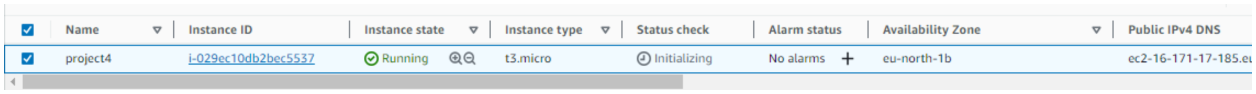
PROJECT 4: MEAN STACK DEVELOPMENT ON AWS

MEAN Stack is a combination of following components:

- MongoDB (Document database) – Stores and allows to retrieve data.
- Express (Back-end application framework) – Makes requests to Database for Reads and Writes.
- Angular (Front-end application framework) – Handles Client and Server Requests
- Node.js (JavaScript runtime environment) – Accepts requests and displays results to end user

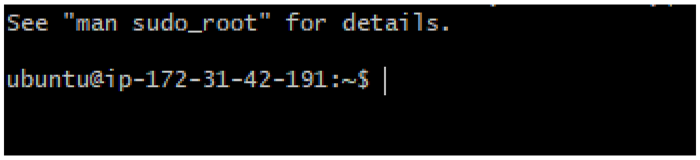
Step 1: Launch an EC2 instance on AWS

We will do this using the free tier account.



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
✓	project4	i-029ec10db2bec5537	Running	t3.micro	Initializing	No alarms	eu-north-1b	ec2-16-171-17-185.eu

Login to the server



```
See "man sudo_root" for details.

ubuntu@ip-172-31-42-191:~$ |
```

Install the following using the commands below

Update ubuntu

sudo apt update

Upgrade ubuntu

sudo apt upgrade -y

Add certificates

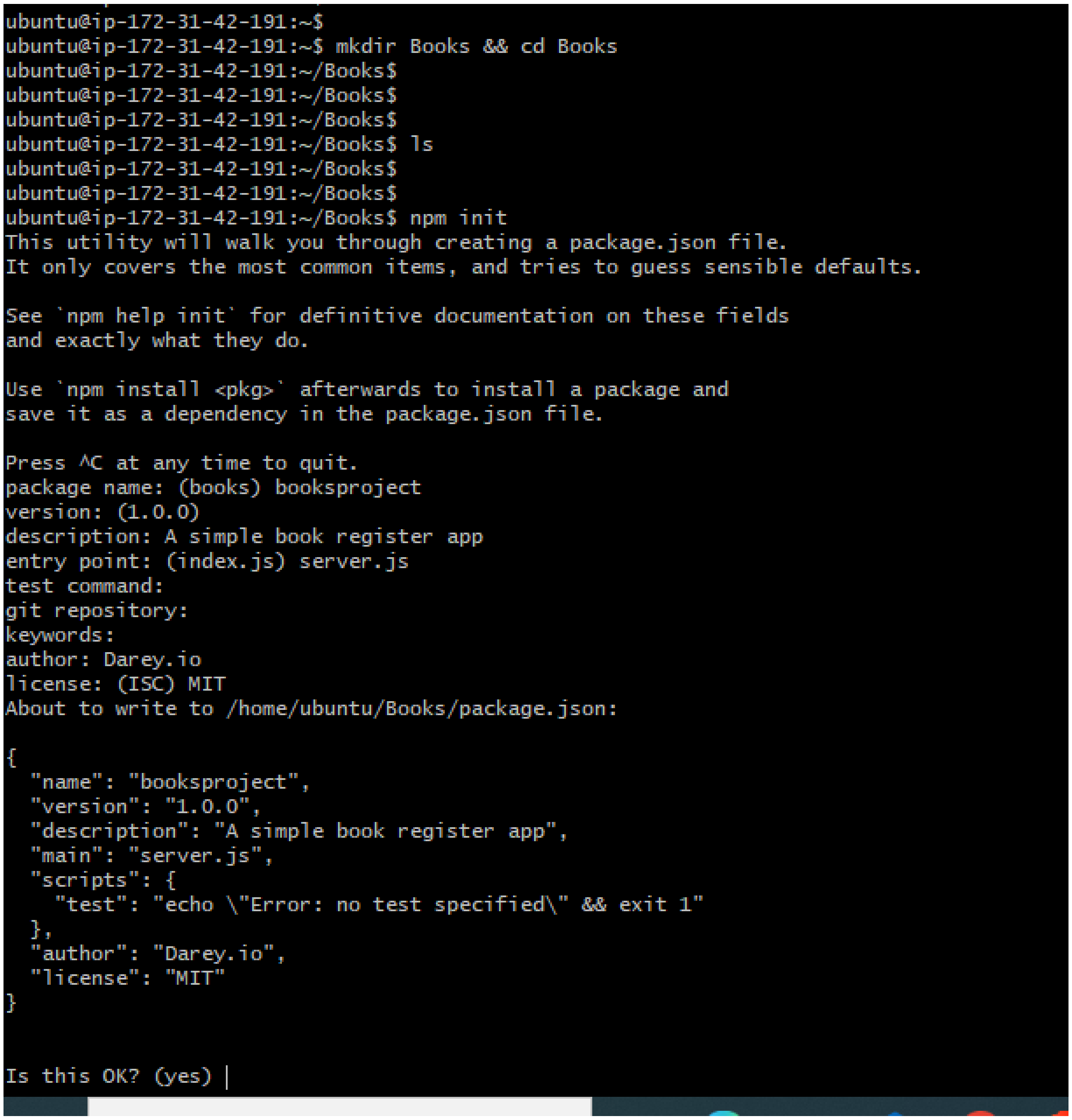
sudo apt -y install curl dirmngr apt-transport-https lsb-release ca-certificates

