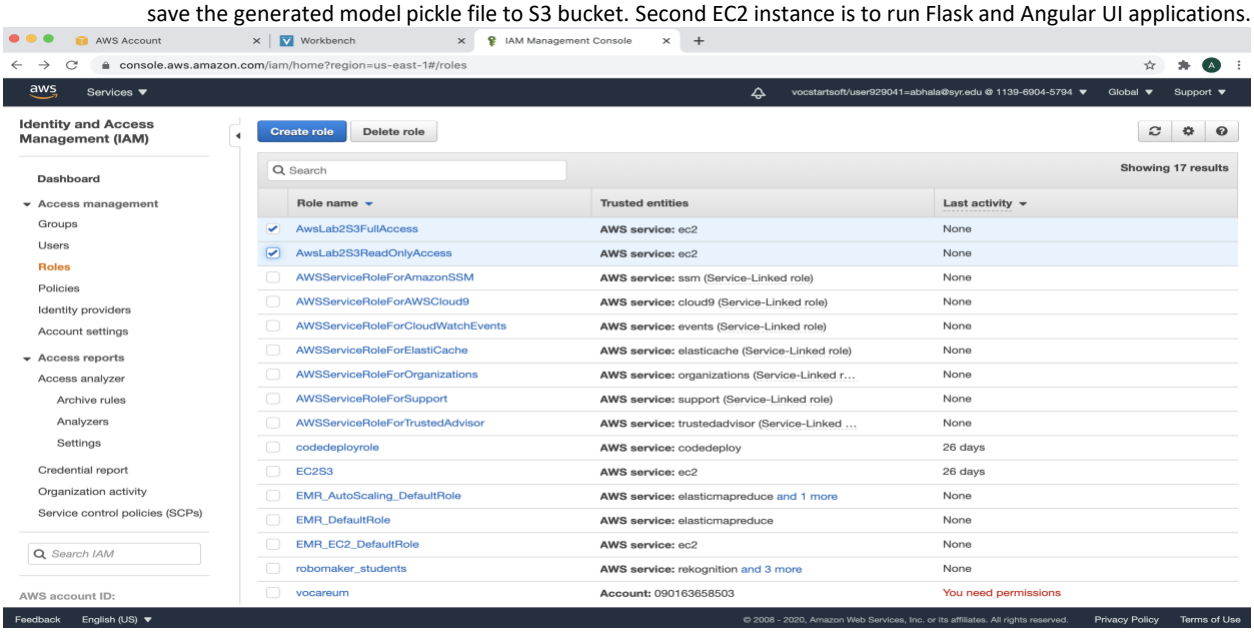
Section 1: Setting up our infrastructure

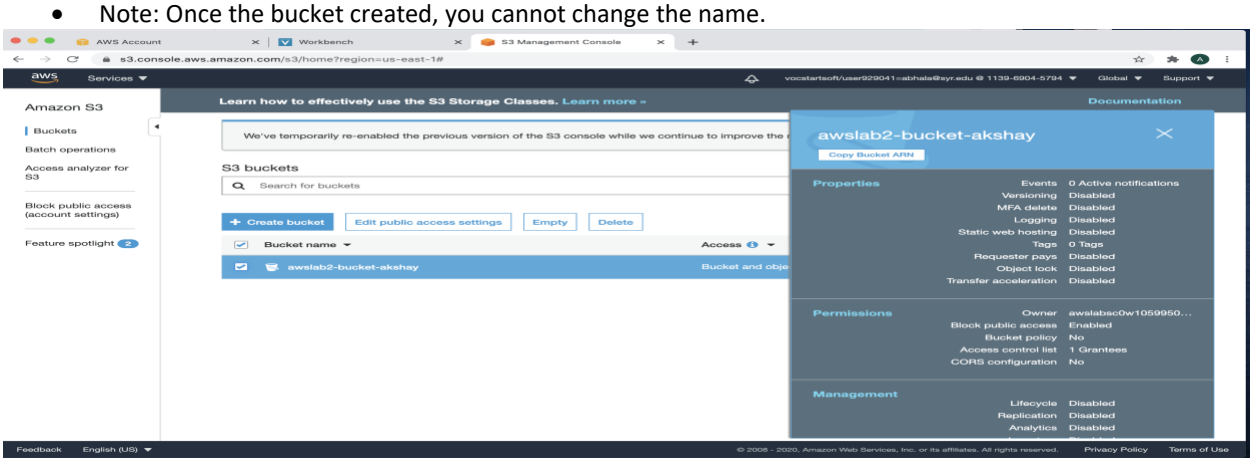
1. **Creating IAM roles**
   a. **First Ec2 instance – AmazonS3FullAcess**
   Purpose: Allow the first EC2 instance full access to our S3 (Simple Storage Service) bucket
   What does it mean: A policy that provides too much access and often granted when a service need to read and write files from S3 bucket.
