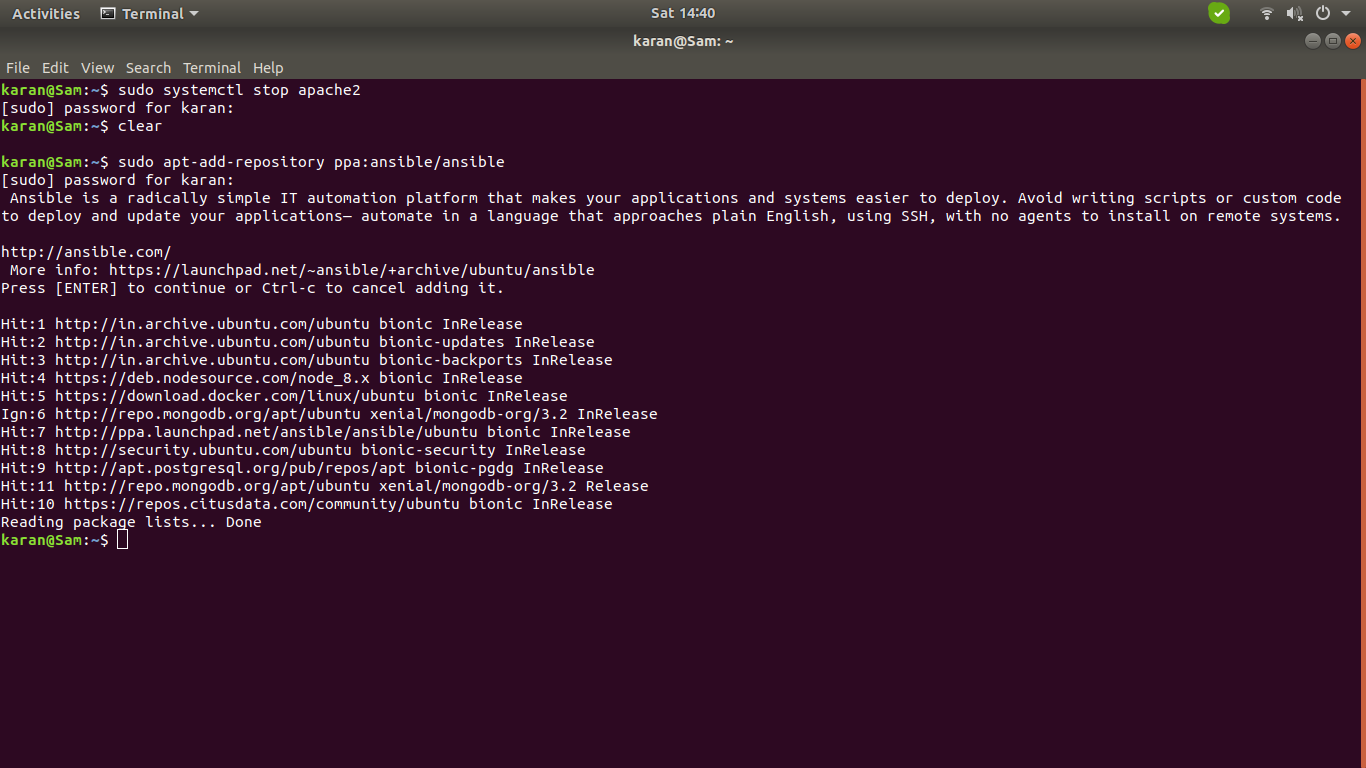
The setup of Citus, Mongo, Nginx, NodeJS, Redis, Supervisor was done running the respective scripts on Ansible. It has been tested on Ubuntu 18.04.

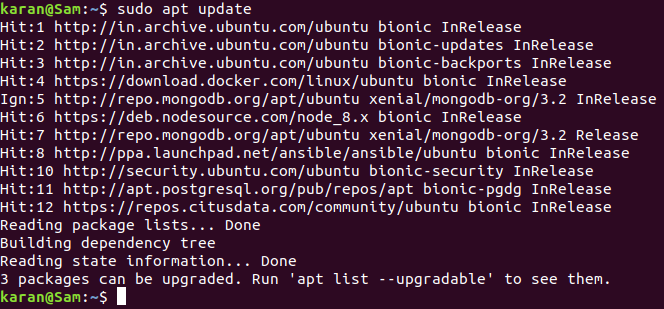
Following are the steps that I took:

1. Install ansible: In a terminal window, run the following commands:

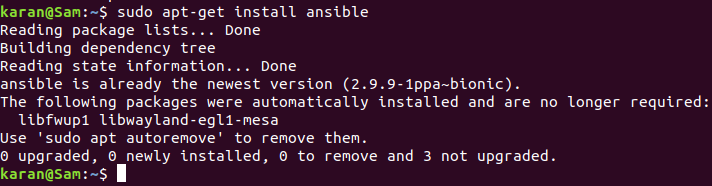
a. sudo apt-add-repository ppa:ansible/ansible



