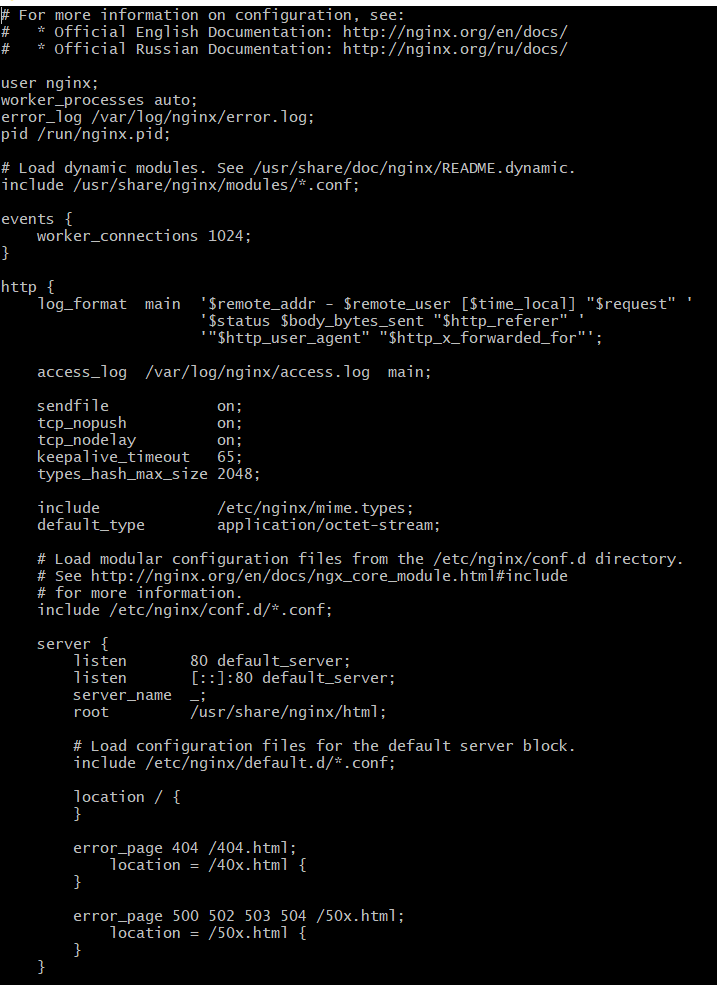
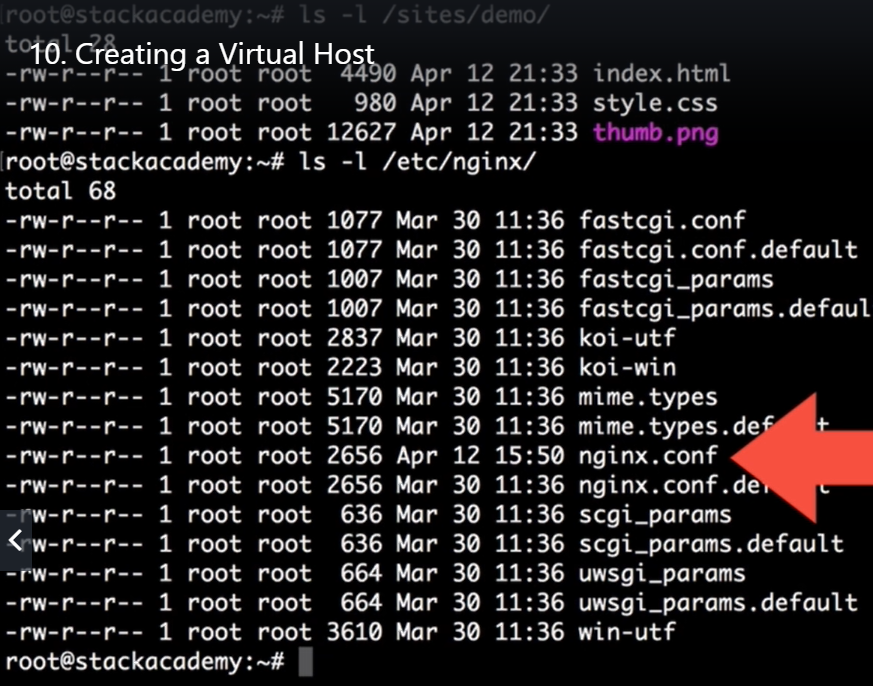
**Nginx configurations:**



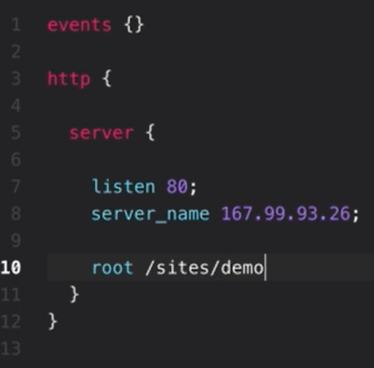
* Below is the default nginx.conf file.