   b. **Second Ec2 instance – AmazonS3ReadOnlyAccess**
   Purpose: Allow the first EC2 instance full access to our S3 (Simple Storage Service) bucket
   What does it mean: This allows everyone to read the objects in the bucket when we configure our bucket as a website.
   c. **Why two instances?**
   First EC2 instance will pull ML code and data from Github and run python code. We will use this instance to also save the generated model pickle file to S3 bucket. Second EC2 instance is to run Flask and Angular UI applications.



2. **Creating S3 Bucket**

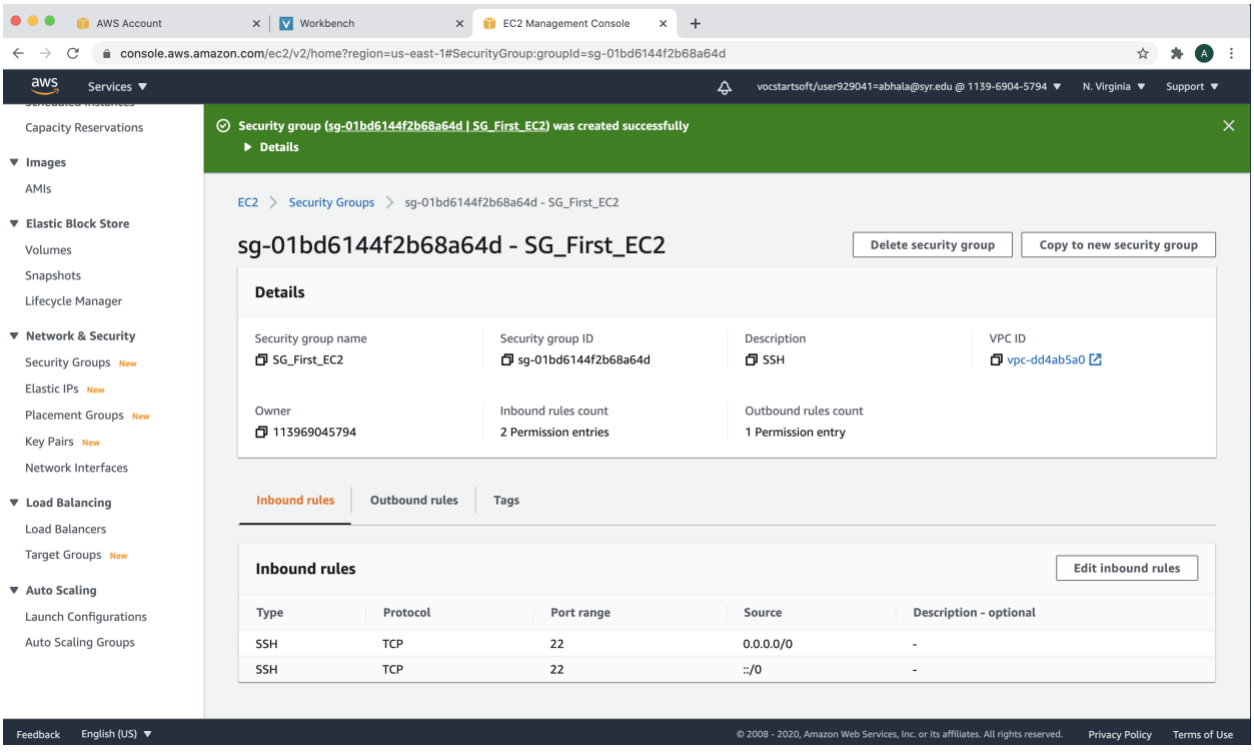
It is storage for the Internet. Amazon ML uses Amazon S3 as a primary data repository for the following tasks:

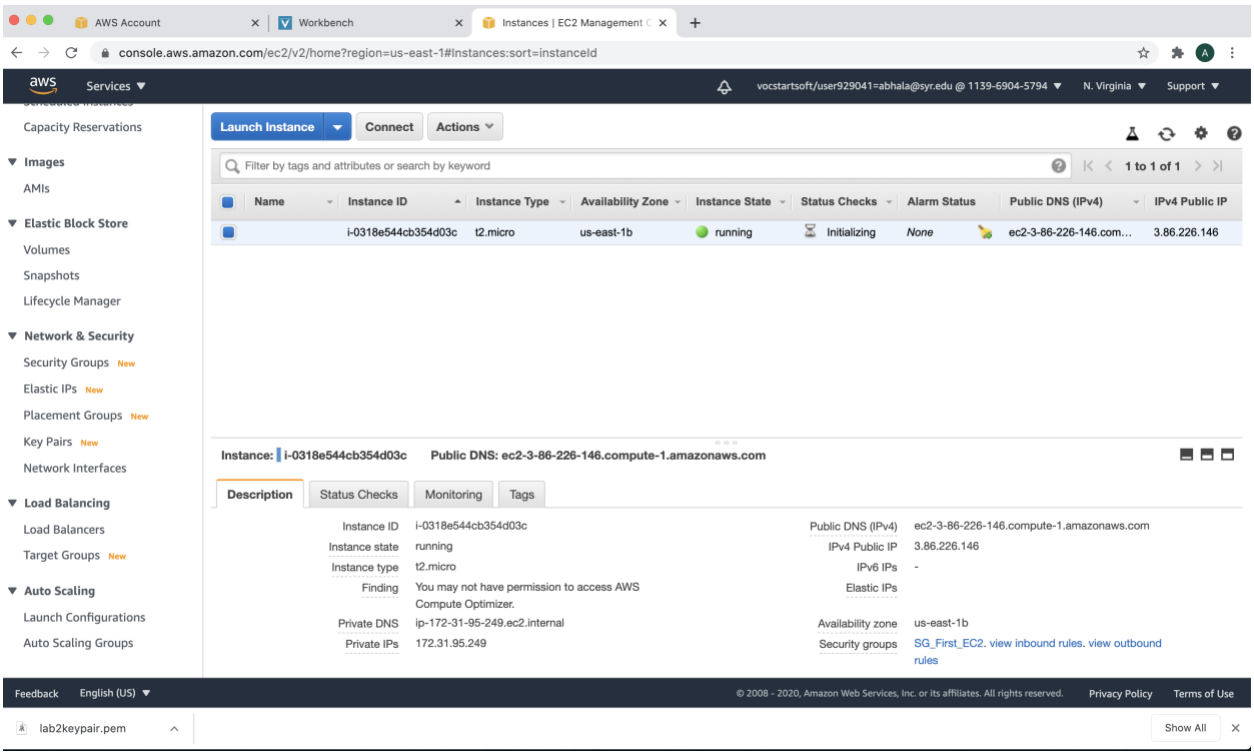
- To access your input files to create datasource objects for training and evaluating your ML models.
- To access your input files to generate batch predictions.
- When you generate batch predictions by using your ML models, to output the prediction file to an S3 bucket that you specify.
- Note: Once the bucket created, you cannot change the name.