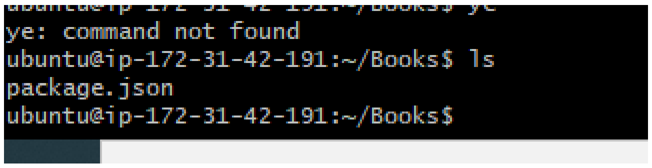
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -

Install NodeJS

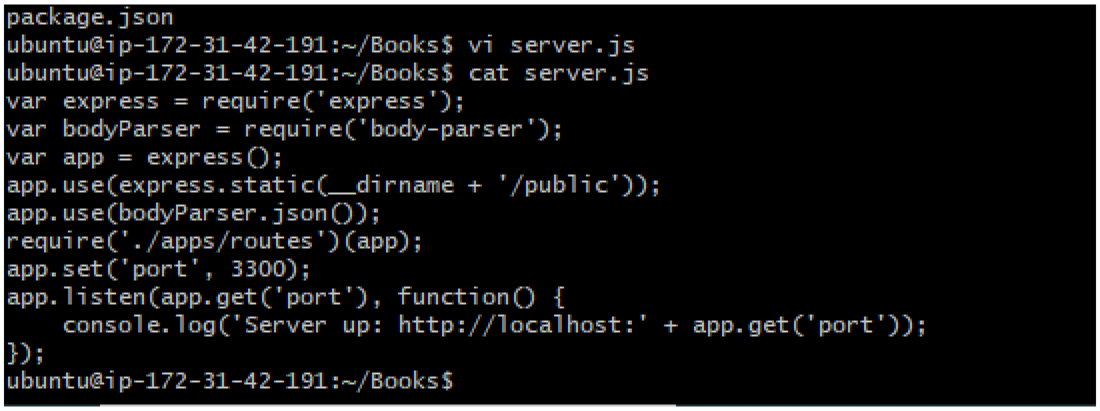
sudo apt install -y nodejs

Step 2 Install MongoDB


```
ubuntu@ip-172-31-42-191:~$
ubuntu@ip-172-31-42-191:~$ mkdir Books && cd Books
ubuntu@ip-172-31-42-191:~/Books$
ubuntu@ip-172-31-42-191:~/Books$
ubuntu@ip-172-31-42-191:~/Books$
ubuntu@ip-172-31-42-191:~/Books$ ls
ubuntu@ip-172-31-42-191:~/Books$
ubuntu@ip-172-31-42-191:~/Books$
ubuntu@ip-172-31-42-191:~/Books$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (books) booksproject
version: (1.0.0)
description: A simple book register app
entry point: (index.js) server.js
test command:
git repository:
keywords:
author: Darey.io
license: (ISC) MIT
About to write to /home/ubuntu/Books/package.json:

{
  "name": "booksproject",
  "version": "1.0.0",
  "description": "A simple book register app",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Darey.io",
  "license": "MIT"
}


Is this OK? (yes) |
```

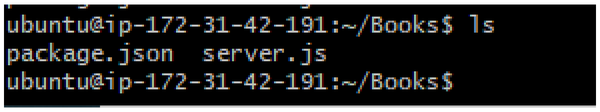

```
ubuntu@ip-172-31-42-191:~/Books$ ye
ye: command not found
ubuntu@ip-172-31-42-191:~/Books$ ls
package.json
ubuntu@ip-172-31-42-191:~/Books$
```

Creating the server.js file and writing the command and saving it