* Let’s say, we copy the code to **/sites/demo** directory and then we need to edit the mail config file which is nginx.conf file.



* Port 80 is for HTTP and 443 is for HTTPS
* By default, nginx assumes as port 80. But we can give it listen port in the config file as below.
* We need to give the server IP and the root path of the code as below.

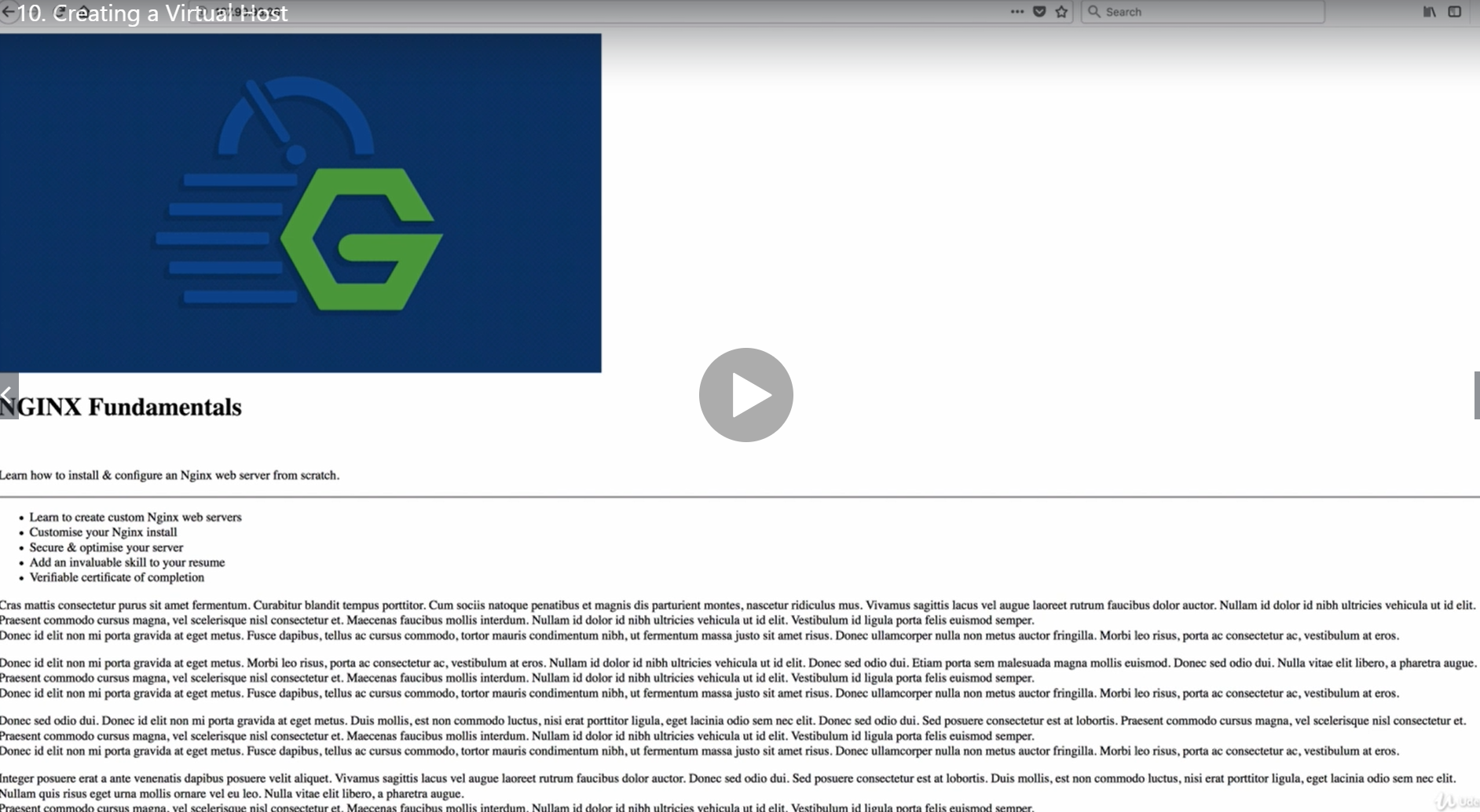


* So here, **/sites/demo** will be the root directory of nginx. If a request comes for **/images/image.png**, by default nginx will look for that in root directory as **/sites/demo/images/image.png**
* Once the configuration is done, we can reload it with reload command without any downtime.