b. sudo apt update



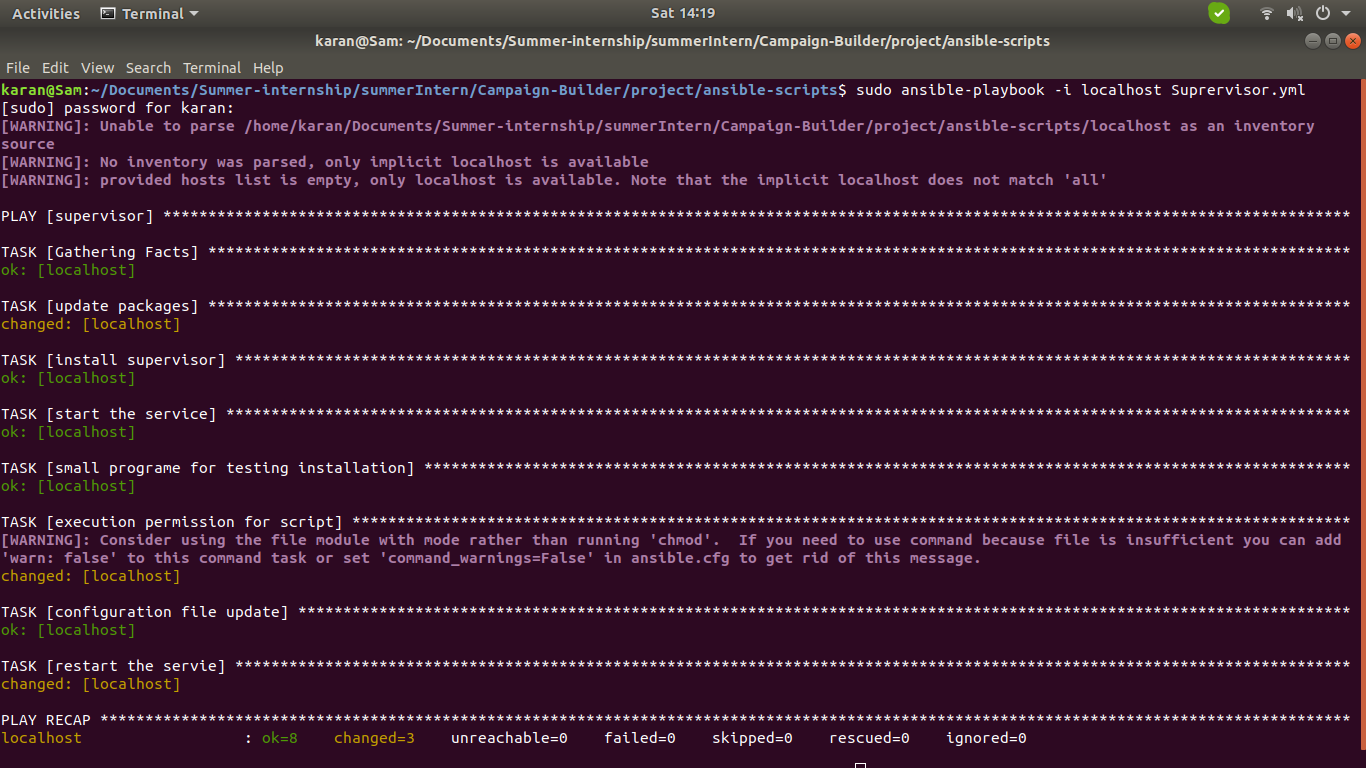
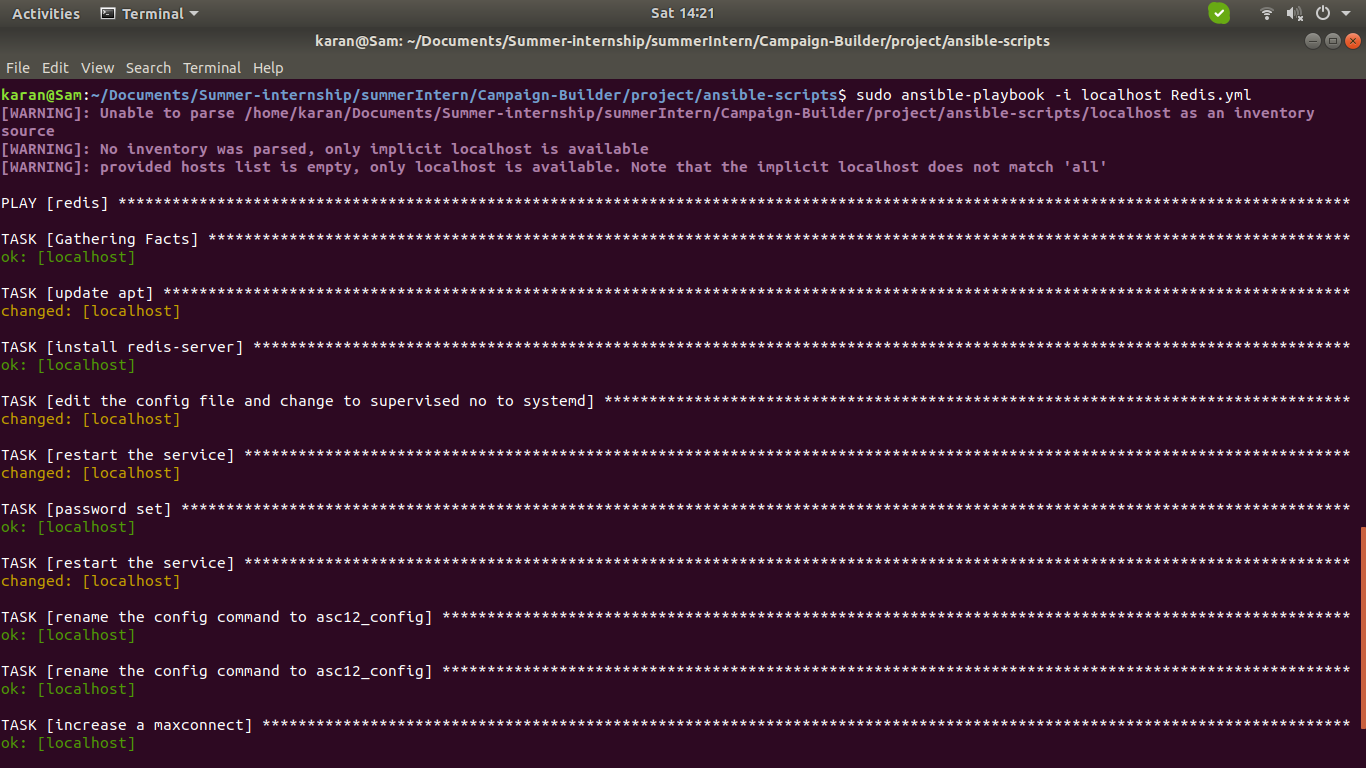
