**Steps to use grpc (in python and angular)**

**Installing grpc**

The following should be installed in the system:

* Python 2.7, or Python 3.4 or higher
* pip version 9.0.1 or higher

Next, run the following commands in the terminal for installing grpc and grpc-tools:

* python -m pip install --upgrade pip
* sudo python -m pip install grpcio
* python -m pip install grpcio-tools (for installing grpc tools)

Run the following to install protocol buffer compiler:

* sudo apt install protobuf-compiler
* sudo pip install google
* sudo pip install six
* sudo pip3 install protobuf

**Using gRPC for code generation**

(Language used - Python)

**Step 1**:

Write a proto file that contains the schema.



**Step 2**:

Create a function that we want to remotely call.

In this case, the function info() returns a list that contains some information about the system configuration.



**Step 3**:

Generate the classes in python using the following command:

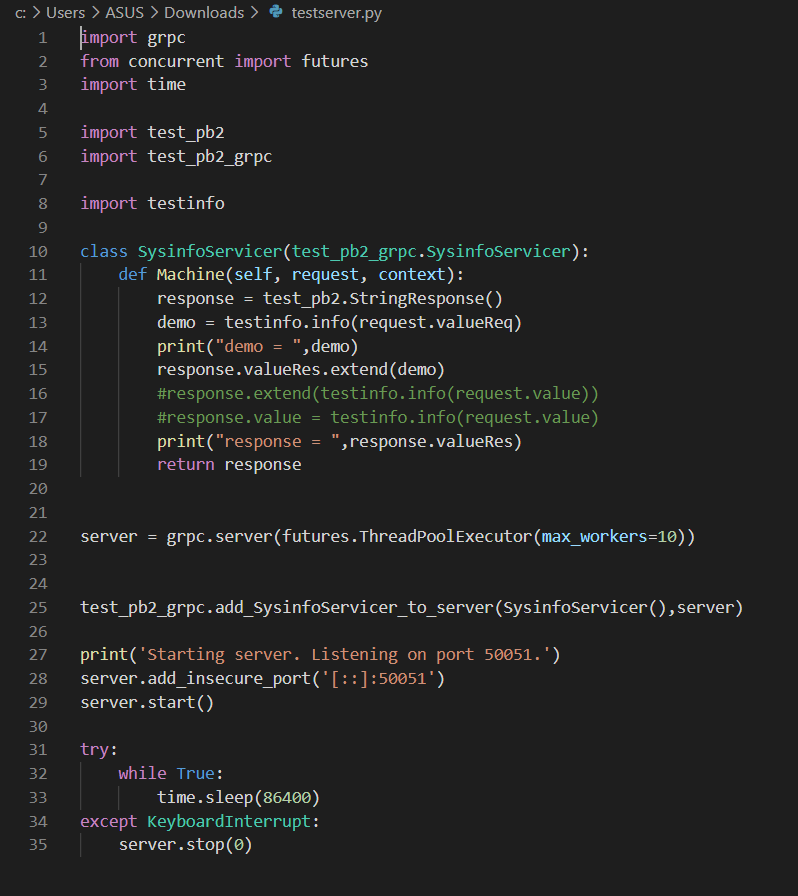
* python3 -m grpc\_tools.protoc -I. --python\_out=. --grpc\_python\_out=. test.proto

The files generated will be as follows:

1. test\_pb2.py - contains message classes.
2. test\_pb2\_grpc.py - contains server and client classes.

**Step 4**:

Create a gRPC server by importing the generated files.



We can start the server using:

python3 testserver.py

**Step 5**:

Create the client using grpc-web (framework used - angular).

gRPC web - A JavaScript implementation of [gRPC](https://grpc.io/) for browser clients. gRPC-web clients connect to gRPC services via a special proxy; by default, gRPC-web uses Envoy.

Create a new angular project.

Run the following in the terminal:

* npm i grpc-web
* npm i protoc-gen-grpc-web
* npm install --save-dev
* npm install --save google-protobuf
* npm install –save-dev @types/google-protobuf
* npm install –save @improbable-eng/grpc-web
* npm i ts-protoc-gen

We can create two new folders for a systematic approach –

Protos – for .proto files

Generated – for generated files.

