

# gRPC Documentation:

## Installation:

1. `export MY_INSTALL_DIR=$HOME/.local`
2. `mkdir -p $MY_INSTALL_DIR`
3. `export PATH="$MY_INSTALL_DIR/bin:$PATH"`
4. `sudo apt install cmake`
5. If 4 doesn't work:  
`wget -q -O cmake-linux.sh`  
[https://github.com/Kitware/CMake/releases/download/v3.19.6/cmake-3.19.6-Linux-x86\\_64.sh](https://github.com/Kitware/CMake/releases/download/v3.19.6/cmake-3.19.6-Linux-x86_64.sh)  
`sh cmake-linux.sh -- --skip-license --prefix=$MY_INSTALL_DIR`  
`rm cmake-linux.sh`
7. `sudo apt install -y build-essential autoconf libtool pkg-config`
8. `git clone --recurse-submodules -b v1.58.0 --depth 1 --shallow-submodules https://github.com/grpc/grpc`

### Build and install gRPC and Protocol Buffers\_

1. `cd grpc`
2. `mkdir -p cmake/build`
3. `pushd cmake/build`
4. `cmake -DgRPC_INSTALL=ON \`  
`-DgRPC_BUILD_TESTS=OFF \`  
`-DCMAKE_INSTALL_PREFIX=$MY_INSTALL_DIR \`  
`../..`
5. `make -j 4`

