How to install:
type(for the pre-requisites): sudo apt-get install libboost-dev libboost-test-dev libboost-program-options-dev libevent-dev automake libtool flex bison pkg-config g++ libssl-dev

sudo apt-get install php5-dev php5-cli (for php)
sudo apt-get install libglib2.0-dev (for c_glib)

for the final installation, download the tarball from the website, http://thrift.apache.org/download/

then to install from the tarball, extract the files using:
tar -xvf /path/to/tarball
cd /path/to/extraction
./configure
sudo make
sudo make install

Also install the eclipse editor for thrift files.
Eclipse --> help --> Install new Software -->
add the URL: http://thrift4eclipse.sourceforge.net/updatesite/
tick the only package shown and install it.

Now Compiling the required libraries(for different languages) :

1. **for JAVA**
   Go to folder /path/to/thrift-version/folder/lib/java/
   execute the command "ant" - compiles using apache ant
   Now the build folder contains all the lib files required.
2. **for PHP**
   No need for compiling any files, php is used in its raw form.

Make the thrift file now.
Tutorial can be found here : http://diwakergupta.github.com/thrift-missing-guide/
Thrift file will include all the services and structures shared between the two languages.
Start with: namespace java <package-name>

Making the JAVA server:
Make a new project in Eclipse with type, "Dynamic Web Project".
Put the "thrift file" in the <project-name>/Java Resources/src/ folder.
Copy the lib files (libthrift-<version>.jar, build/lib/*) to <project-name>/WebContent/WEB-INF/lib/ folder.

Generate the auto-generated java files from the file using the command:
cd path/to/thrift-file/
thrift --gen java -out . <thrift-file-name>

Now we have to implement the services mentioned in the thrift-file by:

1. make a new file in the same package <package-name>.
2. Wrie a class <service-implement> implementing <service-name>.Iface (like this implement all the services)

Now we have to make the server file:
Make a new class implementing "Runnable".
The code is :

```
    public class <Server-name> implements Runnable {
```

```
		/* port to listen */
		private static final int PORT =  9090;

		public void run() {
			try {
				TServerSocket serverTransport = new TServerSocket(PORT);
				HelloService.Processor processor = new 
	HelloService.Processor(new <service-implement>());
				TServer server = new TThreadPoolServer(new 
	TThreadPoolServer.Args(serverTransport).processor(processor));
				System.out.println("Starting server on port: "+PORT);
				server.serve();

			} catch(TTransportException e) {
				System.out.println("Message: "+e.getMessage());
				System.out.println("StackTrace: ");
				e.printStackTrace();
			}
		}

		public static void main(String[] args) {
			new Thread(new <Server-name>()).run();
		}
	}
```

Run the server as a java application.
NOTE: To stop the server you'll need to kill the process via the console.

This completes the making of the server.

Making the PHP client:
First auto-generate the php package from thrift file using the command:
cd path/to/thrift-file/
thrift --gen php <thrift-file>

make a new folder named “thrift” and copy all the php library files in the folder /path/to/thrift-version/folder/lib/php/src/ to this new folder. Also make a new folder named “packages” in “thrift” folder, which now contains the auto-generated php package.

Make a new file <client-file>.php adjacent to the “thrift” folder containing the php library.
Contents of the php file will be:

```
	<?php
		// defining the port and server to listen
		define("PORT", '9090');
		define("SERVER", 'localhost');

		//Global variable where the php library files are stored
		$GLOBALS['THRIFT_ROOT'] = 'thrift';

		//including the library files
		require_once $GLOBALS['THRIFT_ROOT'].'/Thrift.php';
		require_once $GLOBALS['THRIFT_ROOT'].'/protocol/TBinaryProtocol.php';
		require_once $GLOBALS['THRIFT_ROOT'].'/transport/TSocket.php';
		require_once $GLOBALS['THRIFT_ROOT'].'/transport/TBufferedTransport.php';

		//loading the auto-generated package
		require_once $GLOBALS['THRIFT_ROOT'].'/packages/hello/HelloService.php';
	?>
	<?php
```

```
		try {
			//create a thrift connection
			$socket = new TSocket(SERVER, PORT);
			$transport = new TBufferedTransport($socket);
			$protocol = new TBinaryProtocol($transport);

			//create a new hello service client
			$client = new HelloServiceClient($protocol);

			//open the connection
			$transport->open();

			$result = $client->sayHello();
			echo "Result: ".$result;

			$transport->close();
		} catch(TException $tx) {
			echo "Thrift Exception: ".$tx->getMessage()."\r\n";
		}
	?>
```

To Test:

Run the java server.

```
Console: “Starting server on port: 9090”
```

Run the <client-file>.php using the command: php5 client.php

```
Console: “Result: HelloWorld!!”
```

Finally make a folder “client” and copy the client related files there. Also, make a new folder named “server”, copy all the java server files there. So we finally we have a simple apache thrift application making a bridge between java(server) and php(client). :D