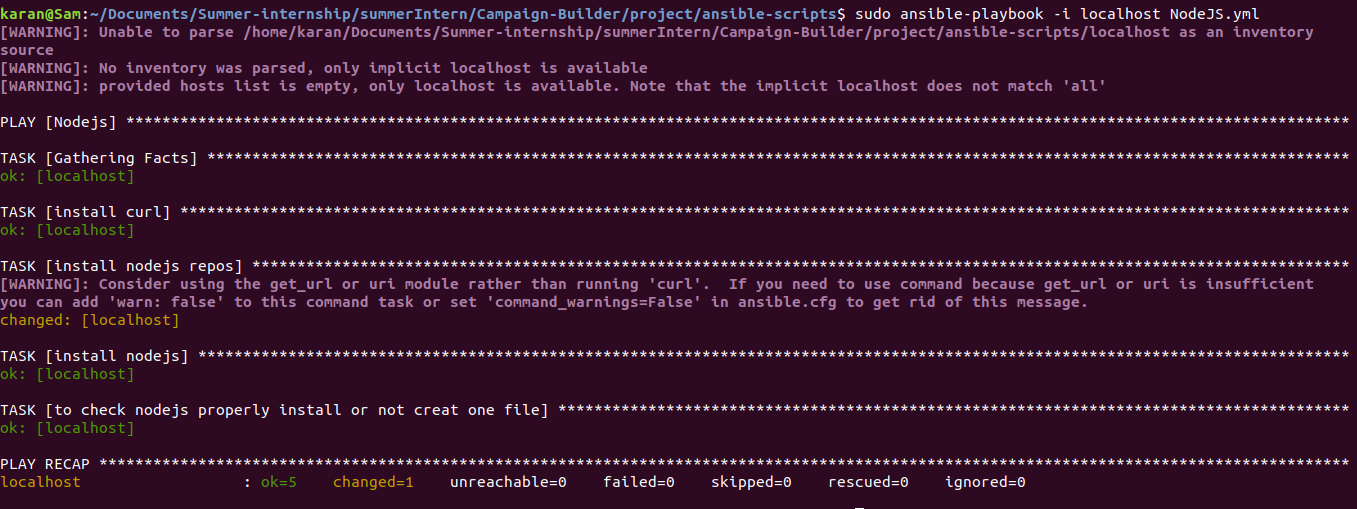
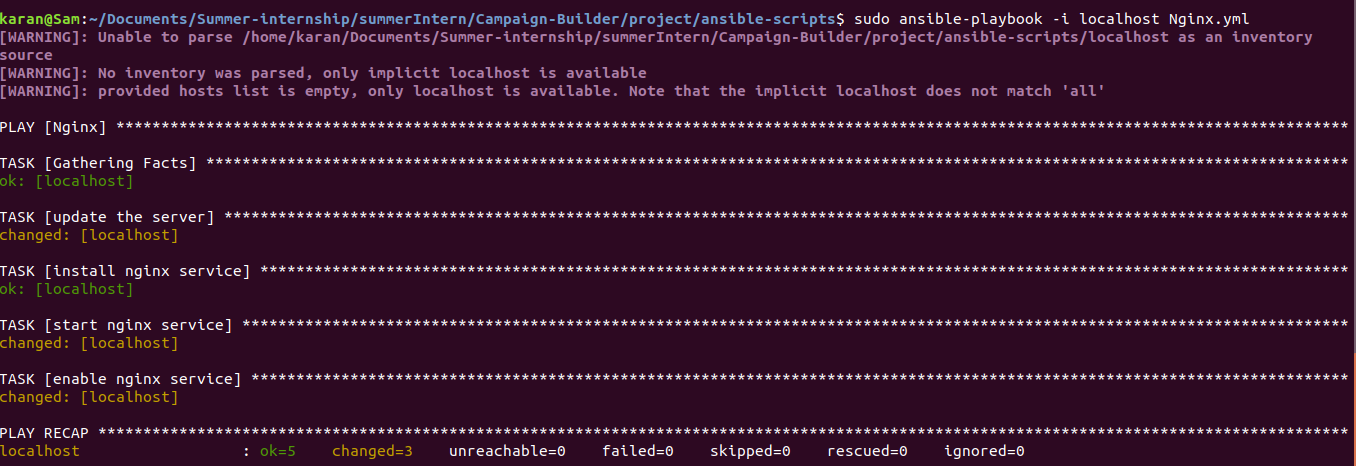
