0% COMPLETE
Challenges - Apache2

Project #4 - Automating Linux Administrative Tasks With Ansible
Intro to Ansible (3:24)
Prerequisites (2:19)
Ansible Inventory File (10:12)
Ansible Ad-Hoc Commands: The Shell Module (5:55)
Ansible Ad-Hoc Commands: The Script Module (4:18)
Ansible Ad-Hoc Commands: The

Challenges - Apache2

The Apache2 Web Server

How to solve these challenges:

- To be consistent with the filenames and paths run the commands on Ubuntu. Install it locally in a VM or at a cloud provider. Use my referral link below to create your account on DigitalOcean and you'll get $100 in credit over 60 days: https://m.do.co/c/f1d4debf213f
- Write your solution **in a terminal or in a shell script** and **test it.**
- If your solution is not correct, then try to understand the error messages, watch the video again, rewrite the solution, and test it again. Repeat this step until you get the correct solution.
- **Save** the solution in a file for future reference or recap.

Challenge #1

- Install Apache, check that it's running, and connect to the Web server using the Browser and server's IP address;

Are you stuck? Do you want to see the solution to this challenge? Click here.

Challenge #2

- Configure Virtual Hosting for a domain and access the website using the domain name;
- In the DocumentRoot directory of the Virtual Host, create a directory called **mysite** and a file named **myfile.html**. Access the file with the Browser;

Are you stuck? Do you want to see the solution to this challenge? Click here.

Challenge #3

- Request a free certificate from Let's Encrypt and configure HTTPS for the Virtual Host;
- Test that HTTP is redirected to HTTPS;

Are you stuck? Do you want to see the solution to this challenge? Click here.

Challenge #4

- Create a directory called **admin** in the Virtual Host's DocumentRoot directory and limit the access to it by IP address using **<Directory> ... </Directory>**;
- Allow only your public/private IP address to access the directory.

Are you stuck? Do you want to see the solution to this challenge? Click here.

Challenge #5

- Consider the previous challenge. Implement the same requirements using **.htaccess** instead of **<Directory> ... </Directory>;**

Are you stuck? Do you want to see the solution to this challenge? Click here.

Challenge #6

- Consider the previous challenge. Disable the use of .htaccess in the **admin** directory;
- Test that the directives in **.htaccess** have no effect on the **admin** directory;

Are you stuck? Do you want to see the solution to this challenge? Click here.

Challenge #7

- Create a directory called **projects** in the Virtual Host's DocumentRoot directory and limit file access by IP address using **<Files> ... </Files>**;
- PDF and XLS files can be accessed only from your public/private IP address. All other files can be accessed from all IP addresses;

Are you stuck? Do you want to see the solution to this challenge? Click here.

Challenge #8

- Consider the previous challenge. Implement the same requirements using **.htaccess** instead of **<Directory> ... </Directory>;**

Challenge #9

- Create a directory called **team** in the Virtual Host's DocumentRoot directory and enable HTTP Digest Authentication for accessing the directory. Use **<Directory> ... </Directory>**;

Are you stuck? Do you want to see the solution to this challenge? Click here.

Challenge #10

- Consider the previous challenge. Add IP based authorization to the **team** directory;
- Access to the **team** directory is allowed if both the user authenticated successfully and came from a permitted source IP address (user your public/private IP address);

Are you stuck? Do you want to see the solution to this challenge? Click here.

Challenge #11

- Create a directory called **website** in the Virtual Host's DocumentRoot directory and configure that **home.html** is delivered by default to the clients when accessing the directory (and only that directory and its subdirectories);
- If the **home.html** does not exist, the server will not list the directory contents;

Are you stuck? Do you want to see the solution to this challenge? Click here.

Challenge #12

- Activate and test HTTP compression using mod_deflate;

Are you stuck? Do you want to see the solution to this challenge? Click here.

Challenge #13

- Activate the **status** module and monitor the server using the module;
- Permit only your public/private IP address to access the monitoring page;

Are you stuck? Do you want to see the solution to this challenge? Click here.