Now run the following:

* protoc --plugin=protoc-gen-ts="./node\_modules/.bin/protoc-gen-ts" --js\_out="import\_style=commonjs,binary:src/app/generated" --ts\_out="service=grpc-web:src/app/generated" src/app/protos/example.proto

Inside the generated folder the following files will be generated:

* Example.pb.service.d.ts
* Example.pb.service.js
* Example.pb.d.ts
* Example.pb.js

**Step 6**:

Install envoy and create an envoy file.

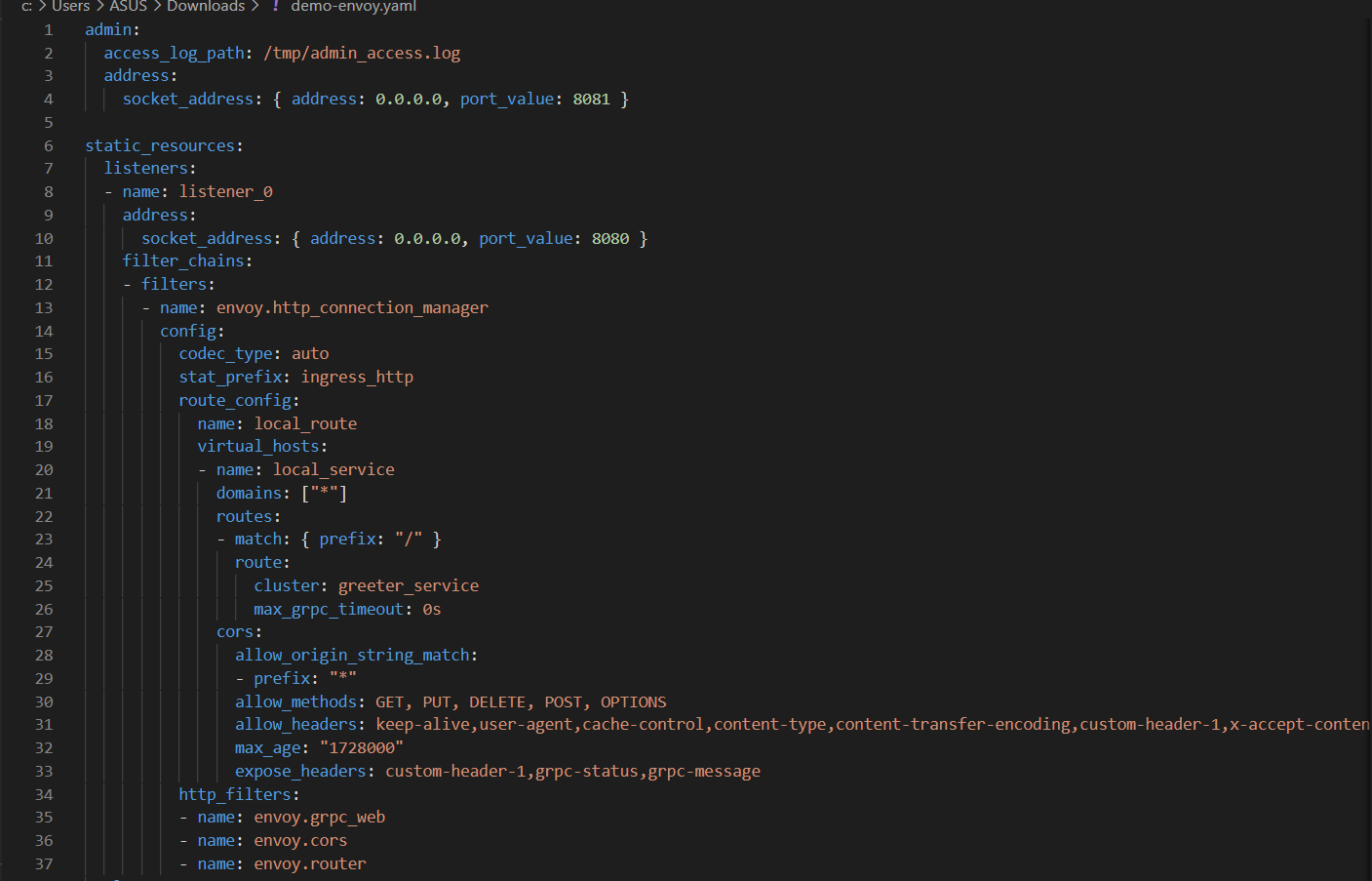
For installation of envoy follow the steps given below (in linux) :