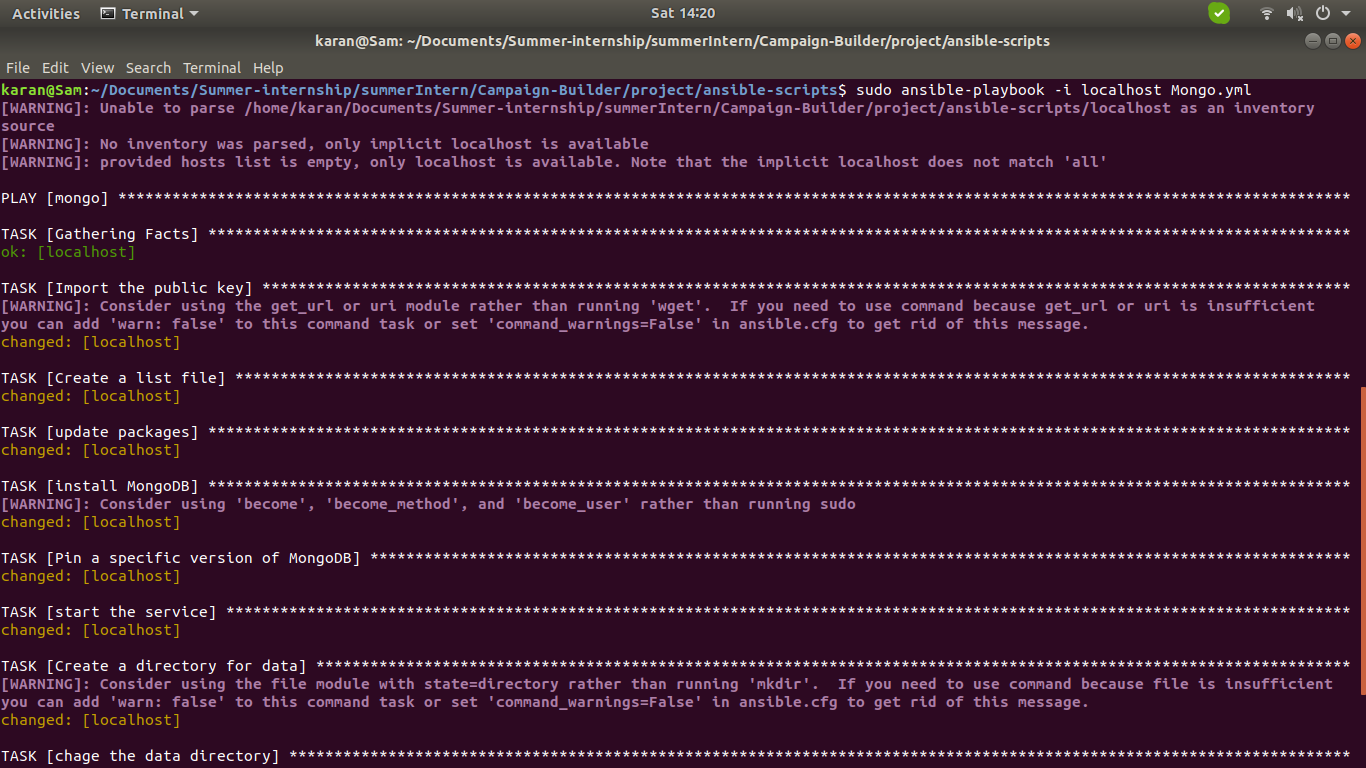
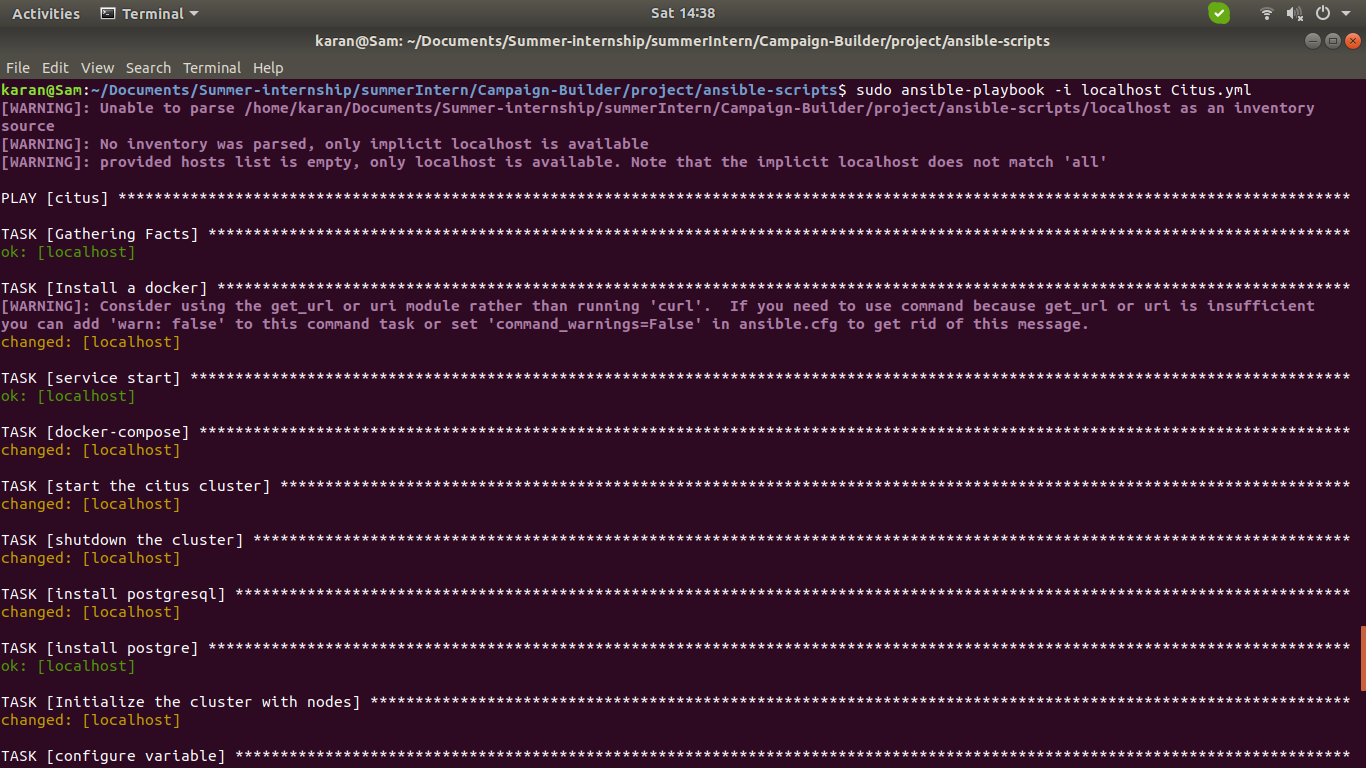
c. sudo apt install ansible