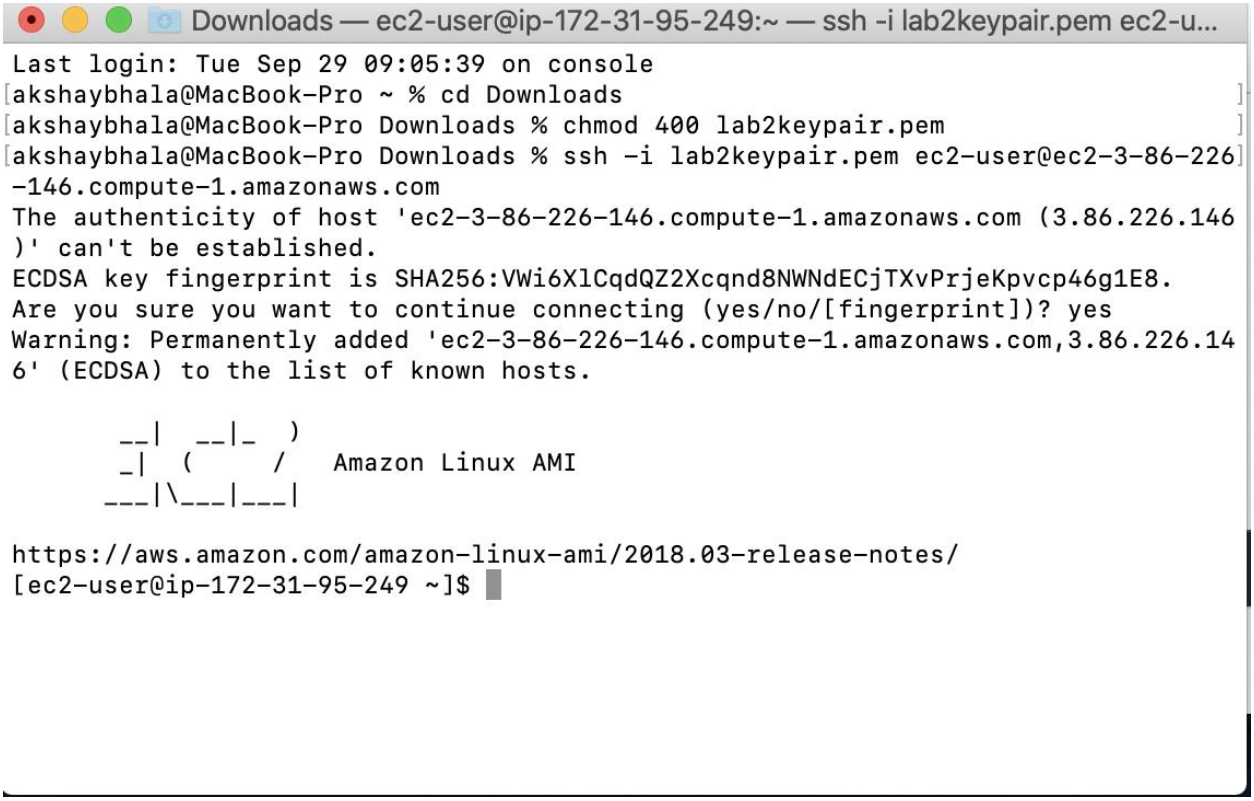


3. Creating first EC2 instance

- A security group acts as a virtual firewall for your EC2 instances to control incoming and outgoing traffic. Inbound rules control the incoming traffic to your instance, and outbound rules control the outgoing traffic from your instance.
- security groups use connection tracking to track information about traffic to and from the instance. Rules are applied based on the connection state of the traffic to determine if the traffic is allowed or denied.
- In first EC2 instance, TCP traffic on port 22 (SSH) to and from the instance is not tracked, because both the inbound and outbound rules allow all traffic (0.0.0.0/0)
- Note: An untracked flow of traffic is immediately interrupted if the rule that enables the flow is removed or modified. For example, if you have an open (0.0.0.0/0) outbound rule, and you remove a rule that allows all (0.0.0.0/0) inbound SSH (TCP port 22) traffic to the instance (or modify it such that the connection would no longer be permitted), your existing SSH connections to the instance are immediately dropped
- Attaching IAM role Awslab2S3Full Access to SG_First_EC2 is to write and run ML code.
- A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value, both of which you define. To add tags, I added the key as “ML-Model” and value as “ML_Model_EC2”.
- A key pair, consisting of a private key and a public key, is a set of security credentials that we can use to prove our identity when connecting to an instance. Amazon EC2 stores the public key, and you store the private key. “lab2keypair”
- After that, SSH into EC2 instance using public DNS or IPv4 public IP. Download all the required packages, executed the following commands on ec2.




