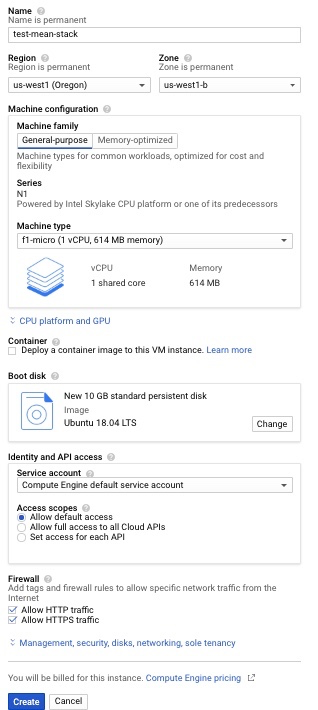
**How to build MEAN Stack via Google Cloud?**

[Link to Github repo](https://github.com/Illthid/itc134-final-project)

Link to Website

[Link to web app tutorial](https://docs.google.com/document/d/1eaw0VFVrWigKKsECKV-mQdEMxzpKIdLrC8vYbHAzTj0/edit?usp=sharing)

1. Turn on your laptop
2. Go to internet explorer (firefox, google chrome, safari...etc)
3. Type: <https://cloud.google.com/>
4. On top right, click **sign-in**
5. After you sign in, go to top right to click on **console**
6. On top left, click what looks like **three stairs** (nav)
7. On the nav, find **Compute Engine** but don’t click it
8. On that Compute Engine, click **VM Instances**
9. On that page, on top, click **CREATE INSTANCE**
10. Type a name like ‘test-mean-stack’ for the purpose of VM
11. Change **region** to say **Oregon**
12. Change the **Machine Type** to **f1-micro**
13. Change the **Boot disk** to **Ubuntu 18.04 LTS**
14. Click **Create** and wait



1. Check to make sure that the instance (test-mean-stack) is running which should be green circle with white check-mark.
2. Using test-mean-stack, find the letters **SSH** on the right and click it which should open up the shell which can take a while to connect to VM so wait.

**Commands:**

1. To install **node.js**, type down: curl -sL https://deb.nodesource.com/setup\_12.x | sudo -E bash -
2. Type down: sudo apt install -y nodejs
3. To check node.js version, type down: node -v
4. npm (Node Package Manager) is for Node.js and usually install together with Node.js. So to check npm version, type down: npm -v
5. To install **angular.js** command line interface , type: sudo -i npm install -g @angular/cli

* it will ask if you want to share data with Angular team, just type “N” for no
* To play with angular.js cli
  + Type: ng version
  + Type:ng helpto see what kind of available commands there (it may not be possible to use them since you will need to have a project which is not important for our project).

1. To install express.js: Type command to make a directory: mkdir helloworld
2. Type command to change directory: cd helloworld
3. Type command to initializes your project which create a package.json: npm init
4. Just press keyboard ‘enter’ until you get to “**entry point: (index.js)**.” Write app.js to replace index.js.
5. Just keep pressing keyboard ‘enter’ until you get to when you need to write command
6. Type command to run express.js to save it on your project and directory: npm install express --save
7. Type command: nano app.js
8. I copied the code from the source by Rahic (more information is in the Reference). Copy the code below and paste it to the nano:

var express = require('express');

var app = express();

app.get('/', function (req, res) {

res.send('Hello World');

});

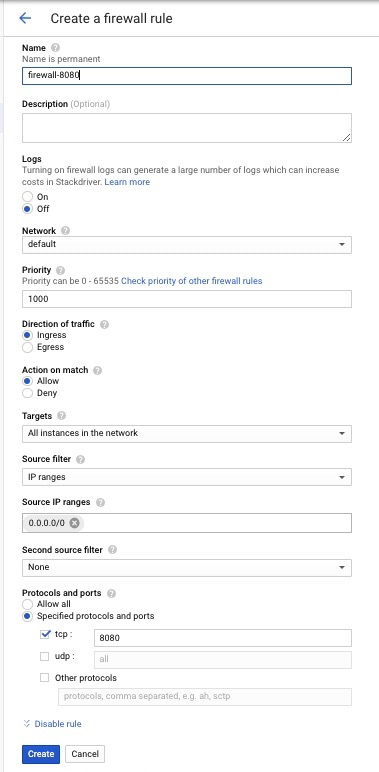
app.listen(8080, function () {

console.log('Example app listening on port 8080!');

});

* Press ctr+x to exit
* It will ask if you want to save so type ‘y’ then press keyboard ‘enter’

1. Go to console of google cloud platform. You will come back to shell soon.
2. Go to the top that shows **search symbol**
3. Type **firewall** and look for the one that states **Firewall rules**
4. There are two searches of ‘firewall rules,’ use one that is **VPC network** and click it
5. On the top, click **CREATE FIREWALL RULE**