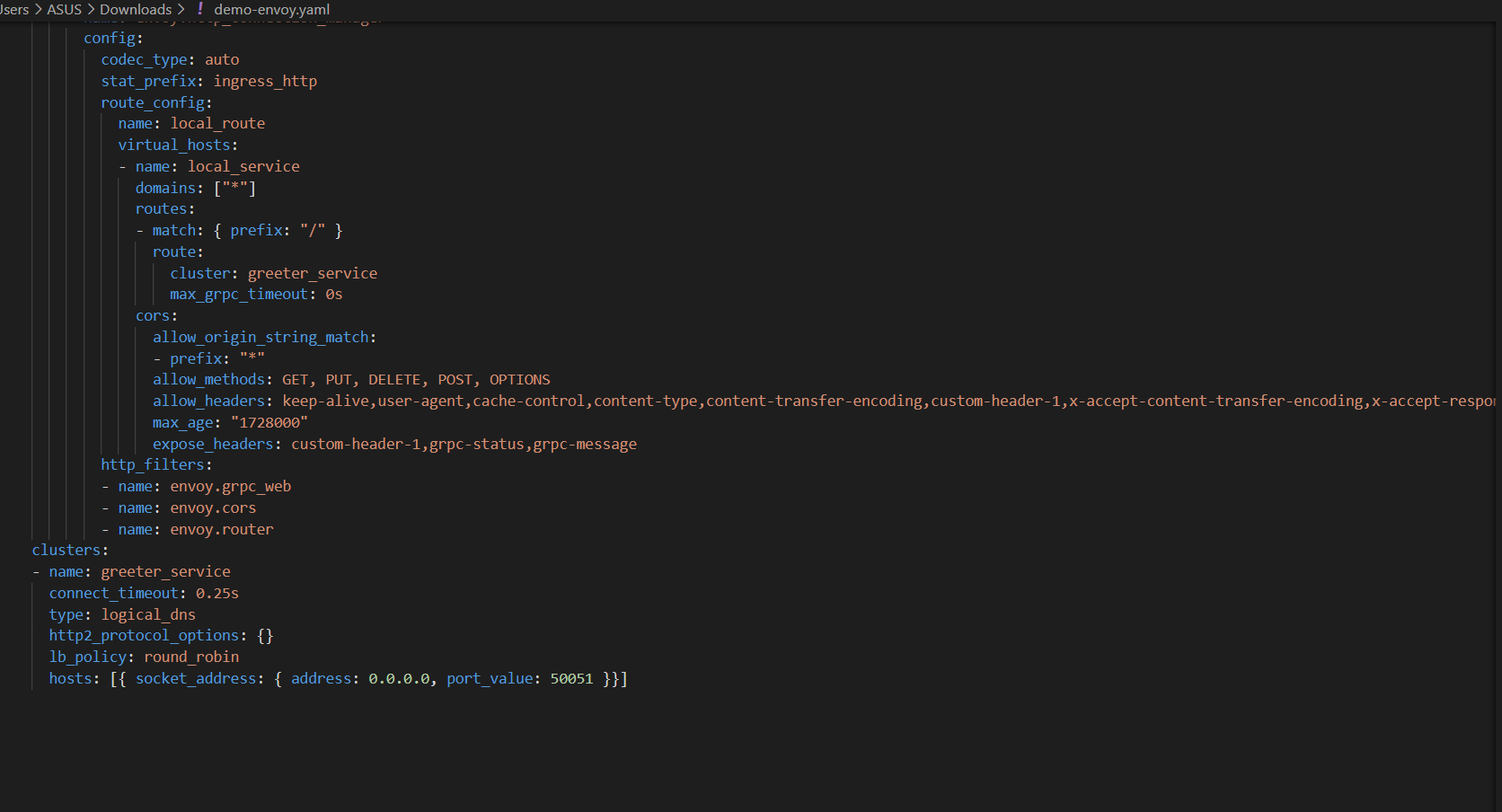
* sudo apt update
* sudo apt install apt-transport-https ca-certificates curl gnupg-agent software-properties-common
* curl -sL '<https://getenvoy.io/gpg>' | sudo apt-key add -
* # verify the key
* apt-key fingerprint 6FF974DB | grep "5270 CEAC"
* sudo add-apt-repository "deb [arch=amd64] <https://dl.bintray.com/tetrate/getenvoy-deb> $(lsb\_release -cs) stable"
* sudo apt update
* sudo apt install -y getenvoy-envoy