AWS Account
Workbench
Instances | EC2 Management
console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:sort=instanceId
Services
vocstartsoft/user929041=abhala@syr.edu @ 1139-6904-5794
N. Virginia
Support
Launch Instance
Connect
Actions
Filter by tags and attributes or search by keyword
1 to 1 of 1
Name
Instance ID
Instance Type
Availability Zone
Instance State
Status Checks
Alarm Status
Public DNS (IPv4)
IPv4 Public IP
i-0318e544cb354d03c
t2.micro
us-east-1b
running
Initializing
None
ec2-3-86-226-146.com...
3.86.226.146
Capacity Reservations
Images
AMIs
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Target Groups
Auto Scaling
Launch Configurations
Auto Scaling Groups
Instance: i-0318e544cb354d03c
Public DNS: ec2-3-86-226-146.compute-1.amazonaws.com
Description
Status Checks
Monitoring
Tags
Instance ID i-0318e544cb354d03c
Instance state running
Instance type t2.micro
Finding You may not have permission to access AWS Compute Optimizer.
Private DNS ip-172-31-95-249.ec2.internal
Private IPs 172.31.95.249
Public DNS (IPv4) ec2-3-86-226-146.compute-1.amazonaws.com
IPv4 Public IP 3.86.226.146
IPv6 IPs -
Elastic IPs
Availability zone us-east-1b
Security groups SG_First_EC2. view inbound rules. view outbound rules
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
lab2keypair.pem
Show All

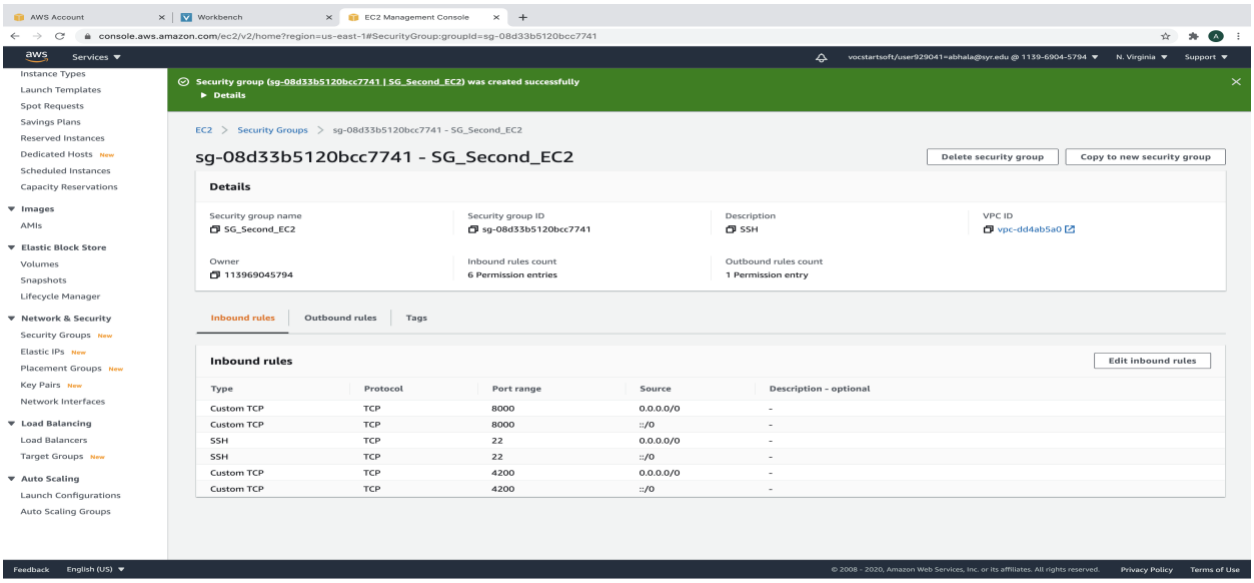

Downloads — ec2-user@ip-172-31-95-249:~ — ssh -i lab2keypair.pem ec2-u...
Last login: Tue Sep 29 09:05:39 on console
akshaybhala@MacBook-Pro ~ % cd Downloads
akshaybhala@MacBook-Pro Downloads % chmod 400 lab2keypair.pem
akshaybhala@MacBook-Pro Downloads % ssh -i lab2keypair.pem ec2-user@ec2-3-86-226-146.compute-1.amazonaws.com
The authenticity of host 'ec2-3-86-226-146.compute-1.amazonaws.com (3.86.226.146)' can't be established.