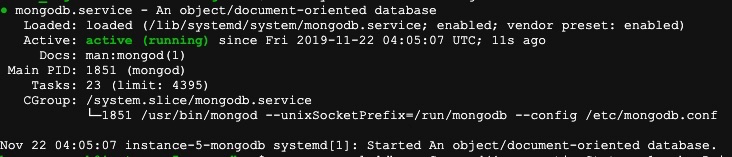


1. For **name**, type ‘firewall-8080’ (8080 is a port that you will need to open on a google cloud platform. The one you had to write on nano showed the port 8080).
2. Change **Targets** to **All instances in the network**
3. Type on **Source IP ranges**: 0.0.0.0/0
4. On **Protocols and ports**: click **tcp** and write 8080
5. Click **create**
6. Go to the nav to find **Compute Engine** and click **VM Instances**
7. Click the **External IP** of the instance’s name **test-mean-stack** which should show error
8. Now go back to shell
9. Type the command: node app.js

* It should show this: 

1. Go to the External IP tab and include http and 8080 to your numbers. http://00.00.000.000:8080. You already have your External IP numbers so just use http:// and 8080. Remember not to use https.
2. To leave the express, click the keyboard, ctr+c
3. Type the command to move up one directory: cd ..
4. To install mongodb (the database), type down: sudo apt install -y mongodb
5. To check the status of mongodb, type: sudo systemctl status mongodb

* You will see what shows active: active running which means the database server is running and that you are connected to mongodb.



1. Type: mongo --eval 'db.runCommand({ connectionStatus: 1, showPrivileges: true})'

* If you see the term “ok” : 1, that means your connection is working.

1. To start mongodb, type: sudo systemctl start mongodb
2. To work with database, type: mongo

Copied some code from Professor Conger’s [blog](https://congerprep.blogspot.com/2016/07/first-steps-with-nosql-mongo.html) to place some data into database

[

{\_id : **1**, product: "IPad", price : **894.50**, quantityAvailable : **13**},

{\_id : **2**, product: "Chrome Book", price : **245.99**, quantityAvailable: **23**},

{\_id : **3**, product: "Bose Lap Top Speakers", price : **89.50**, quantityAvailable : **10**},

{\_id : **4**, product: "Norton Anti Virus", price : **75.50**, quantityAvailable : **2**},

{\_id : **5**, product: "Mechanical Keyboard", price : **125.50**, quantityAvailable : **3**},

{\_id : **6**, product: "Android Tablet", price : **345.23**, quantityAvailable : **5**}

]

1. To exit mongodb, type: exit
2. To leave shell, type down: exit which should close it down
3. On the console of google cloud platform, make sure to click on **light-gray square** on the left of ‘test-mean-stack’ which should show blue check-mark
4. Go to top that shows **blue square**, click it to shut down the VM of ‘test-mean-stack’

You are done

**References:**

[“CLI Overview and Command Reference.” *ANGULAR*, 2019, angular.io/cli.](https://angular.io/cli)

[Dmytro, Melnyk. “Time to ‘Hello, World’: Running Node.js on VMs, Containers, Apps, and Functions - Part 2.” *Medium*, Node.js Collection, 25 May 2018, medium.com/the-node-js-collection/time-to-hello-world-part2-gce-95e9df907b93.](http://medium.com/the-node-js-collection/time-to-hello-world-part2-gce-95e9df907b93)

[“How To Install Node.js on Google Cloud with Ubuntu 18.04 and Nginx.” *Cloudbooklet*, 21 Sept. 2019,](https://www.cloudbooklet.com/how-to-install-node-js-on-google-cloud-with-ubuntu-18-04-and-nginx) [www.cloudbooklet.com/how-to-install-node-js-on-google-cloud-with-ubuntu-18-04-and-nginx/](http://www.cloudbooklet.com/how-to-install-node-js-on-google-cloud-with-ubuntu-18-04-and-nginx/)[.](https://www.cloudbooklet.com/how-to-install-node-js-on-google-cloud-with-ubuntu-18-04-and-nginx)

[“How to Open Port 8080 in Google Cloud Platform?” *YouTube*, YouTube, 16 July 2017, www.youtube.com/watch?v=JmjqPpQdtW8.](http://www.youtube.com/watch?v=JmjqPpQdtW8.)

[“Installing .” *Installing Express*, 2017, expressjs.com/en/starter/installing.html.](https://expressjs.com/en/starter/installing.html)

[“Install MongoDB on Ubuntu 18.04 LTS - Google Cloud.” *Cloudbooklet*, 21 Sept. 2019, www.cloudbooklet.com/install-mongodb-on-ubuntu-18-04-lts-google-cloud/.](https://www.cloudbooklet.com/install-mongodb-on-ubuntu-18-04-lts-google-cloud/)

[Kumar, Rahul. “How to Install Angular CLI on Debian 10/9/8.” *TecAdmin*, 30 July 2019, tecadmin.net/install-angular-on-debian/.](https://tecadmin.net/install-angular-on-debian/)

[Rahic, Adnan. “‘Hello World!" App with Node.js and Express.” *Medium*, 17 Jan. 2017, medium.com/@adnanrahic/hello-world-app-with-node-js-and-express-c1eb7cfa8a30.](http://medium.com/@adnanrahic/hello-world-app-with-node-js-and-express-c1eb7cfa8a30)