**GUIDE FOR CHATSCRUM DEPLOYMENT ON LINUX SERVER**

**Prerequisites:**

1. Create a gitlab account if you don’t have one already (https://gitlab.com)

2. In order to complete this guide, you should have a fresh CentOS 8 server instance with a non-root user with sudo privileges configured (Instance type: t2.medium atleast)

3. Allow *all traffic* in your server’s security group

4. Install *httpd* and *git* on your server using *yum*

5. You should have MySQL installed and configured on the server. Ensure to start the service and enable it to start automatically on server reboot. (The Chatscrum application uses MySQL database).<https://www.hostinger.com/tutorials/how-to-install-mysql-on-centos-7>

**NOTE:**

Linuxjobber’s chatscrum application has two major parts. There is the Angular part that handles the frontend view of the application. Then there is the Django path that handles routing and communication with the MySQL database.

**Deploying the Django part to Linux server**

1. **Log into mysql and create a database**

* *create database chatscrum;*

1. **Prepare your project folder for deployment:**

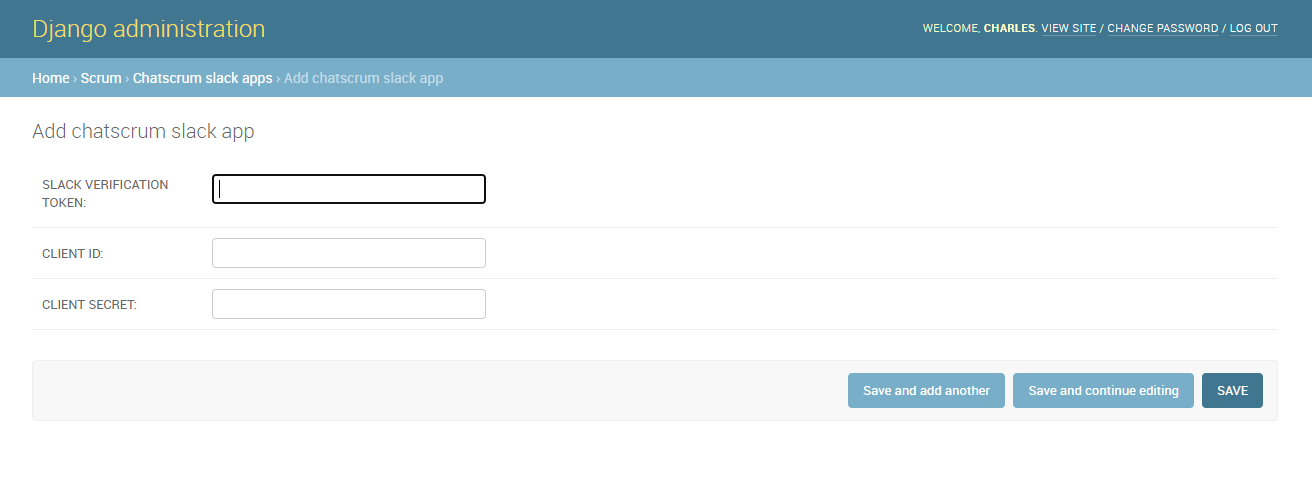
* Use git to get the chatscrum project source code into your home directory (*/home/youruser; “youruser” reps the logged in user*)
* Edit */home/youruser/chatscrum/Django/ScrumMaster/ScrumMaster/settings.py*
* Set DEBUG to False
* Add your elastic IP address to allowed hosts. If you already have the asterisk, skip this step.
* Set database configurations to use the credentials of your mysql server
* Set the STATIC\_ROOT (STATIC\_ROOT = os.path.join(BASE\_DIR, ‘static’)). This references the folder where static files will be collected at when *manage.py collectstatic* is run

