

# Automating NGINX Installation on RHEL using Ansible In AWS EC2!

🔧 In this post, I'll walk you through how I automated the installation of NGINX on a Red Hat Enterprise Linux (RHEL) server using Ansible.

HOW TO INSTALL NGINX USING ANSIBLE




ANSIBLE

+



NGINX



 Amazon EC2 (Elastic Compute Cloud) allows you to create virtual servers on the cloud. To launch an instance:

**Step1:** Navigate to EC2 in the AWS Management Console.:

**Step2:** Click on Launch Instance.:

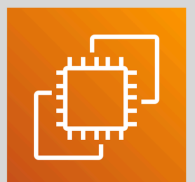
**Step3:** Choose RHEL its your choice to choose AMI in this case i used RHEL. You can use Amazon linux ,Ubuntu, any one you want but make sure the commands are diffrent for different OS

**Step4:** Select the t2.micro instance type .

**Step5:** Configure storage and security groups (allow port 22 for SSH and port 80 for HTTP).

**Step6:** Launch the instance .

**Step7:** Connect instance, i used MOBAEXTREAM .



Amazon Elastic Compute Cloud  
(Amazon EC2)

# Writing the Ansible Playbook

**Step1:** After connecting the instance install ansible in both Master and Slave nodes. using “sudo dnf install ansible -y command”

**Step2:** create a file called “inventory.ini” and put your slave private ip in the file and give name for them as group i given as WEB .

**Step3:** Create a file named as nginx.yaml and write play book as mentioned below .

```
---
- name: installing nginx
  hosts: web
  become: 'yes'
  tasks:
    - name: installing nginx
      ansible.builtin.dnf: # this command is used to install packages on RHEL
        name: nginx
        state: latest
    - name: start and enable nginx
      ansible.builtin.service:
        name: nginx
        state: started
        enabled: 'yes'
```

# Executing the Ansible Playbook

**Step1:** After creating play book we need to execute for executing play book use command mentioned below

**Step2:** `ansible-playbook -i inventory.ini -e ansible_user=ec2-user -e ansible_password=DevOps321 nginx.yaml` “- this command works on RHEL OS only”

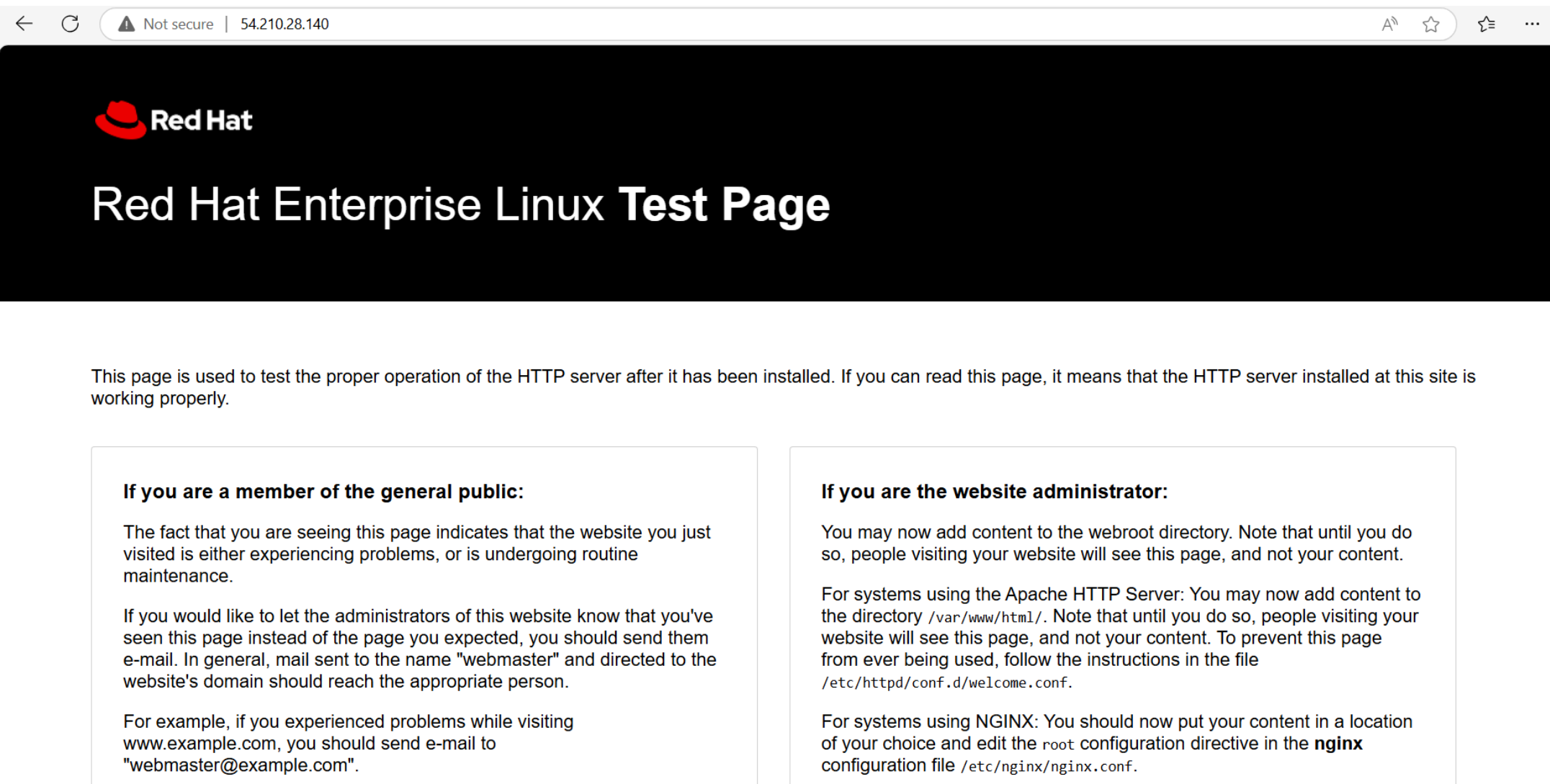
**You can see output as give below after execution**

```
54.162.193.1 | 172.31.36.113 | t2.micro | null
[ ec2-user@ip-172-31-36-113 ~ ]$ ansible-playbook -i inventory.ini -e ansible_user=ec2-user -e ansible_password=DevOps321 ng
PLAY [installing nginx] *****
TASK [Gathering Facts] *****
ok: [172.31.46.239]
TASK [installing nginx] *****
changed: [172.31.46.239]
TASK [start and enable nginx] *****
changed: [172.31.46.239]
PLAY RECAP *****
172.31.46.239 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```


# OUTPUT

**Step1:** Copy your slave node public IP address and paste it on your browser you can get output as shown below

**You can see output as give below**



← ↻ ⚠ Not secure | 54.210.28.140 🔊 ☆ ⌵ ⋮



## Red Hat Enterprise Linux Test Page

This page is used to test the proper operation of the HTTP server after it has been installed. If you can read this page, it means that the HTTP server installed at this site is working properly.

**If you are a member of the general public:**

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting `www.example.com`, you should send e-mail to `"webmaster@example.com"`.

**If you are the website administrator:**

You may now add content to the webroot directory. Note that until you do so, people visiting your website will see this page, and not your content.

For systems using the Apache HTTP Server: You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

For systems using NGINX: You should now put your content in a location of your choice and edit the `root` configuration directive in the **nginx** configuration file `/etc/nginx/nginx.conf`.

