h 17m 32s	1/4	DISK OK - free space: / 4565 MB (56% inode=97%):
	SSH	OK	10-20-2024 15:43:18	1d 6h 17m 32s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)
	Swap Usage	CRITICAL	10-20-2024 15:43:42	1d 7h 44m 11s	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	10-20-2024 15:44:59	1d 6h 17m 32s	1/4	PROCS OK: 36 processes with STATE = RSZDT
my-ec2-instance	HTTP	OK	10-20-2024 15:40:47	0d 0h 17m 1s	1/3	HTTP OK: HTTP/1.1 200 OK - 1347 bytes in 0.001 second response time
	HTTP_myweb	OK	10-20-2024 15:46:44	0d 0h 2m 37s+	1/3	HTTP OK: HTTP/1.1 200 OK - 1347 bytes in 0.002 second response time

Results 1 - 12 of 12 Matching Services

Conclusion: This case study involved setting up an automated CI/CD pipeline with Jenkins to deploy a web app on AWS EC2, and using Nagios for monitoring. We faced challenges like SSH configuration, limited Jenkins disk space, and SSL issues with Nagios, which were resolved through security adjustments and required package installations. Key takeaways included the importance of secure automation and effective monitoring for maintaining a reliable deployment process.