**Chatbot Dev Botkits Setup in EC2 | Setup Ngnix Engine | Cloudfront Routing 🡪 ELBs**  
  
**In the “OS: Amazon Linux” EC2 Instance**

**Install nvm, node, npm**

**Commands**

curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.2/install.sh | bash

. ~/.nvm/nvm.sh or source ~/.bashrc

nvm install 16.14.0

nvm -v (checking)

nvm use 16.14

nvm install node

node -v (checking)

npm -v (checking)

--------------------------------------------------

sudo amazon-linux-extras install epel

sudo yum-config-manager --enable epel

**Install Redis**

**Commands**

sudo yum update -y

sudo yum install redis -y

sudo systemctl start redis

sudo systemctl enable redis

sudo systemctl status redis

**To run gst, tap, legal,casetext chatbots**

**# directories where chatbots are located in “a208085-csp-chatbot-nodejs-ec2-preprod” instance**

/usr/bin/triva\_chatbot/botkit/gst

/usr/bin/triva\_chatbot/botkit/tap

/usr/bin/triva\_chatbot/botkit/legal

/usr/bin/triva\_chatbot/botkit/casetext

**commands:**  
  
cd /usr/bin/triva\_chatbot/botkit

cd gst/

**Copy respective chatbot files from s3 bucket to ec2 instance**

**S3 bucket name:** a208085-triva-pradeepa-websdk-poc

**Command**

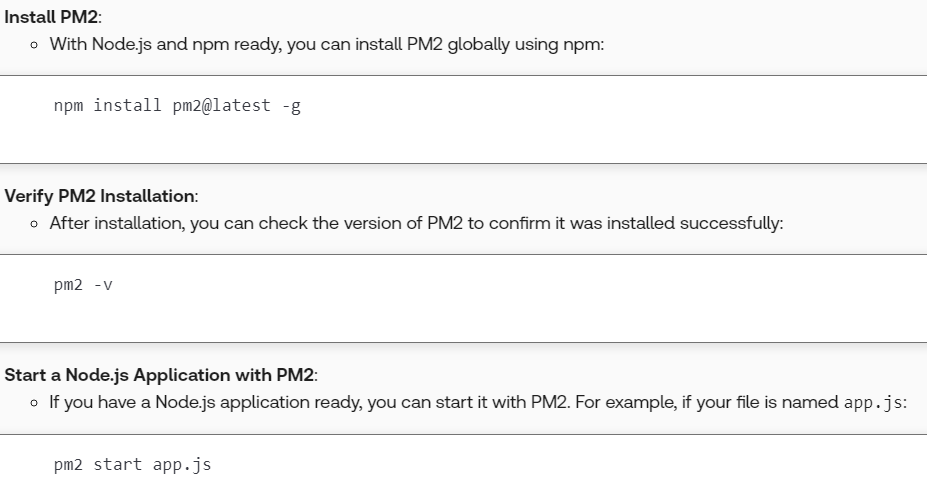
aws s3 cp s3://a208085-triva-pradeepa-websdk-poc/GST\_Dev/ . –recursive

ls

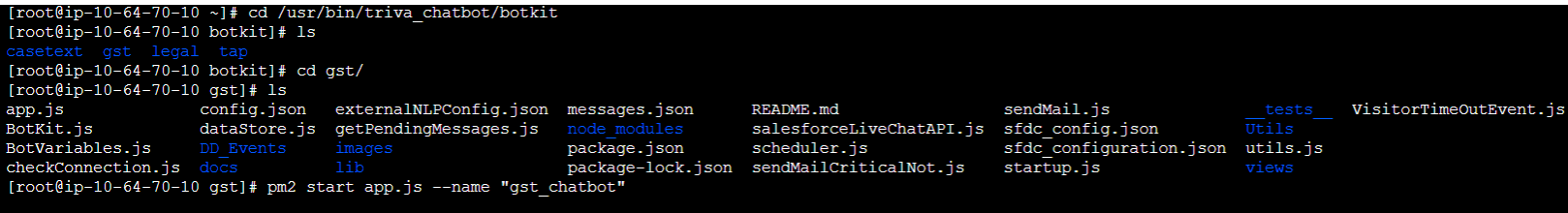
Run ‘npm install’ command to install node\_modules in each chatbot.

