**GUIDE FOR CHATSCRUM DEPLOYMENT ON WINDOWS SERVER 2016**

**Prerequisites:**

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

2. You should have an accessible EC2 machine running Microsoft Windows server 2016

3. You should have Internet Information Service (IIS) installed and configured on the server. Ensure that CGI application development role service is installed along. If you have carried out IIS installation before now without ticking CGI service, perform a feature-based installation and have it installed.

4. You should have *URL Rewrite (An IIS extension)* installed on your server.<https://www.iis.net/downloads/microsoft/url-rewrite>

5. Download and install Git for windows on your server

6. Download and install a code editor you can use on your server. E.g.Visual Studio Code

7. You should have MySQL installed and configured on the server. (The Chatscrum application uses MySQL database).<https://mid.as/kb/00145/install-configure-mysql-on-windows#download-mysql>

8. You should have *node* installed and configured on your server: <https://phoenixnap.com/kb/install-node-js-npm-on-windows>

9. Use Git to get the chatscrum project source code from the repo onto your server (Request for access to the project repo)

**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. We will be going through the deployment of both.

**Deploying the Django part to IIS**

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

* *create database chatscrum;*

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

* Create a *django/* folder on your C: drive
* Copy the contents within *Django/ScrumMaster/,* excluding *py34env/ directory,* into *C:/django*
* Create *web.config* file in *C:/django* and write the following content into it:



* Create *static/* folder within *C:/django* to hold static files for your app. Within *C:/django/static*, create a *web.config* file and add the following content:



* Edit *C:/django/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 3.6 in C:/Python36, and ensure it is added to *system path*. (Go with custom installation to be able to customize installation location)
* Use the terminal as administrator to install and enable *wfastcgi*
* Open a CMD terminal as Administrator, and run the command *pip install wfastcgi*
* Afterwards, run the command *wfastcgi-enable*
* Copy the Python path, and replace the scriptProcessor="<to be filled in>" in *C:/django/web.config* file with the Python path returned by *wfastcgi-enable*.
* Edit *C:/django/requirements2.txt*: add *boto3*, replace *slackclient* 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)
* Using the terminal, navigate into C:/django and run *pip install -r requirements2.txt.* This will install all required packages globally.
* Run *manage.py makemigrations* and fix errors if any
* Run *manage.py migrate* to set up tables in your database
* Run *manage.py createsuperuser --username <your-name>* , to create a superuser account.

1. **Application, Application pool, handlers and virtual directory set up on IIS:**

* Open Internet Information Services (IIS) Manager. Under connections select the server, then in the center pane under Management select Configuration Editor. Under Section select system.webServer/handlers. Under Section select Unlock Section. This is required because the default website configuration creates a [route handler](https://pypi.org/project/wfastcgi/#route-handlers) for the whole project.
* *On the IIS console,* right click on application pools and add a new application pool using *django* as name. Leave other fields at default. Application pool defines the environment that will launch the application.
* Right click on Default Web Site and add a new application. Set alias as django. Select the application pool created earlier. Set *C:/django* as the physical path.
* Right click on Default Web Site, and add a virtual directory. Alias should be *static* and the physical path should point to *C:/django/static.* Static files are stored in virtual directories on IIS.

1. **Collect static files:**

* Navigate into *C:/django* via the terminal and run *manage.py collectstatic*. This should collect all static files into *C:/django/static*

1. **Configure the URL to recognise your app name (django):**

* IIS will serve the django app at <http://localhost/django>. So it is necessary to reconfigure the django app URL’s to accommodate the preceding ‘django’.
* Edit *C:/django/ScrumMaster/urls.py* to look like this:



1. **Grant full access to python to the application pool django app uses:**

* Navigate to C:/, right-click on Python36, and edit Properties. Under Security, add IIS AppPool\django and grant full control. App pools are responsible for launching applications. So you need to give it permission to python so that it can run your django application.

**8.** **Allow all traffic in your server’s security group and/or firewall:**

* Edit the inbound rule of your server’s security group and allow *all traffic*. Source should be set to *Anywhere*

**9. Refresh the server and navigate to localhost:**

* Head over to IIS console and refresh the Default Web Site.
* Navigate to <http://localhost/django> (or [http://IP-address/django](http://ip-address/django) or <https://domain/django> ) to view the django application.