1. **Prepare your server’s environment:**

* Install python: <https://tecadmin.net/install-python-3-7-on-centos/> (Use these options during configuration => **./configure --prefix=/usr/local --enable-shared --enable-optimizations**)
* Install mod\_wsgi: An interface that will run python applications on apache: (<https://www.marek.tokyo/2018/08/apache-24-modwsgi-python-37-django.html>) If you encounter any error during ‘make altinstall’, run the following command before ‘make altinstall’: *sudo ldconfig /usr/local/lib*
* Install *virtualenv;* A cli utility for creating virtual environments (*pip3.7 install virtualenv*)
* Create a virtual environment named *venv* within the project. ( *virtualenv /home/youruser/chatscrum/Django/ScrumMaster/venv*)
* Activate the virtual environment (*source /home/youruser/chatscrum/Django/ScrumMaster/venv/bin/activate*)
* Edit */home/youruser/chatscrum/Django/ScrumMaster/requirements2.txt*: add *boto3 and cryptography==3.3.2* to the end of the file as these are dependencies that were omitted. Comment out *mysqlclient* package as it won’t be needed; *PyMysql* was used instead. Replace *slackclient* version with the latest version. Replace *zope.interface* with the latest version. (You can simply remove the specified version number to have the latest version of the package installed *‘==xxx’*)
* Install all required packages by running *sudo* *pip3.7 install -r requirements2.txt.* This will install all required packages within the virtual environment created.
* Navigate into */home/youruser/chatscrum/Django/ScrumMaster/* where ‘manage.py’ is, and run *python3.7* *manage.py makemigrations* to read all django models
* Run *python3.7* *manage.py migrate* to set up tables in your database
* Run *python3.7* *manage.py collectstatic* to collate static files
* Run *python3.7 manage.py createsuperuser --username <your-name>* and follow the prompt to create a superuser account.

1. **Start up the django server and create a record in the admin:**

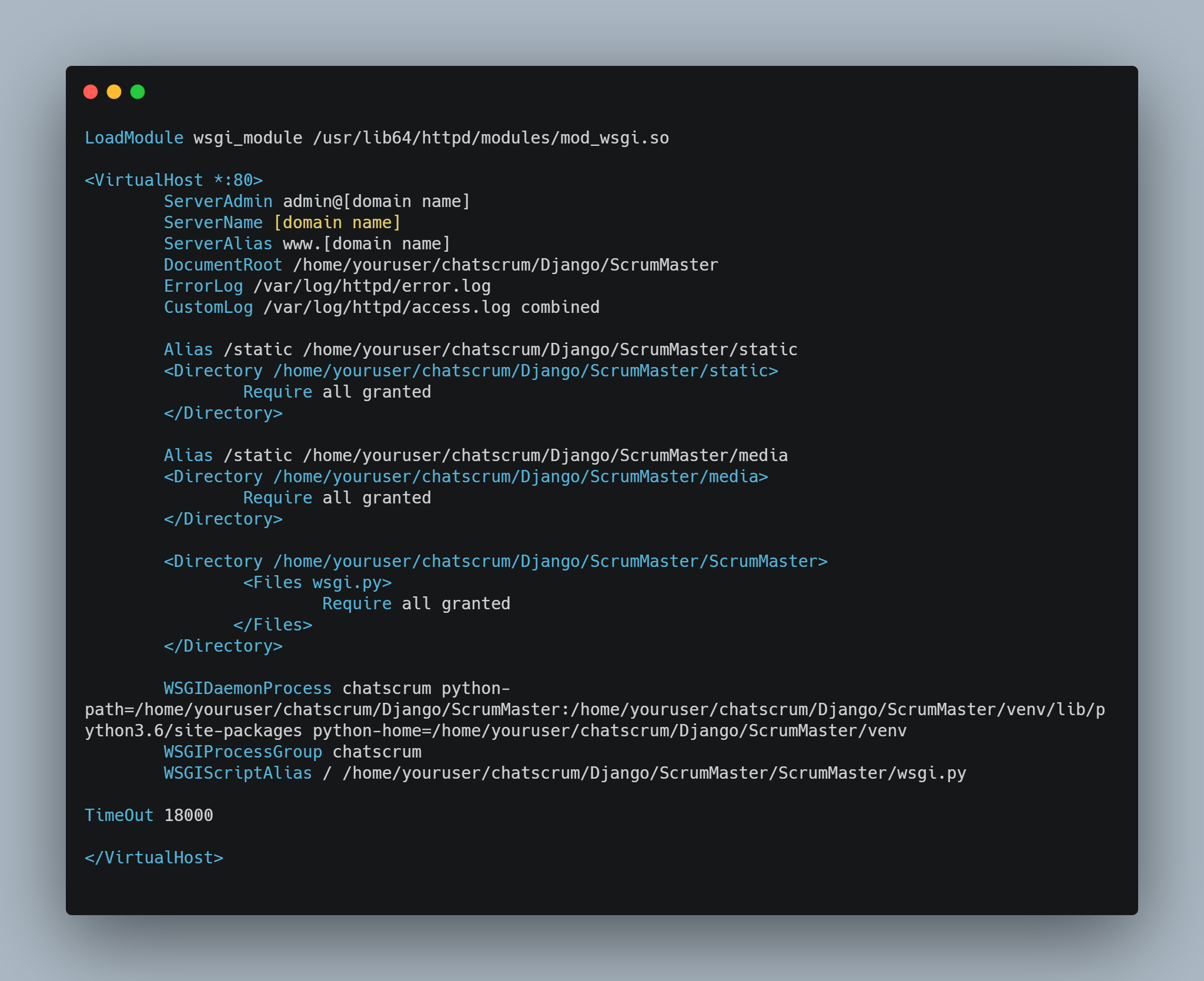
* Still in */home/youruser/chatscrum/Django/ScrumMaster/*, run *python3 manage.py runserver 0.0.0.0:8000* to start up the django server
* Navigate to *[address]:8000/admin* via your browser to access the django admin interface. Replace *[address]* with the IP address of your machine
* You will be needing a **Chatscrum Slack Apps** record in place before authentication can be carried out successfully. For now, log in and create a new record in **Chatscrum Slack Apps.** Fill in random contents in the three fields and save it. It will suffice.



