**Lesson 6 Lesson-End Project**

**Website Deployment Using Ansible**

**Project Agenda:** To deploy a static HTML website with Ansible

**Description:** Create a playbook that automates setting up a remote Nginx server to host a static HTML website on Ubuntu 20.04.

**Tools required:** Ansible and jinja2

**Prerequisites:** You need to have Python 2.7 or higher, minimum 8 GB RAM, and SSH or SCP communicator.

**Expected Deliverables:**

Initialize the directory

Apply the configuration

Deploy the website

**Steps to be followed:**

1. Setting up the HTML files
2. Creating a playbook
3. Executing the playbook

**Step 1: Creating a playbook**

1.1 Confirm the availability of Terraform by running the below command:

***terraform -version***

1.2 Create a new directory on your Ansible control node where you’ll set up the Ansible files:

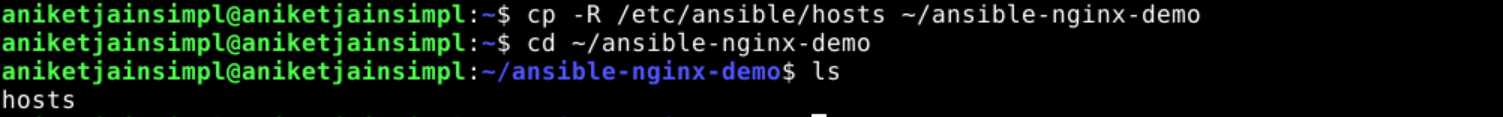
**mkdir ~/ansible-nginx-demo**



1.3 Copy the host or inventory file to this directory using the command below:

**cp -R /etc/ansible/hosts ~/ansible-nginx-dem**

**cd ~/ansible-nginx-demo**



1.4 Click on the URL given below and download the files needed for deploying a website:

<https://github.com/do-community/html_demo_site/archive/refs/heads/main.zip>

Graphical user interface, text, application

Description automatically generated

1.5 Now, go to the downloads and unzip the main file:

***cd Downloads***

***ls***

***unzip html\_demo\_site-main.zip***

Text

Description automatically generated

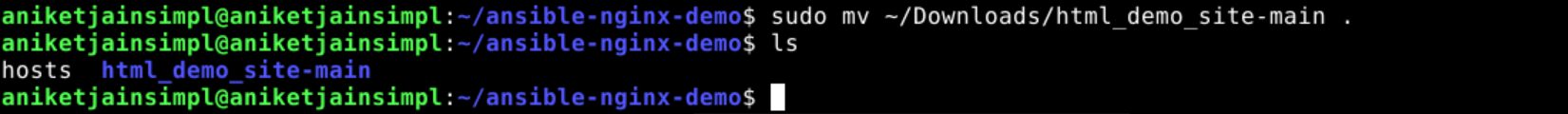
1.6 Now go back to the directory using the below command:

**cd ~/ansible-nginx-demo**

1.7 Move the file from downloads to this directory:

**sudo mv ~/Downloads/html\_demo\_site-main .**

**ls**



1.8 You can check all files that are moved using the below command:

**ls -la html\_demo\_site-main**

Graphical user interface, text

Description automatically generated

**Step 2: Executing the playbook:**

* 1. Create a nginx jinja2 file and add the following code:

**sudo nano nginx.conf.j2**



**server {**

**listen 80;**

**root {{ document\_root }}/{{ app\_root }};**

**index index.html index.htm;**

**server\_name {{ server\_name }};**

**location / {**

**default\_type "text/html";**

**try\_files $uri.html $uri $uri/ =404;**

**}**

**}**

Text

Description automatically generated

* 1. Create a yml file and add the following code in it:

**sudo nano playbook.yml**



**---**

**- hosts: all**

**become: yes**

**vars:**

**server\_name: "{{ ansible\_default\_ipv4.address }}"**

**document\_root: /var/www**

**app\_root: html\_demo\_site-main**

**tasks:**

**- name: Update apt cache and install Nginx**

**apt:**

**name: nginx**

**state: latest**

**update\_cache: yes**

**- name: Copy website files to the server's document root**

**copy:**

**src: "{{ app\_root }}"**

**dest: "{{ document\_root }}"**

**mode: preserve**

**- name: Apply Nginx template**

**template:**

**src: files/nginx.conf.j2**

**dest: /etc/nginx/sites-available/default**

**notify: Restart Nginx**

**- name: Enable new site**

**file:**

**src: /etc/nginx/sites-available/default**

**dest: /etc/nginx/sites-enabled/default**

**state: link**

**notify: Restart Nginx**

**- name: Allow all access to tcp port 80**

**ufw:**

**rule: allow**

**port: '80'**

**proto: tcp**

**handlers:**

**- name: Restart Nginx**

**service:**

**name: nginx**

**state: restarted**

Text

Description automatically generated

**Step 3: Setting up the HTML files**

* 1. Execute the playbook using the below command:

**ansible-playbook playbook.yml**

Text

Description automatically generatedGraphical user interface

Description automatically generated

Once the playbook is finished, if you go to your browser and access your server’s hostname or IP address you should now see the following page:

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated with medium confidence

You have successfully automated the deployment of a static HTML website to a remote Nginx server, using Ansible.