6. `make install`

7. `popd`

## Running an Example:

### Hello World Example:

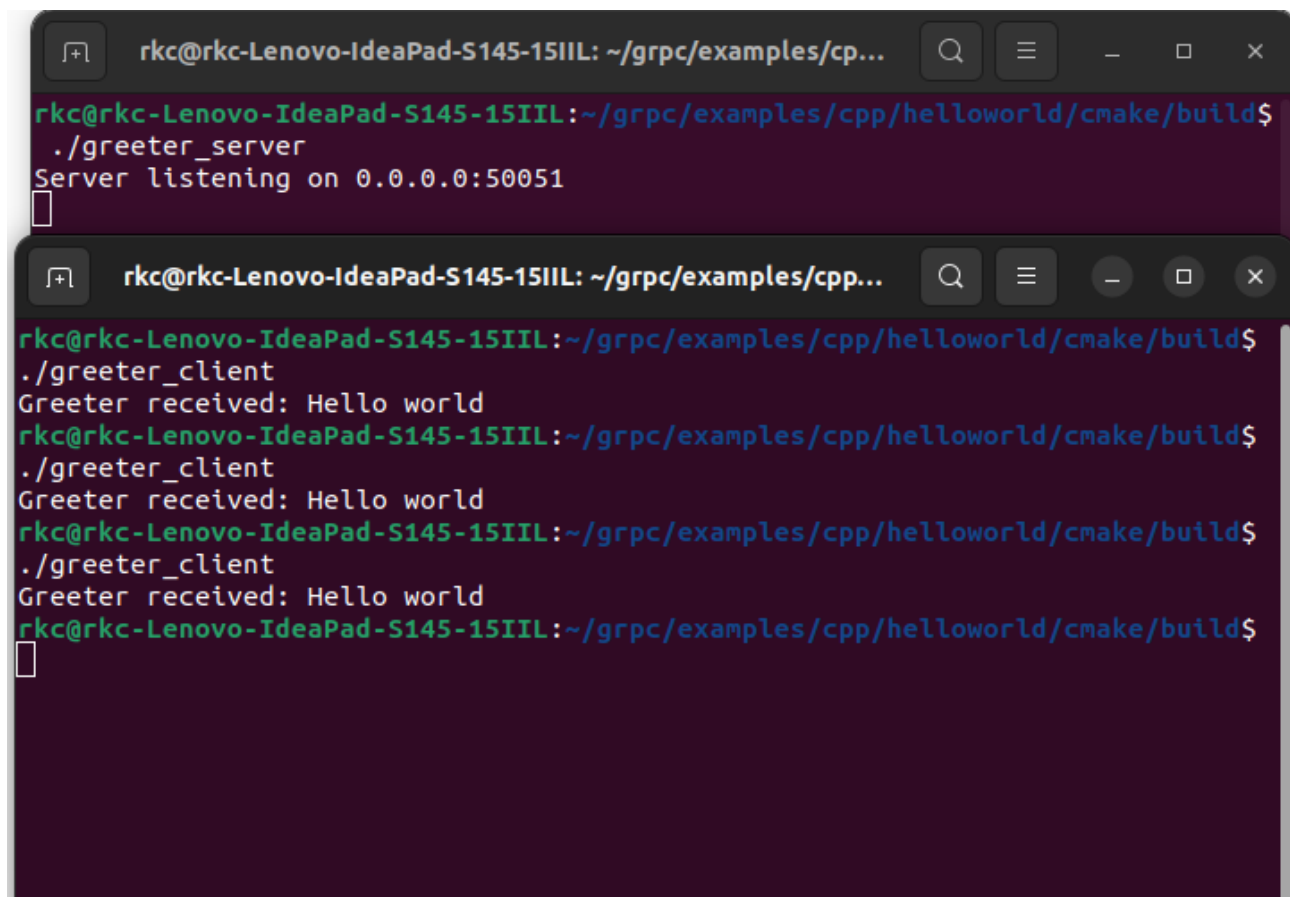
```
cd examples/cpp/helloworld
```

```
mkdir -p cmake/build
```

```
$ pushd cmake/build
```

```
$ cmake -DCMAKE_PREFIX_PATH=$MY_INSTALL_DIR ../..
```

```
$ make -j 4
```



```
rkc@rkc-Lenovo-IdeaPad-S145-15IIL: ~/grpc/examples/cpp...  
rkc@rkc-Lenovo-IdeaPad-S145-15IIL:~/grpc/examples/cpp/helloworld/cmake/build$  
./greeter_server  
Server listening on 0.0.0.0:50051  
  
rkc@rkc-Lenovo-IdeaPad-S145-15IIL:~/grpc/examples/cpp...  
rkc@rkc-Lenovo-IdeaPad-S145-15IIL:~/grpc/examples/cpp/helloworld/cmake/build$  
./greeter_client  
Greeter received: Hello world  
rkc@rkc-Lenovo-IdeaPad-S145-15IIL:~/grpc/examples/cpp/helloworld/cmake/build$  
./greeter_client  
Greeter received: Hello world  
rkc@rkc-Lenovo-IdeaPad-S145-15IIL:~/grpc/examples/cpp/helloworld/cmake/build$  
./greeter_client  
Greeter received: Hello world  
rkc@rkc-Lenovo-IdeaPad-S145-15IIL:~/grpc/examples/cpp/helloworld/cmake/build$
```

## Creating own service routines:

Files needed:

1. .proto (to be created in examples/protos)
2. Makefile (in working directory)

`make your_service.grpc.pb.o`

Which generates the files in working directory:

1. `route_guide.pb.h`, the header which declares your generated message classes
2. `route_guide.pb.cc`, which contains the implementation of your message classes
3. `route_guide.grpc.pb.h`, the header which declares your generated service classes
4. `route_guide.grpc.pb.cc`, which contains the implementation of your service classes

3. Create client and servers.  
`your_service_server.cc` and  
`your_service_client.cc`

`make` – to run them and generate executable files

(<https://grpc.io/docs/languages/cpp/basics/>)

## Source:

1. <https://grpc.io/docs/languages/cpp/quickstart/#install-grpc>
2. <https://grpc.io/docs/languages/cpp/basics/>
3. <https://grpc.io/docs/what-is-grpc/introduction/>
4. [https://github.com/grpc/grpc/blob/v1.58.0/examples/cpp/route\\_guide/CMakeLists.txt](https://github.com/grpc/grpc/blob/v1.58.0/examples/cpp/route_guide/CMakeLists.txt)