### 

1. **Interfacing with Apache (httpd)**

* *The last thing we want to do* is to interface the application with an apache web server using mod\_wsgi.
* Create *chatscrum.conf* file within */etc/httpd/conf.d/* ; this file will contain our apache virtual host configuration for chatscrum app.
* Write the following content into the file:



Ensure you replace [domain name] with the domain name pointing to your elastic IP. Also replace ‘youruser’ with the logged in username. Also specify your correct python version installation in the ‘python-path’=> **python3.7**

* Run *apachectl configtest* to confirm there are no errors in your configuration
* *wsgi.py* is the entry point of our application using python’s module syntax. Inside this file, a function named *application* is defined, and is used to communicate with the application. We will write some codes that will activate the virtual environment where we installed our required packages for the chatscrum app.
* Edit */home/youruser/chatscrum/Django/ScrumMaster/ScrumMaster/wsgi.py* and add the following contents just before *application = get\_wsgi\_application():*

: 

* Set SElinux security to Permissive
* Grant access to all files within your home directory to apache so it can read the chatscrum folder. Best way to do this will be to grant *rx* access to the *‘youruser’* group bits of your home directory(*/home/youruser*), then add *apache* to this group.
* Run *sudo chmod -R 766 /var/log* to open access to apache to write logs

1. **Rounding off**

* Activate the virtual environment and start up the server just like before. Fix errors if there is.
* Stop the server and start *httpd* service using *systemctl* utility. Now, apache will be able to access and server the django application through the WSGI interface.
* Navigate to http://*[IP/hostname]/admin* via your browser to access the Django admin interface. (Replace IP or hostname with the correct value)

