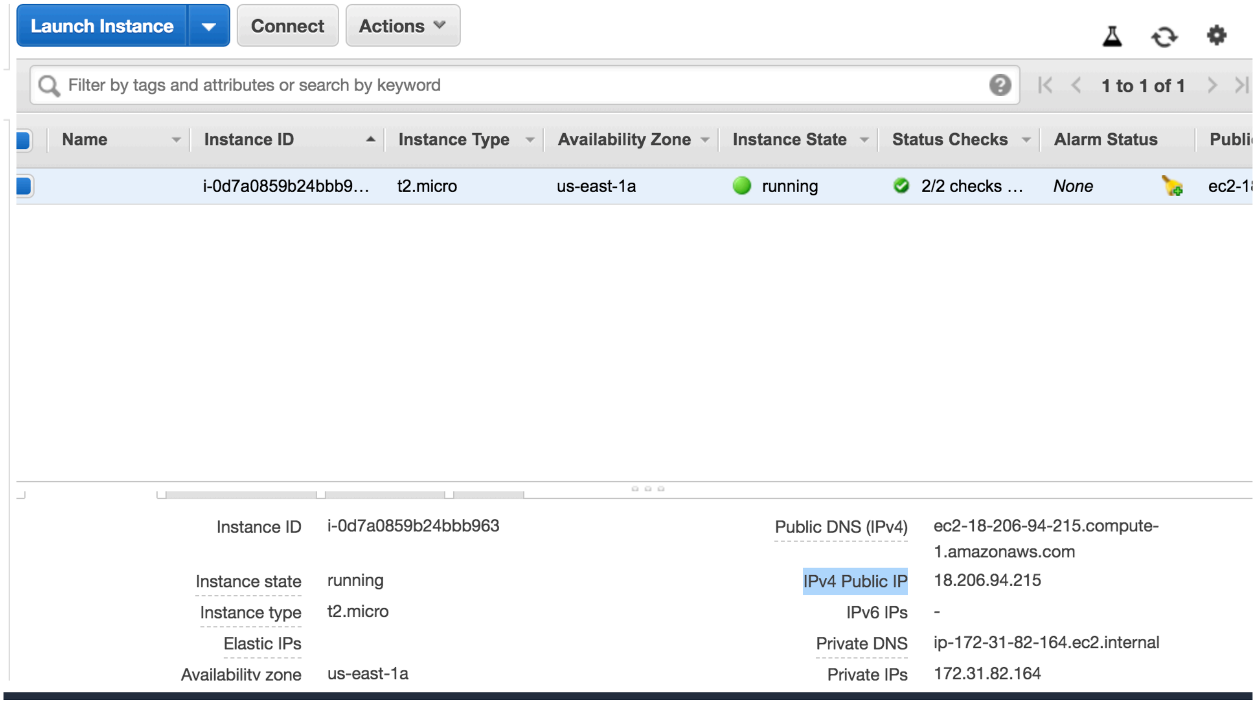
- setting up the server on an Amazon Elastic Compute Cloud (EC2) Ubuntu Server, the one I used is 64 bit image, Ubuntu.
- create a new ssh key pair, the one I created- AME394Fall2019.pem. Then you will get a downloaded file, which is your key.
- https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:sort=instanceId

- here you can see my key pair name and IPv4 Public IP address (blue color)
- Visit your EC2 Dashboard, and instance state should be "running" eventually, then you can find your public IP address.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Publi
	i-0d7a0859b24bbb9…	t2.micro	us-east-1a	● running	✓ 2/2 checks …	*None*	ec2-1

Instance ID	i-0d7a0859b24bbb963	Public DNS (IPv4)	ec2-18-206-94-215.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	18.206.94.215
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-82-164.ec2.internal
Availability zone	us-east-1a	Private IPs	172.31.82.164

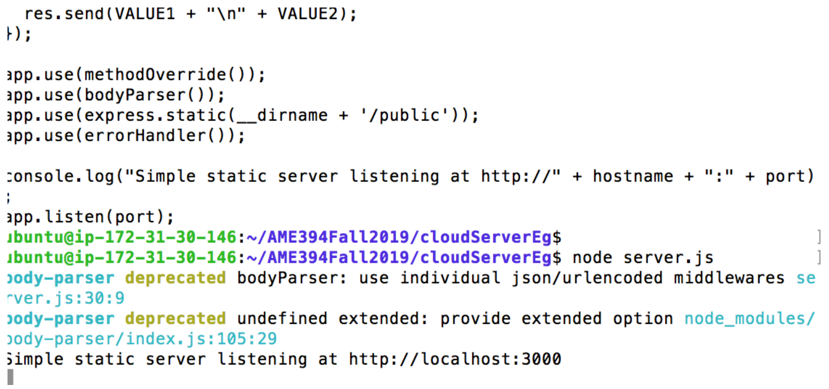
- Then, go to Terminal and following these steps below:

  1 cd Downloads/AME394Fall2019/cloudServerEg

  2 ssh ubuntu@18.206.94.215 (your IP address)