```
package.json
ubuntu@ip-172-31-42-191:~/Books$ vi server.js
ubuntu@ip-172-31-42-191:~/Books$ cat server.js
var express = require('express');
var bodyParser = require('body-parser');
var app = express();
app.use(express.static(__dirname + '/public'));
app.use(bodyParser.json());
require('./apps/routes')(app);
app.set('port', 3300);
app.listen(app.get('port'), function() {
    console.log('Server up: http://localhost:' + app.get('port'));
});
ubuntu@ip-172-31-42-191:~/Books$
```

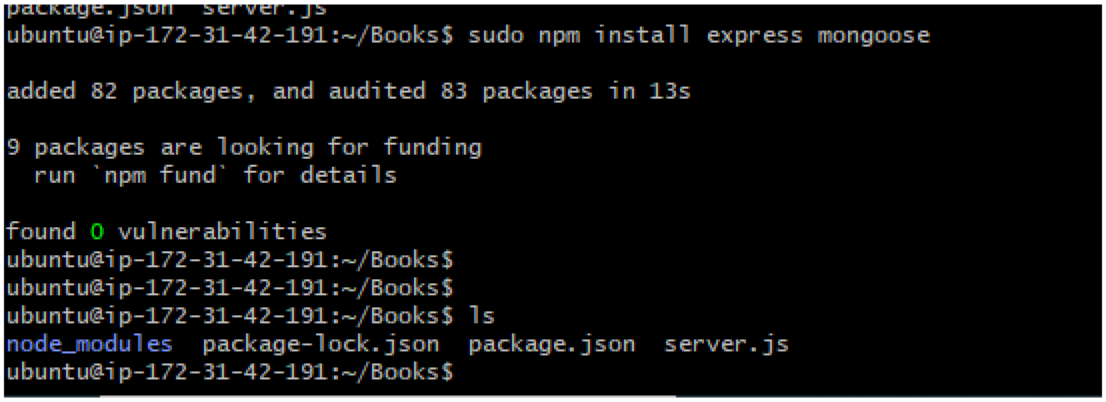
LS to list the files in the book direcotry



```
ubuntu@ip-172-31-42-191:~/Books$ ls
package.json  server.js
ubuntu@ip-172-31-42-191:~/Books$
```

Installing express mongoose

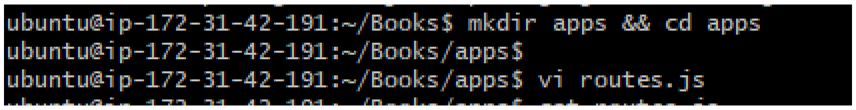
Notice that the files nin theBook directory is increasing as we are installing



```
package.json  server.js
ubuntu@ip-172-31-42-191:~/Books$ sudo npm install express mongoose

added 82 packages, and audited 83 packages in 13s

9 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
ubuntu@ip-172-31-42-191:~/Books$
ubuntu@ip-172-31-42-191:~/Books$
ubuntu@ip-172-31-42-191:~/Books$ ls
node_modules  package-lock.json  package.json  server.js
ubuntu@ip-172-31-42-191:~/Books$
```