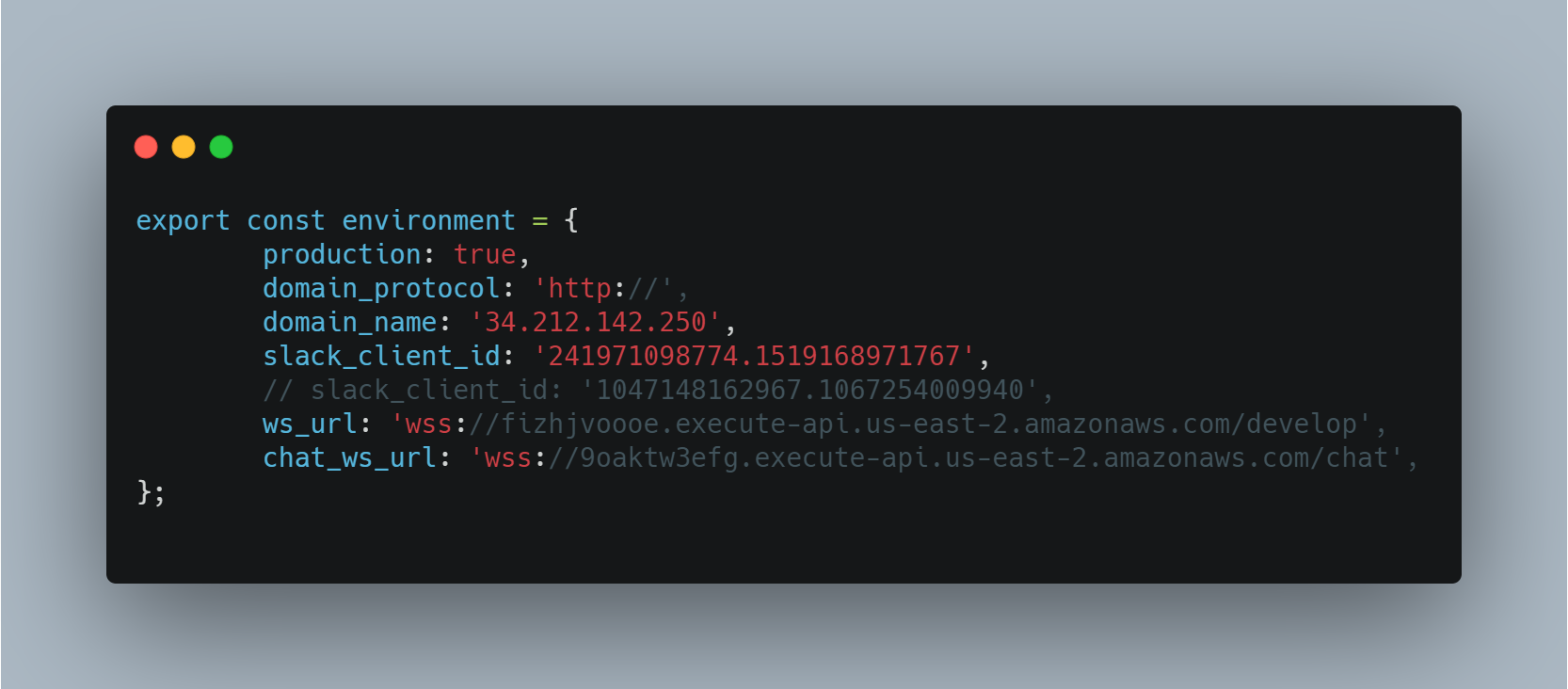
**Deploying the Angular Part (Linux server)**

**1. Prep your server environment:**

* Install and configure *node and npm* on your server: <https://linuxize.com/post/how-to-install-node-js-on-centos-7/>
* Using *npm*, install *angular cli* globally on your server: (*sudo npm install -g @angular/cli@9*)