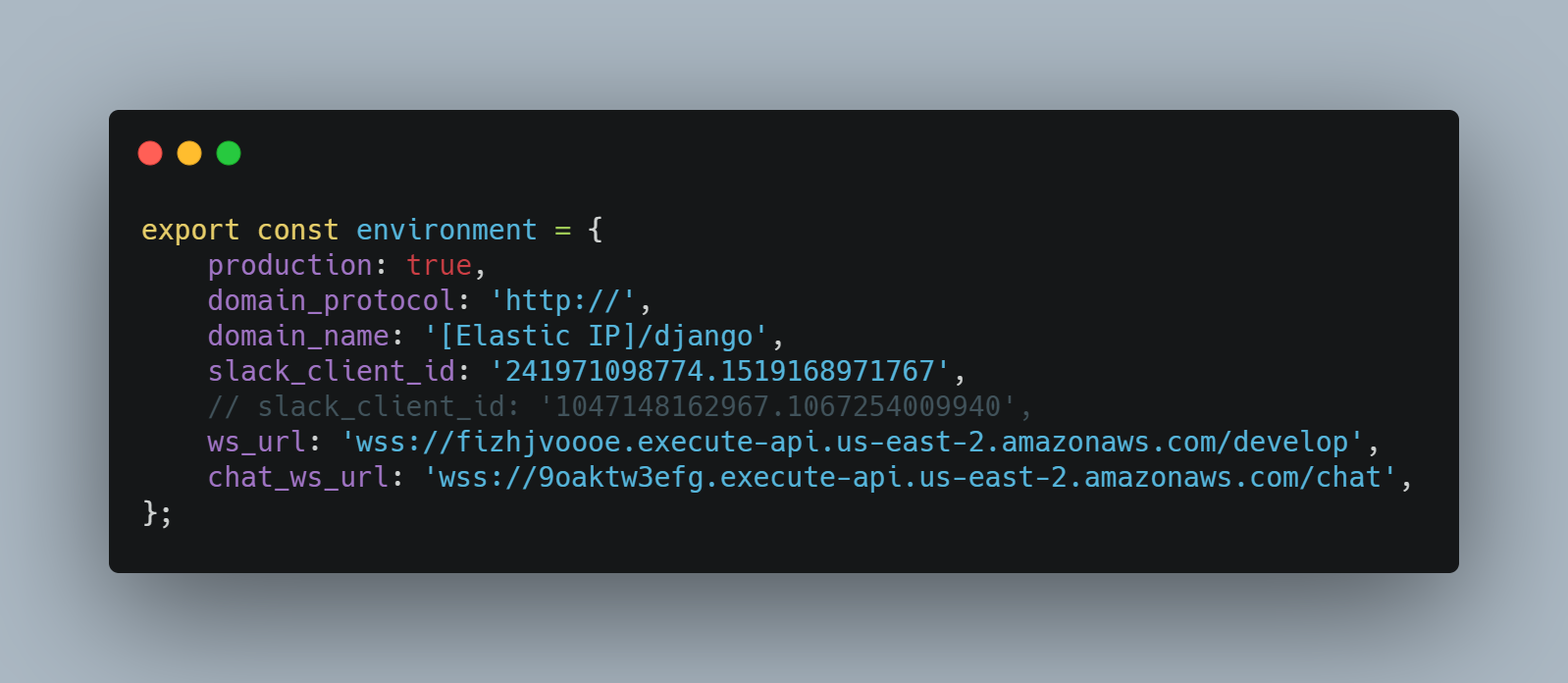
10. **Create a record in Chatscrum Slack Apps table:**

* Navigate to [http://IP-address/django/admin](http://ip-address/django/admin) and login with your superuser credentials
* You will be needing a Chatscrum Slack App record in place before authentication can be carried out successfully. For now, fill in random content in the three fields. It will suffice.

**Deploying the Angular Part on IIS**

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

* One thing to note in this step has to do with paths specified in your html files. Preceding paths with a forward slash (/) will not work in production environment, so it is necessary you remove all preceding slashes in reference paths and hrefs. This option will work well in development, as well as production environment. Use the code editor to edit the codes.
* Set your environment variables in *environments.prod.ts* as this is the file that production environment will take cognizance of. Replace [Elastic IP] with the actual elastic IP of your server. The domain\_protocol + domain\_name setup just points to our django app for communication with the Angular part:



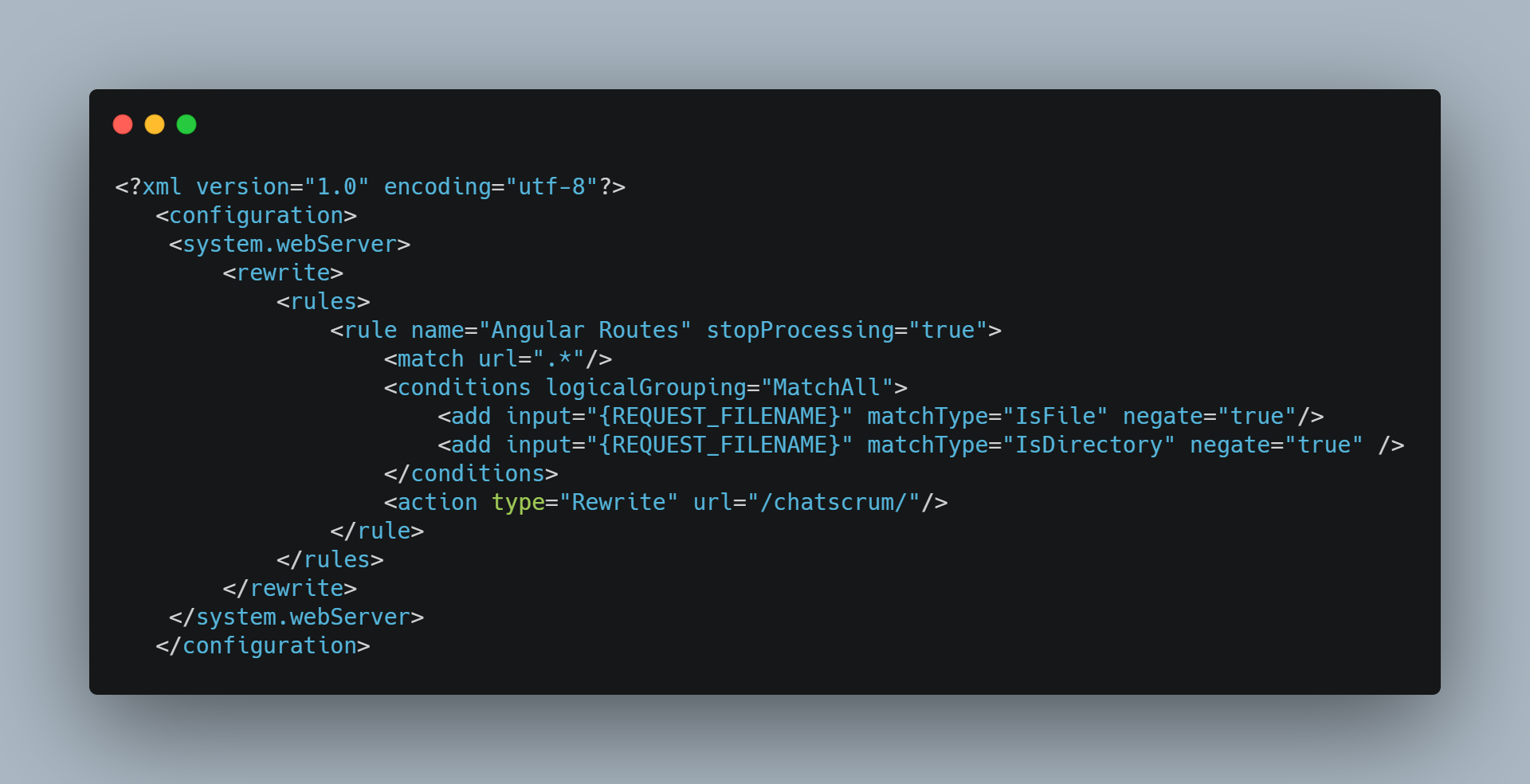
* Within the angular workspace directory (Chatscrum-Angular), run *npm install* from the terminal to install all required packages. At this point, you can test the app if everything is working fine before proceeding to the next steps. Run *ng serve* to start up a local server and then navigate to [http://localhost:4200](http://localhost:4200/) via your machine’s browser to view and test your app.
* Run *ng build --prod --base-href /chatscrum*/ from within the workspace directory. This will compile the Angular app into an output directory named *dist/.* *–base-href* defines the reference to the application in production. *dist/* will contain *chatscrum/* folder ready for production deployment.

2. **Copy the project folder onto the server’s C: drive:**

* Copy *chatscrum/* directory within *dist/*, and paste into *C: drive*
* Create a configuration file *web.config* within *C:/chatscrum* folder, and write the following xml configuration into the file:

· .

·



**3.** **Create a new application pool for your angular app:**

* From the IIS console, create a new application pool. It is good practice to have each application within a website use a separate application pool.

**4.** **Create a new application within the Default Web Site on IIS:**

* From the IIS console, add a new application to the default website. Name it *chatscrum*, and select the *C:/chatscrum* as the physical path. Also select the application pool created in (3).

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

* Replace [IP ADDRESS] with the elastic IP of your server
* Or if you have configured a domain name, navigate to [https://<domain>/chatscrum](https://domain/chatscrum)