Create app directory in book directory and change directry to app

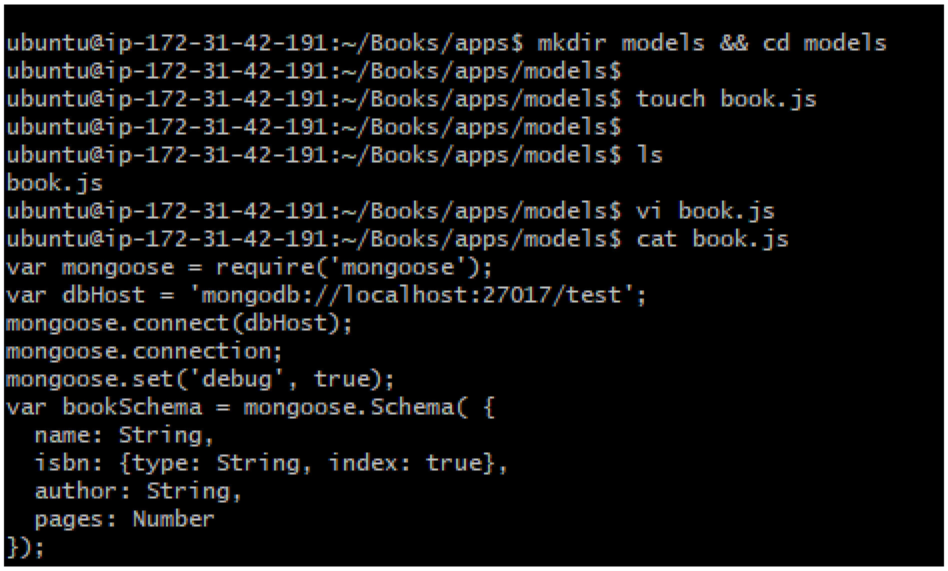
Create routes.js in app adirectory and copy the code below in it . you can cat   route.js to ensure you pasted the right write up in it



```
ubuntu@ip-172-31-42-191:~/Books$ mkdir apps && cd apps
ubuntu@ip-172-31-42-191:~/Books/apps$
ubuntu@ip-172-31-42-191:~/Books/apps$ vi routes.js
```

In the app directory, create another directory models and change to the model direcotyr

Create a file book.js and copy the code below into it. You can do this using touch or vi command



```
ubuntu@ip-172-31-42-191:~/Books/apps$ mkdir models && cd models
ubuntu@ip-172-31-42-191:~/Books/apps/models$
ubuntu@ip-172-31-42-191:~/Books/apps/models$ touch book.js
ubuntu@ip-172-31-42-191:~/Books/apps/models$
ubuntu@ip-172-31-42-191:~/Books/apps/models$ ls
book.js
ubuntu@ip-172-31-42-191:~/Books/apps/models$ vi book.js
ubuntu@ip-172-31-42-191:~/Books/apps/models$ cat book.js
var mongoose = require('mongoose');
var dbHost = 'mongodb://localhost:27017/test';
mongoose.connect(dbHost);
mongoose.connection;
mongoose.set('debug', true);
var bookSchema = mongoose.Schema( {
  name: String,
  isbn: {type: String, index: true},
  author: String,
  pages: Number
});
```

Cat book.js to eusure the right coed was typed into book.js

Step 4 – Access the routes with AngularJS

We are going to the root irectory Book directory

Use the command below

Cd../..