ECDSA key fingerprint is SHA256:VWi6XlCqdQZ2Xcqnd8NWNdECjTXvPrjeKpvcp46g1E8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-86-226-146.compute-1.amazonaws.com,3.86.226.146' (ECDSA) to the list of known hosts.
__| __|_)
_| (/ Amazon Linux AMI
___|___|___|
https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
[ec2-user@ip-172-31-95-249 ~]$

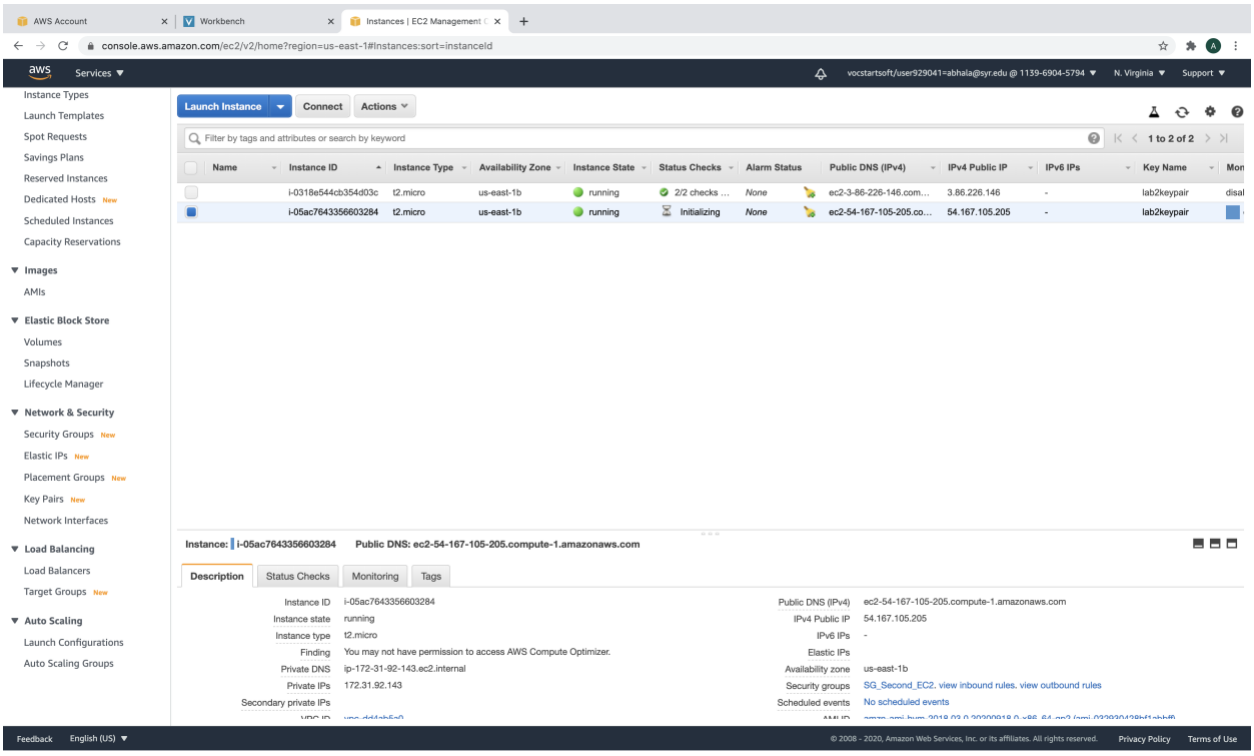
- Our first Instance is ready to run ML code

```
---> Package git-core-doc.noarch 0:2.18.4-2.71.amzn1 will be installed
---> Package perl-Git.noarch 0:2.18.4-2.71.amzn1 will be installed
--> Processing Dependency: perl(Error) for package: perl-Git-2.18.4-2.71.amzn1.noarch
---> Package perl-TermReadKey.x86_64 0:2.30-20.9.amzn1 will be installed
---> Package python26.x86_64 0:2.6.9-2.92.amzn1 will be installed
--> Processing Dependency: libpython2.6.so.1.0()(64bit) for package: python26-2.6.9-2.92.amzn1.x86_64
--> Running transaction check
---> Package perl-Error.noarch 1:0.17020-2.9.amzn1 will be installed
---> Package python26-libs.x86_64 0:2.6.9-2.92.amzn1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package              Arch       Version                  Repository        Size
================================================================================
Installing:
 git                  x86_64     2.18.4-2.71.amzn1        amzn-updates     183 k
Installing for dependencies:
 git-core             x86_64     2.18.4-2.71.amzn1        amzn-updates      10 M
 git-core-doc         noarch     2.18.4-2.71.amzn1        amzn-updates     3.1 M
 perl-Error           noarch     1:0.17020-2.9.amzn1      amzn-main         33 k
 perl-Git             noarch     2.18.4-2.71.amzn1        amzn-updates      77 k
 perl-TermReadKey     x86_64     2.30-20.9.amzn1          amzn-main         33 k
 python26             x86_64     2.6.9-2.92.amzn1         amzn-updates     5.8 M
 python26-libs        x86_64     2.6.9-2.92.amzn1         amzn-updates     697 k

Transaction Summary
================================================================================
Install  1 Package (+7 Dependent packages)

Total download size: 20 M
Installed size: 55 M
Downloading packages:
(1/8): perl-Error-0.17020-2.9.amzn1.noarch.rpm             |  33 kB   00:00