  3 then your machine is going to ask you "Are you sure you want to continue connecting (yes/no)? type "yes"

  4 connecting to your machine…



```
  res.send(VALUE1 + "\n" + VALUE2);
});

app.use(methodOverride());
app.use(bodyParser());
app.use(express.static(__dirname + '/public'));
app.use(errorHandler());

console.log("Simple static server listening at http://" + hostname + ":" + port)
;
app.listen(port);
ubuntu@ip-172-31-30-146:~/AME394Fall2019/cloudServerEg$
ubuntu@ip-172-31-30-146:~/AME394Fall2019/cloudServerEg$ node server.js
body-parser deprecated bodyParser: use individual json/urlencoded middlewares se
rver.js:30:9
body-parser deprecated undefined extended: provide extended option node_modules/
body-parser/index.js:105:29
Simple static server listening at http://localhost:3000
```

5. node server.js

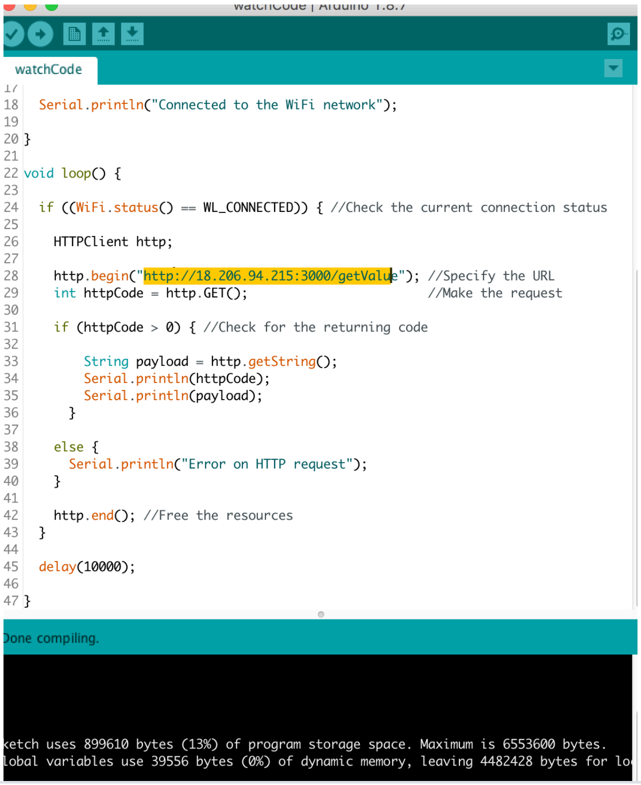
6, then server is receiving and shows your an address. http://localhose:3000

7. open the code called watchCode/Arduino 1.8.7

8. under the void loop -> change the IP address to your IP address. and then running it.

9. go to google chrome, type localhost:3000/index.html, you should see a simple html-based web. type something

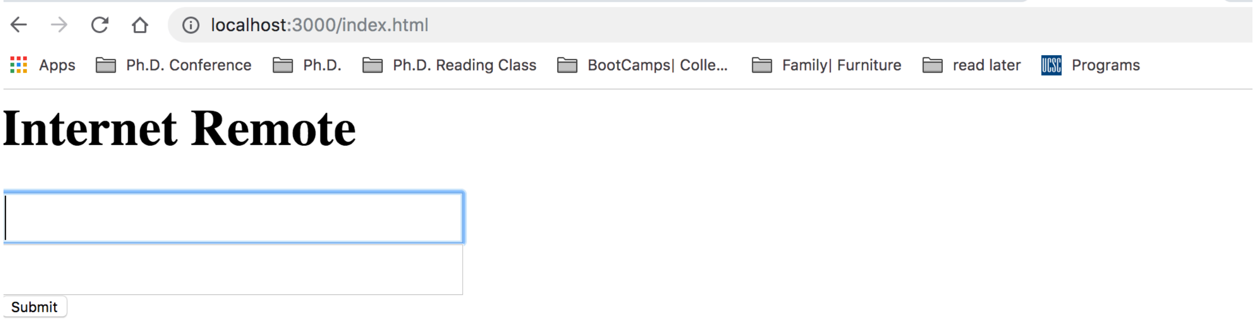
10, copy and past the yellow address(below) to a new webpage, you should see something that your typed on last step.



```
  Serial.println("Connected to the WiFi network");

}

void loop() {

  if ((WiFi.status() == WL_CONNECTED)) { //Check the current connection status

    HTTPClient http;

    http.begin("http://18.206.94.215:3000/getValue"); //Specify the URL
    int httpCode = http.GET();                        //Make the request

    if (httpCode > 0) { //Check for the returning code

        String payload = http.getString();
        Serial.println(httpCode);
        Serial.println(payload);
      }

    else {
      Serial.println("Error on HTTP request");
    }

    http.end(); //Free the resources
  }

  delay(10000);

}
```

```
one compiling.
```

```
ketch uses 899610 bytes (13%) of program storage space. Maximum is 6553600 bytes.
lobal variables use 39556 bytes (0%) of dynamic memory, leaving 4482428 bytes for lo
```



Internet Remote