Make a directory name public and cange directory to public

```
mkdir public && cd public
```

Create a file in public named script.js and add the below to the file

Use the cat command to ensure the correct code is pasted

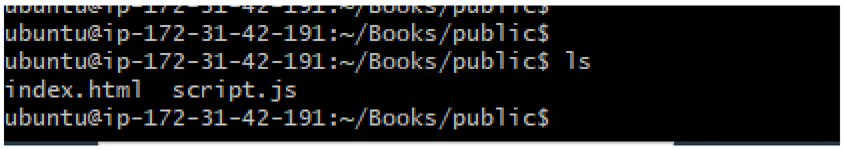
In the public directory, create index.html for the fornt end

n public folder, create a file named index.html;

vi index.html

Cpoy and paste the code below into index.html file.

So in the public directory we ave 2 files now



```
ubuntu@ip-172-31-42-191:~/Books/public$
ubuntu@ip-172-31-42-191:~/Books/public$
ubuntu@ip-172-31-42-191:~/Books/public$ ls
index.html  script.js
ubuntu@ip-172-31-42-191:~/Books/public$
```

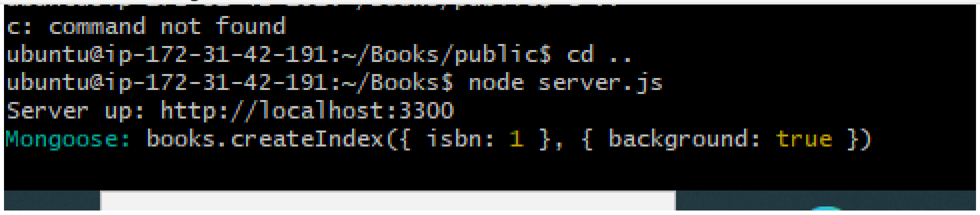
Change directory back to books

cd..

from the books directory, start the server by running this command

Start the server by running this command:

```
node server.js
```



```
c: command not found
ubuntu@ip-172-31-42-191:~/Books/public$ cd ..
ubuntu@ip-172-31-42-191:~/Books$ node server.js
Server up: http://localhost:3300
Mongoose: books.createIndex({ isbn: 1 }, { background: true })
```

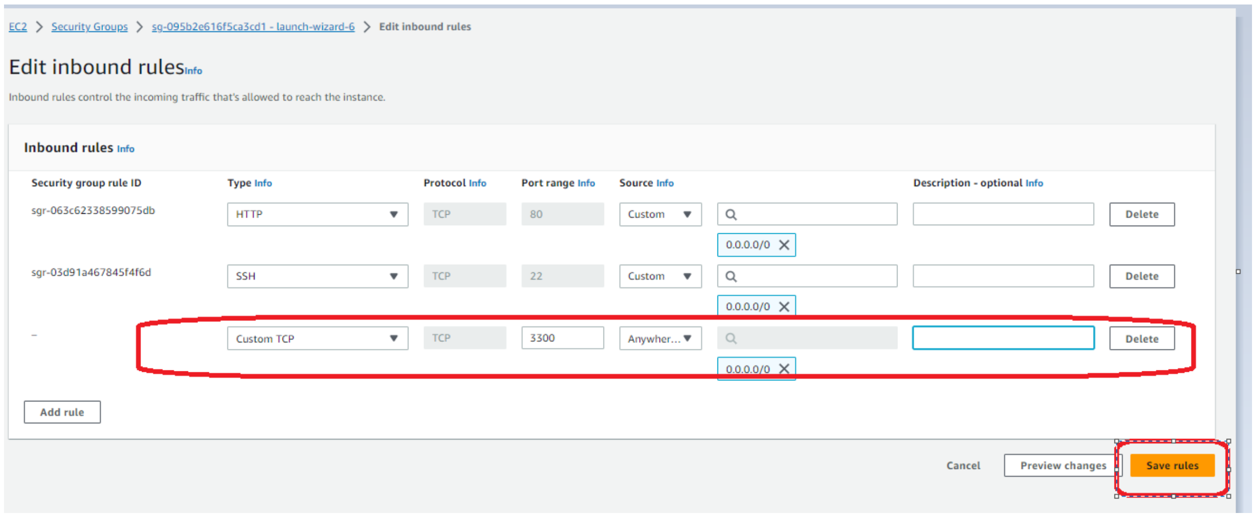
To open the application on a browse, you need to edit the inbound rule and add 3300 port.