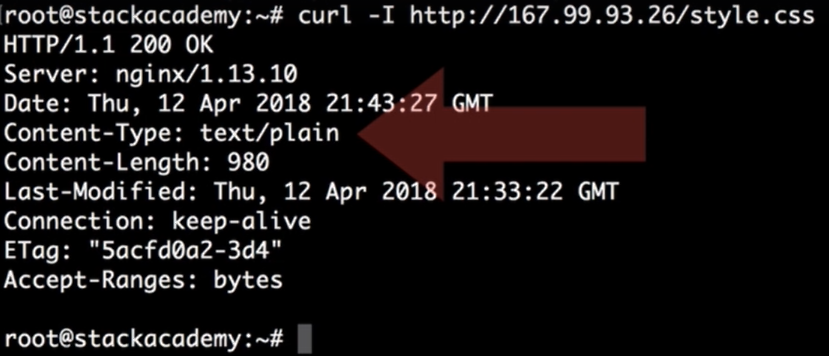
We can verify the new configurations with -t option.

* **Nginx -t 🡪 to check the new configurations**
* **Systemctl reload nginx 🡪 to reload the configuration.**

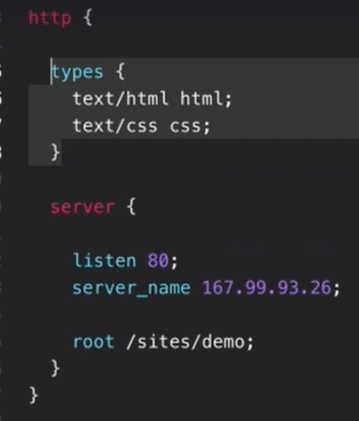
Now, we can get the nginx webpage with the content in code deployed.



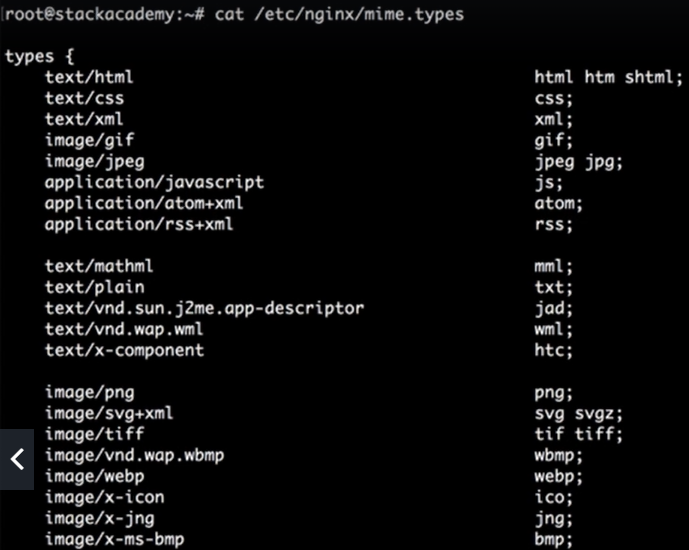
* But this is not the expected page here.
* It is may be because the nginx is sending the wrong mime type here and the style types are not configured.
* We can confirm this css file response with curl as below.



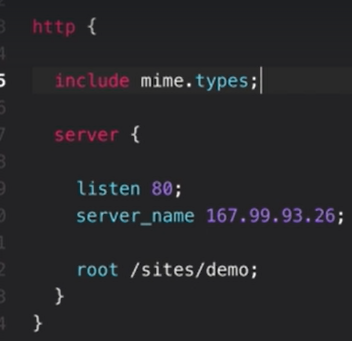
* It is reading the content type as plain text. To fix the plain text, we can add the nginx to read these files in config file as below.



* But nginx has already a file called mime.types in /etc/nginx directory defining all these types as below.

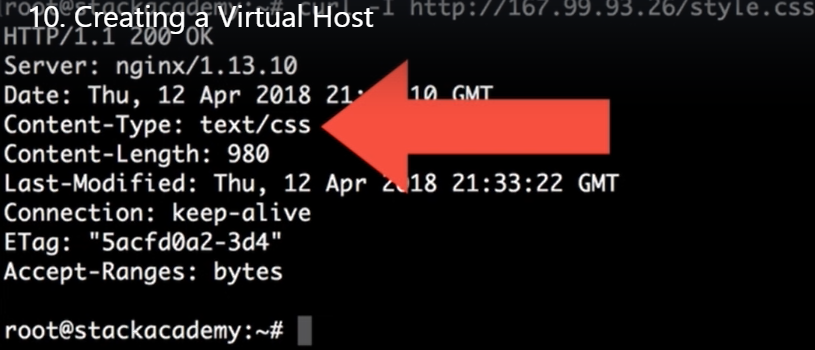


* it’s including various types of files extensions as above.
* So instead of defining all these, we can simply include these content types in our config file as below.



This setting comes by default

* Save this, reload the configuration. Then we can get the proper response in curl command and as well as the proper image in browser as well.



* Now everything works fine with browser as well.