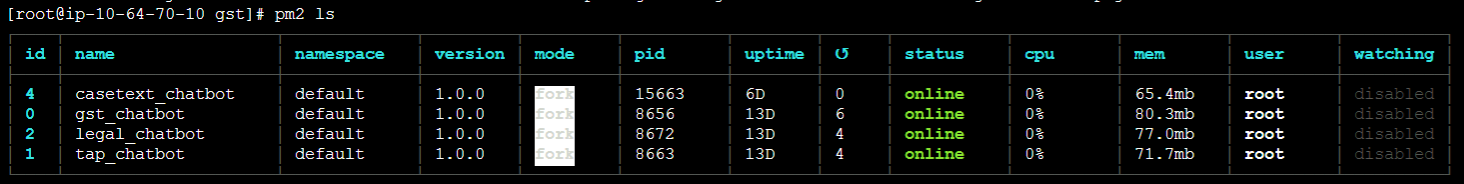
ls

**To start node app forever, install pm2 if not installed.**

****  
**# to start apps by providing names**pm2 start app.js --name “give\_anyname\_to\_app”

pm2 start app.js --name “gst\_chatbot”

**example**  
  
  
**after running all bots**



**Install Ngnix**

**Commands**

sudo amazon-linux-extras install nginx1 -y

sudo systemctl start nginx

sudo systemctl enable nginx

sudo systemctl status nginx

cd /etc/nginx/

**# open nginx configuration file in nano editor**

nano nginx.conf

**# nginx proxy setup**  
server {

listen 80;

listen [::]:80;

server\_name localhost;

root /usr/share/nginx/html;

location /TR\_GST\_DEV {

proxy\_pass http://localhost:8206; # Forward requests to localhost:8206

proxy\_set\_header Host $host; # Preserve the original Host header

proxy\_set\_header X-Real-IP $remote\_addr; # Forward the real IP address

}

}  
  
server {

listen 83;

listen [::]:83;

server\_name localhost;

root /usr/share/nginx/html;

location /TR\_CS\_DEV {

proxy\_pass http://localhost:8003; # Forward requests to localhost:8003

proxy\_set\_header Host $host; # Preserve the original Host header

proxy\_set\_header X-Real-IP $remote\_addr; # Forward the real IP address

}

}  
  
**# after editing nano file**

Ctrl + O to save

Click Enter

Ctrl + x to exit and come out with saved changes

**# Check configuration for syntax errors after making new changes**

sudo nginx -t

**# Reload NGINX to apply changes**

sudo systemctl reload nginx

**Check in EC2 with localhost and proxy pass**

gst

localhost –

curl <http://localhost:8206/TR_GST_DEV/delay>

proxy pass –

curl <http://localhost:80/TR_GST_DEV/delay>

tap

localhost –

curl <http://localhost:8204/TR_TAP_DEV>

proxy pass –

curl <http://localhost:81/TR_TAP_DEV>

legal

localhost –

curl http://localhost:8002/TR\_CS\_DEV

proxy pass –

curl <http://localhost:82/TR_CS_DEV>

casetext

localhost –

curl <http://localhost:60031/TR_CASETEXT_DEV/delay>

proxy pass –

curl <http://localhost:83/TR_CASETEXT_DEV/delay>

**ELB**

* There should be Listener 443
* ELB Security Group inbound rules should be allow listed with ‘cloudfront prefix list security group’.
* Rules with condition path pattern and target groups.
* Each target group should be created with the nginx server ports respectively and should be registered with EC2 instance.

**Cloudfront**

* Distribution
* Add ELB as a origin
* Add path pattern in behavior (this path pattern and ELB Rule-target group condition path pattern should be same)

**check in EC2 with cloudfront alternate domain**  
curl <https://ltw-poc-triva.thomsonreuters.com/TR_GST_DEV/delay>  
  
  
**check in web**

<https://ltw-poc-triva.thomsonreuters.com/TR_GST_DEV/delay>

https://ltw-poc-triva.thomsonreuters.com/TR\_TAP\_DEV

https://ltw-poc-triva.thomsonreuters.com/TR\_CS\_DEV

<https://ltw-poc-triva.thomsonreuters.com/TR_CASETEXT_DEV/delay>  
  
  
Further entire setup is automated with CI/CD Pipeline…