For more details:

<https://www.envoyproxy.io/docs/envoy/latest/start/install#install-envoy-on-ubuntu-linux>

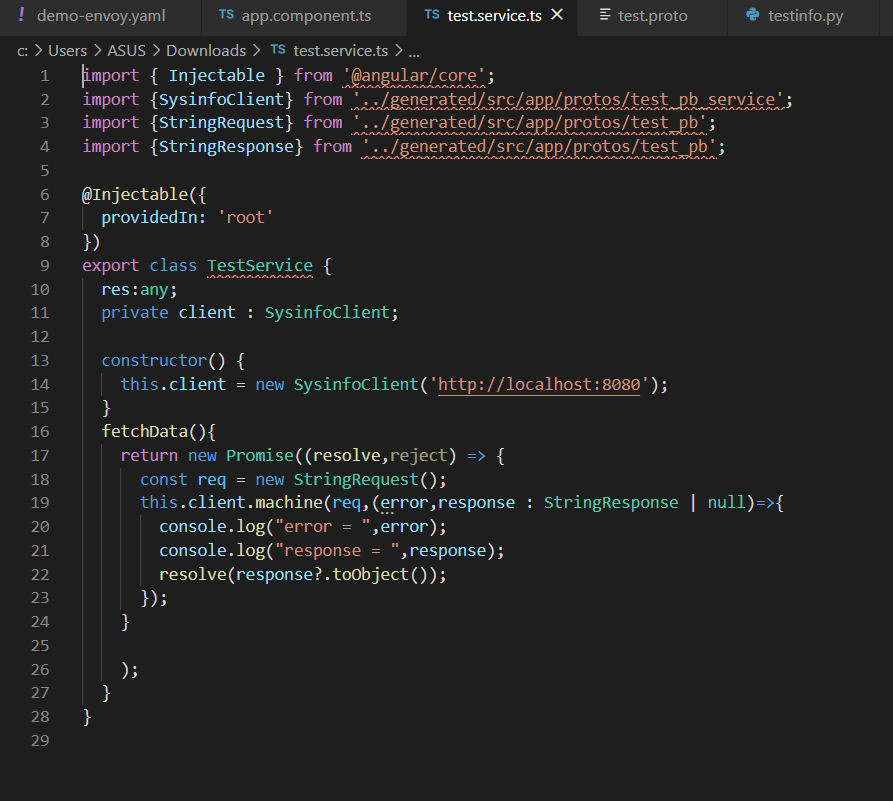
Create the configuration file for envoy proxy





**Step 7**:

Write a service in angular to use the envoy.yaml file and to fetch data from the grpc server.

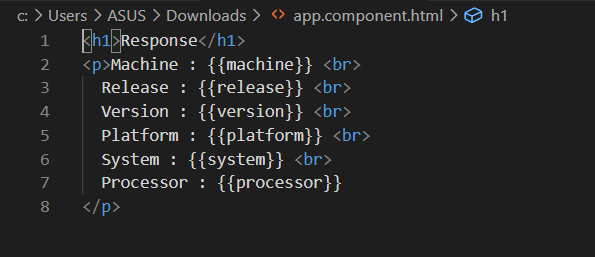


And then use the service made above in the app.component.ts file.



**Step 8**:

Write the code to display data on the screen.



**Step 9**:

* Start the grpc server with the command -

Python3 filename.py

* Start the envoy with the command -

Envoy –c {{path to the envoy file}}/filename.yaml

* Start the angular server with the command -

ng serve

Note:

Grpc server - localhost:50051

Envoy - localhost:8080

Angular server – localhost:4200

**The output would be as follows:**