To this go to your AWS portal

EC2 instances

Security group

Edit inbound rule



EC2 > Security Groups > sg-095b2e616f5ca3cd1 - launch-wizard-6 > Edit inbound rules

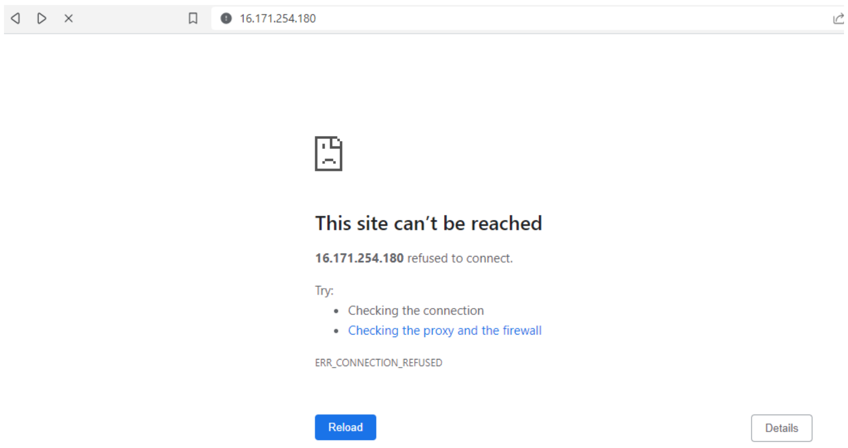
Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-063c62338599075db	HTTP	TCP	80	Custom	0.0.0.0/0		Delete
sgr-03d91a467845f4f6d	SSH	TCP	22	Custom	0.0.0.0/0		Delete
–	Custom TCP	TCP	3300	Anywher...	0.0.0.0/0		Delete

Initially we were ubable to access the app because we had not enabled the application on port 3300. We were getting the error message below



16.171.254.180

This site can’t be reached

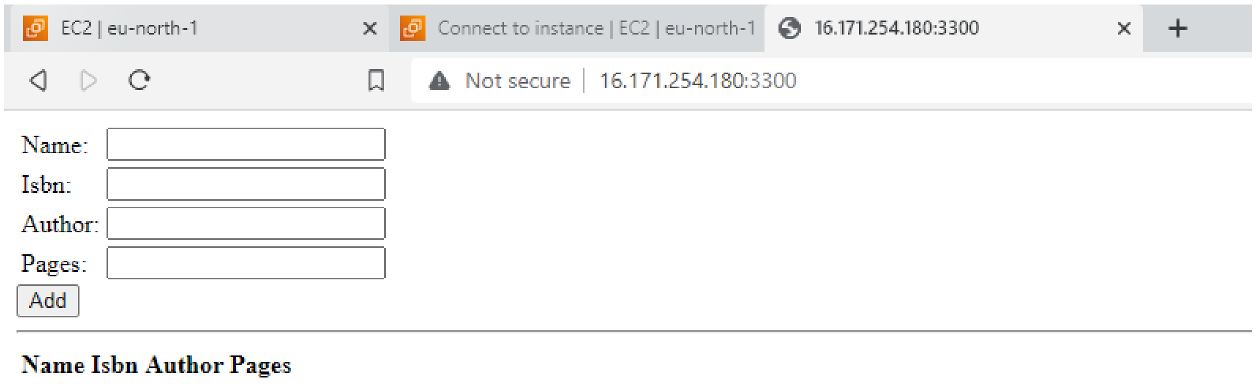
16.171.254.180 refused to connect.

Try:

- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_REFUSED

After enabling tcp on port 3300 this is what we ger.



16.171.254.180:3300

Name:	
Isbn:	
Author:	
Pages:	

Name Isbn Author Pages