2. Open the directory in which the yml script files are there

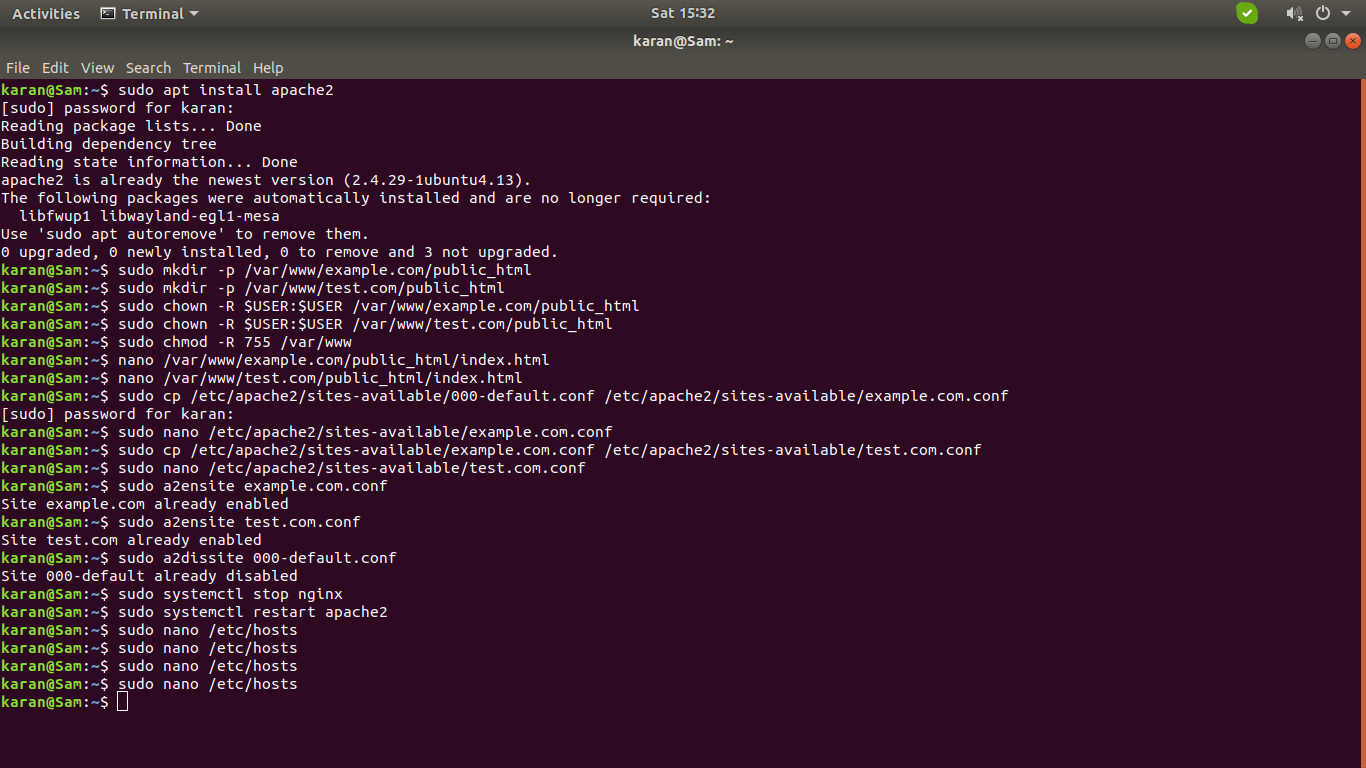
3. For each one of the script, run the following command:

sudo ansible-playbook -i localhost <yml-script-file-name>



4. The setup of all the required scripts will happen.

The setup of virtual hosts was done using apache2 web server. Following are the steps that I took:



1. Install apache2: In a terminal window, run the following command

sudo apt install apache2

2. Create the directory structure: Open a terminal window inside some folder and run the following commands:

a. sudo mkdir -p /var/www/example.com/public\_html

b. sudo mkdir -p /var/www/test.com/public\_html

3. Grant permissions so that user can modify the files created. Run the following commands:

a. sudo chown -R $USER:$USER /var/www/example.com/public\_html

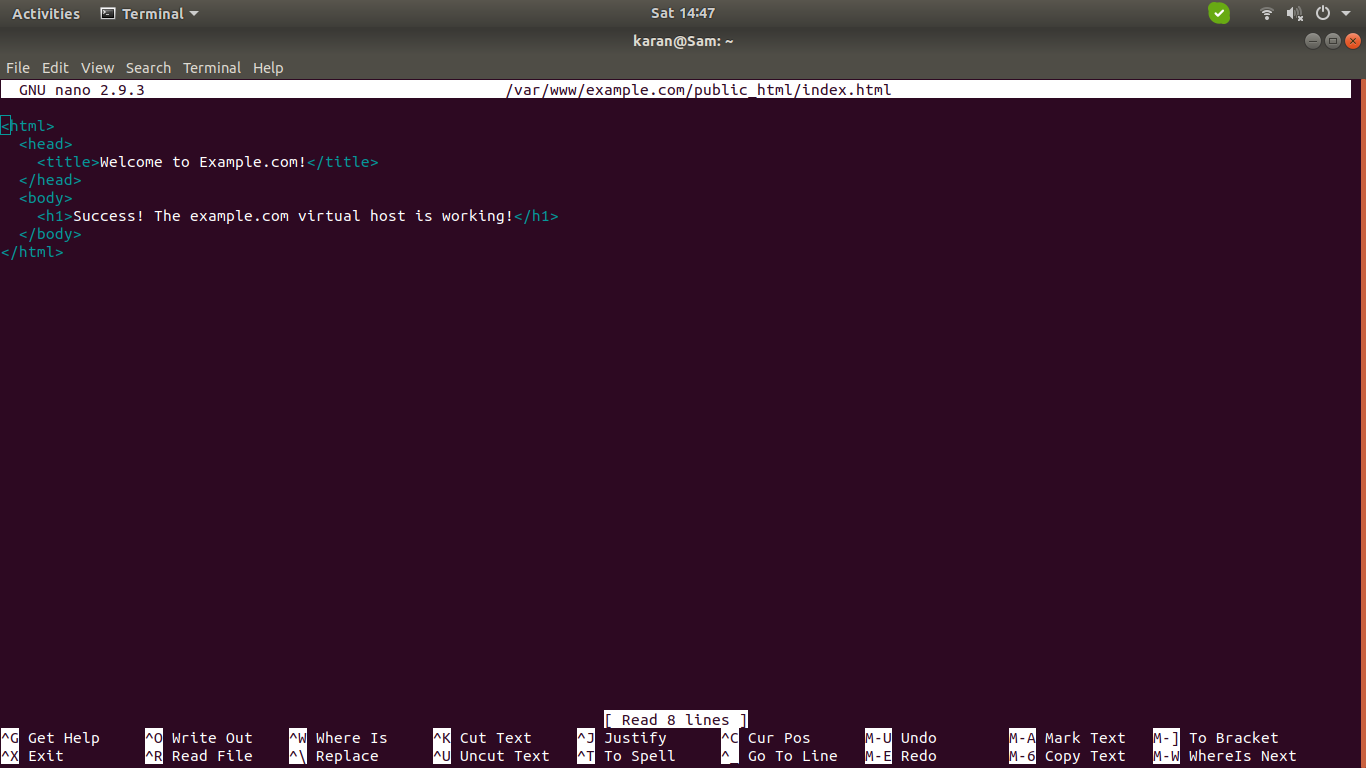
b. sudo chown -R $USER:$USER /var/www/test.com/public\_html