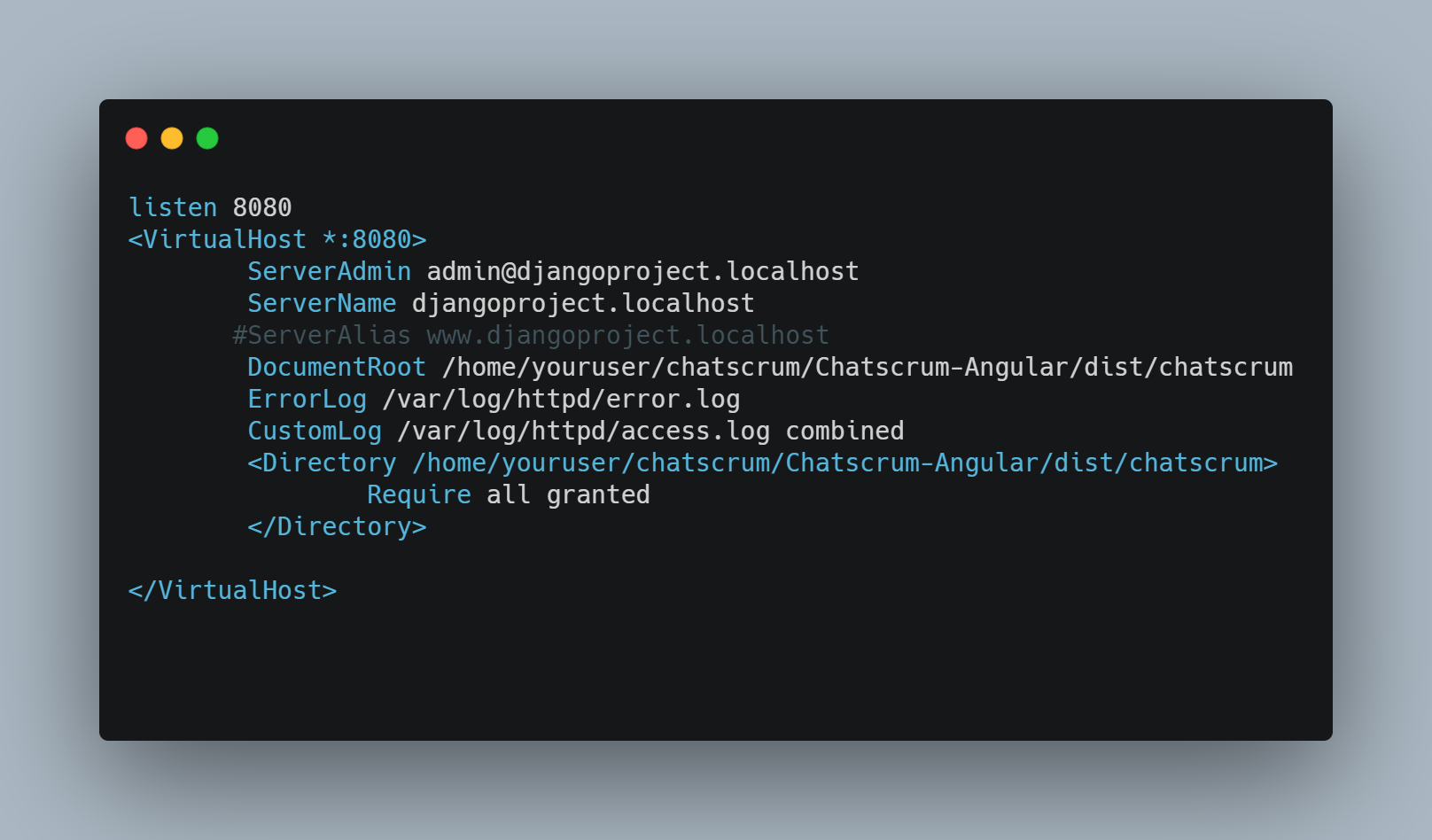
**2.** **Prep your application for production deployment:**

* Edit */home/youruser/chatscrum/Chatscrum-Angular/src/environments/environment.prod.ts* and set your environment variables in *environments.prod.ts* as this is the file that production environment will take cognizance of. Value of the domain\_name should be the actual elastic IP or hostname of your server. The domain\_protocol + domain\_name setup just points to our django app for communication with the Angular part: (you can see the content within environment.ts file which is in same directory as environments.prod.ts. The former is used in a development environment)



* Within the angular workspace directory (Chatscrum-Angular) where package.json exists, run *npm install* from the terminal to install all required packages.
* Run *ng build --prod*  from within the workspace directory. This will compile the Angular app into an output directory named *dist/.* *dist/* will contain *chatscrum/* folder ready for production deployment.

**3.** **Edit the chatscrum.conf configuration file to add a new virtualhost (/etc/httpd/conf.d/chatscrum.conf):**



You can use any free port you desire. Replace ‘youruser’ with your username. In essence, The angular app will be served on PORT *8080*, while the Django app will be served on PORT *80*.

**4.** **Edit httpd default conf file(/etc/httpd/conf/httpd.conf) and define fallback resource for the Angular app:**

****

This will help apache point back to index.html whenever a url endpoint without a file is navigated to.

**5.** **Visit http://*[IP ADDRESS]:8080* via your browser:**

* Replace [IP ADDRESS] with the elastic IP or domain name of your server