c. sudo chmod -R 755 /var/www

4. Create sample pages for the sites to be hosted: Run the following commands:

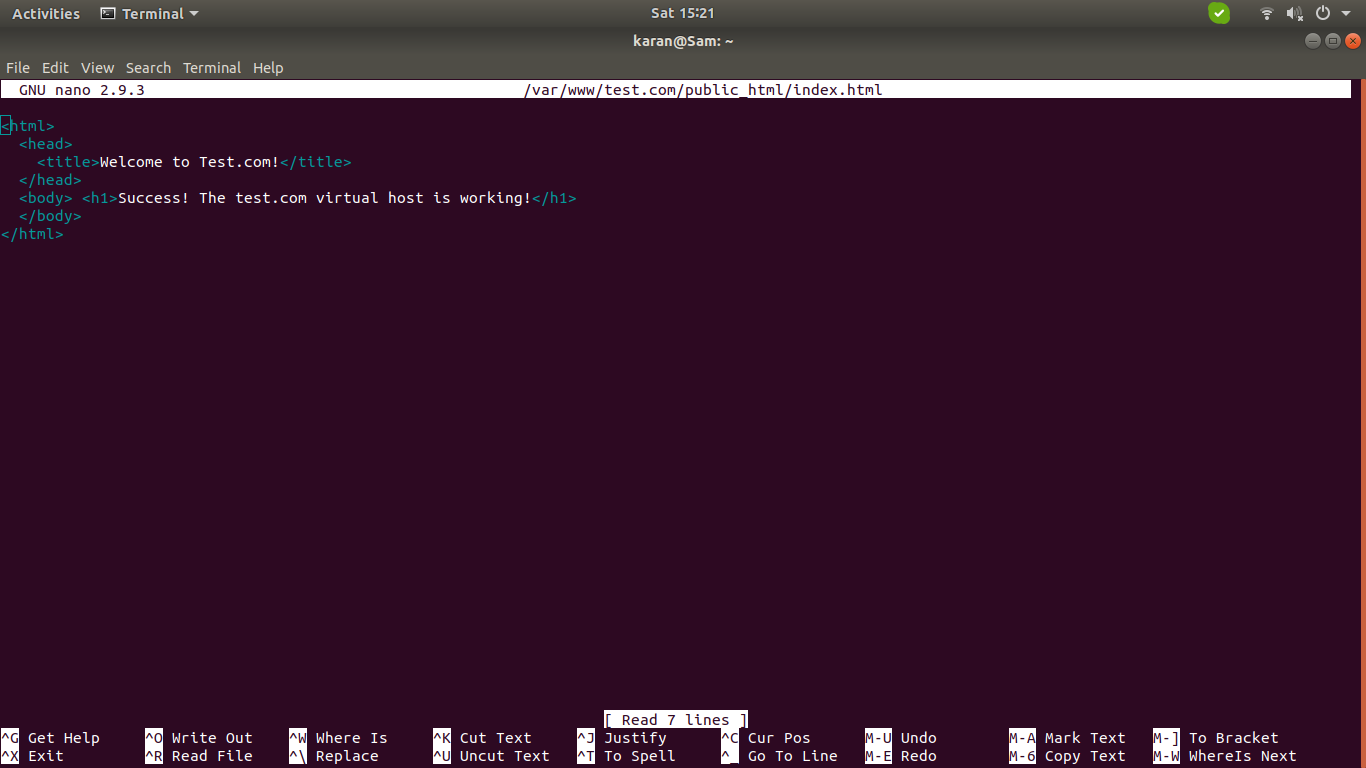
a. nano /var/www/example.com/public\_html/index.html

Write the desired html into the nano editor. Save and close the file



b. nano /var/www/example.com/public\_html/index.html

Write the desired html into the file. Save and close the file



5. Create first virtual file hosts: Run the following commands:

a. sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/example.com.conf

b. sudo nano /etc/apache2/sites-available/example.com.conf

c. Copy and paste following text in the file opened in nano editor:

<VirtualHost \*:80>

ServerAdmin admin@example.com

ServerName example.com

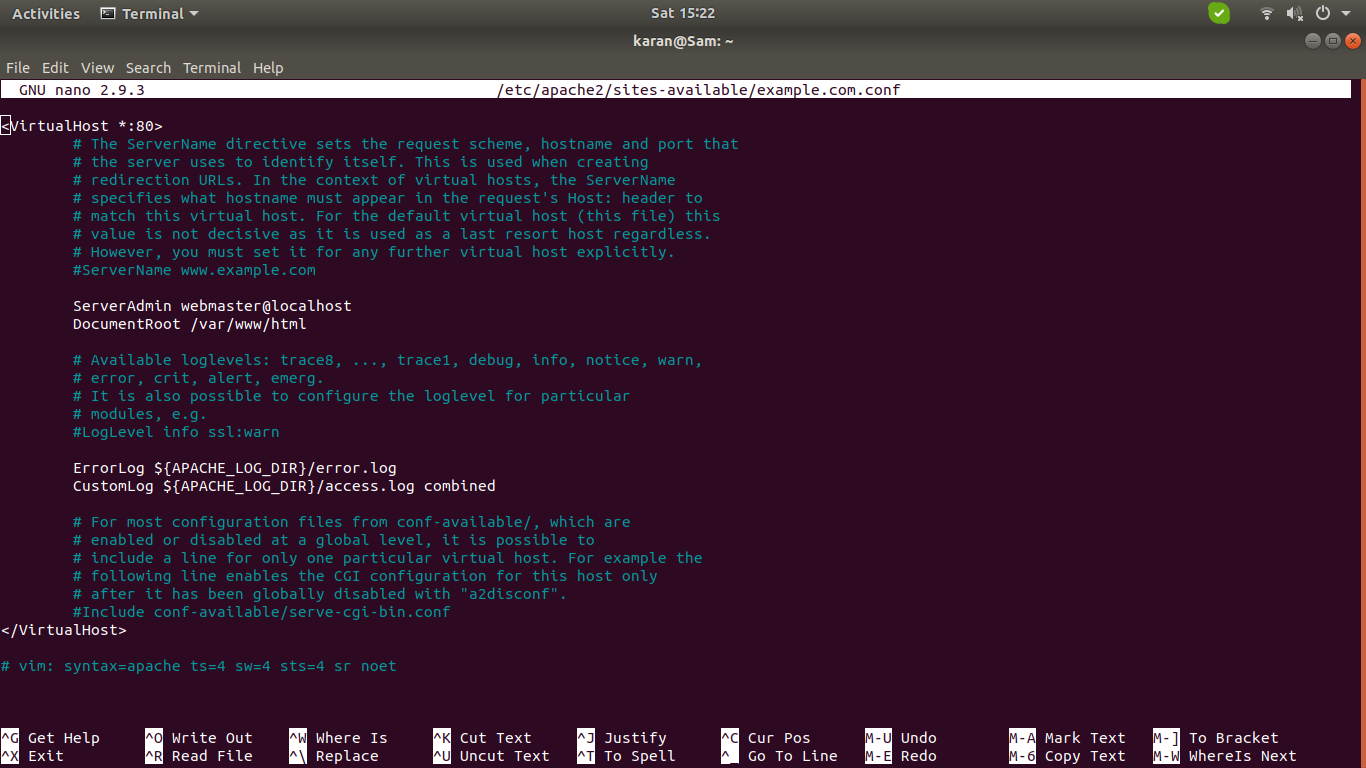
ServerAlias www.example.com

DocumentRoot /var/www/example.com/public\_html

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>



d. Save and close the file

6. Create second virtual file host: Run the following commands:

a. sudo cp /etc/apache2/sites-available/example.com.conf /etc/apache2/sites-available/test.com.conf

b. sudo nano /etc/apache2/sites-available/test.com.conf

c. Copy and paste following text in the file opened in nano editor:

<VirtualHost \*:80>

ServerAdmin admin@test.com

ServerName test.com

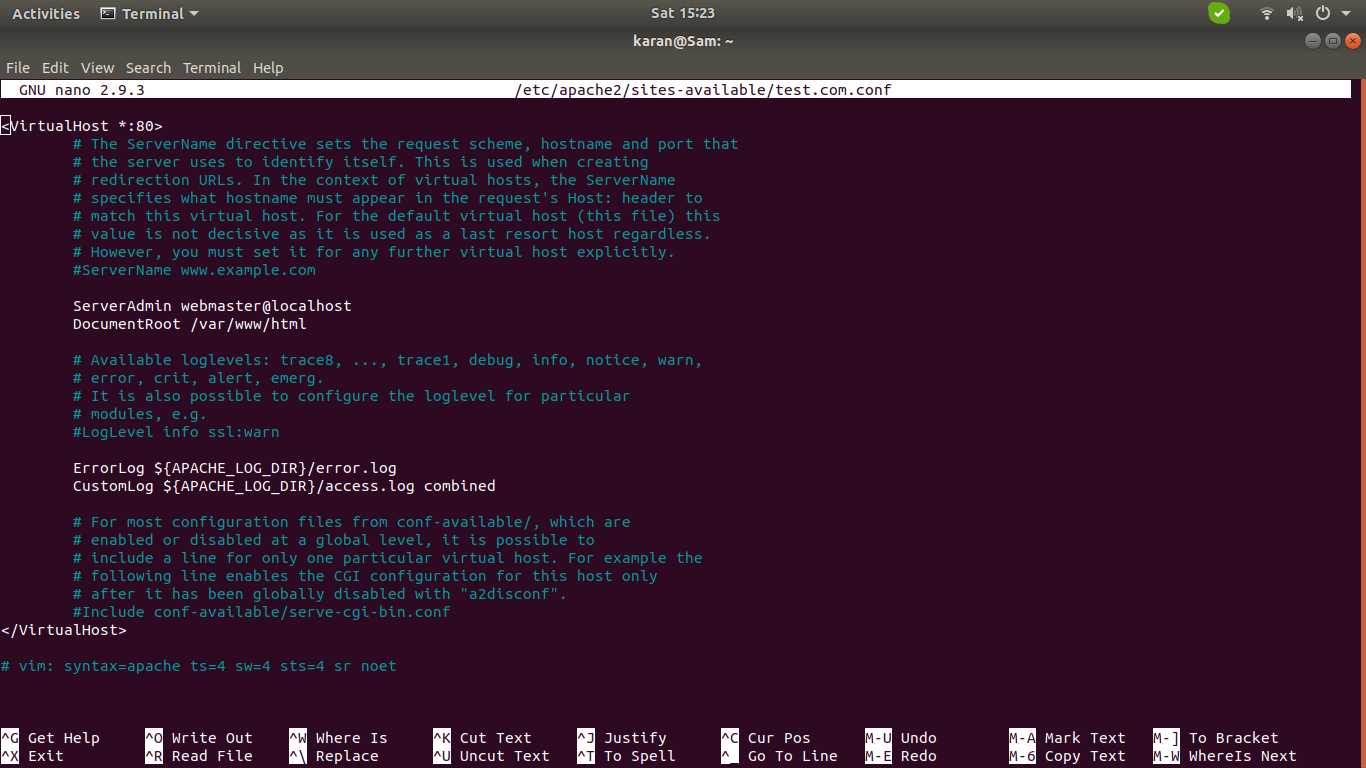
ServerAlias www.test.com

DocumentRoot /var/www/test.com/public\_html

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>



d. Save and close the file

7. Enable new virtual file hosts: Run following commands:

a.sudo a2ensite example.com.conf

b.sudo a2ensite test.com.conf

c.sudo a2dissite 000-default.conf

d.sudo systemctl restart apache2

8. If you get an error : AH00072: makesock: could not bind to address [::]:80 (98)Address already in use: AH00072: makesock: could not bind to address [::]:80

You have to stop the Nginx service which may be running in the background. Run the following command:

sudo systemctl stop nginx

After that, rerun the command:

sudo systemctl restart apache2

9. Set up local hosts file: Run following commands

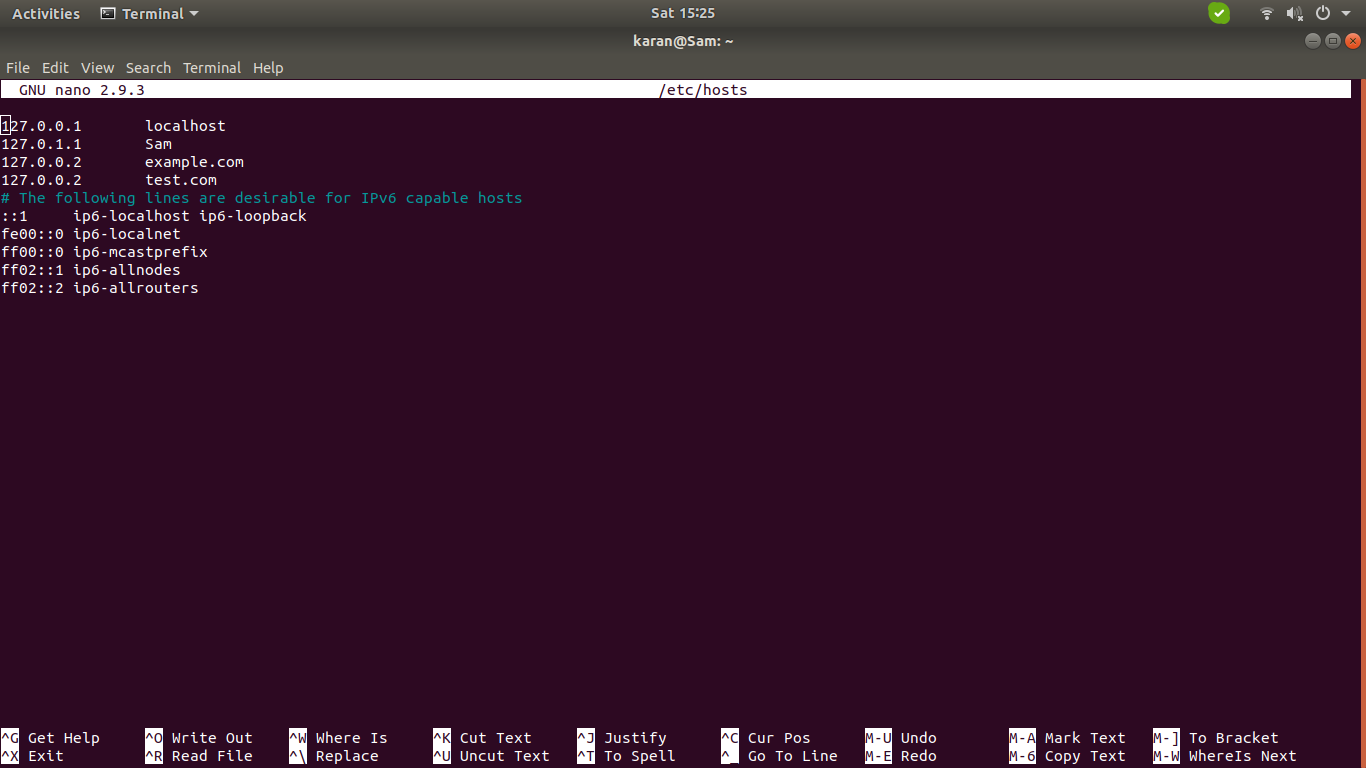
a. sudo nano /etc/hosts

b. Inside the file that opens up, add the following lines:

<your\_server\_IP> example.com

<your\_server\_IP> test.com

(I gave my server IP as 127.0.0.2)



c. Save and close the file

10. Test your results:

In a browser window, go to the domains example.com and test.com to see your result. I closed my internet connection to ensure that the browser looks for pages on local machine