(2/8): perl-TermReadKey-2.30-20.9.amzn1.x86_64.rpm         |  33 kB   00:00
(3/8): git-2.18.4-2.71.amzn1.x86_64.rpm                    | 183 kB   00:00
(4/8): perl-Git-2.18.4-2.71.amzn1.noarch.rpm               |  77 kB   00:00
(5/8): git-core-doc-2.18.4-2.71.amzn1.noarch.rpm           | 3.1 MB   00:00
(6/8): python26-libs-2.6.9-2.92.amzn1.x86_64.rpm           | 697 kB   00:00
(7/8): python26-2.6.9-2.92.amzn1.x86_64.rpm                | 5.8 MB   00:00
(8/8): git-core-2.18.4-2.71.amzn1.x86_64.rpm               |  10 MB   00:01
--------------------------------------------------------------------------------
Total                                           10 MB/s |  20 MB  00:01
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : python26-libs-2.6.9-2.92.amzn1.x86_64                       1/8
  Installing : python26-2.6.9-2.92.amzn1.x86_64                            2/8
  Installing : git-core-2.18.4-2.71.amzn1.x86_64                           3/8
  Installing : perl-TermReadKey-2.30-20.9.amzn1.x86_64                     4/8
  Installing : 1:perl-Error-0.17020-2.9.amzn1.noarch                       5/8
  Installing : git-core-doc-2.18.4-2.71.amzn1.noarch                       6/8
  Installing : git-2.18.4-2.71.amzn1.x86_64                                7/8
  Installing : perl-Git-2.18.4-2.71.amzn1.noarch                           8/8
  Verifying  : 1:perl-Error-0.17020-2.9.amzn1.noarch                       1/8
  Verifying  : perl-Git-2.18.4-2.71.amzn1.noarch                           2/8
  Verifying  : perl-TermReadKey-2.30-20.9.amzn1.x86_64                     3/8
  Verifying  : python26-2.6.9-2.92.amzn1.x86_64                            4/8
  Verifying  : git-core-2.18.4-2.71.amzn1.x86_64                           5/8
  Verifying  : git-core-doc-2.18.4-2.71.amzn1.noarch                       6/8
  Verifying  : git-2.18.4-2.71.amzn1.x86_64                                7/8
  Verifying  : python26-libs-2.6.9-2.92.amzn1.x86_64                       8/8

Installed:
  git.x86_64 0:2.18.4-2.71.amzn1

Dependency Installed:
  git-core.x86_64 0:2.18.4-2.71.amzn1
  git-core-doc.noarch 0:2.18.4-2.71.amzn1
  perl-Error.noarch 1:0.17020-2.9.amzn1
  perl-Git.noarch 0:2.18.4-2.71.amzn1
  perl-TermReadKey.x86_64 0:2.30-20.9.amzn1
  python26.x86_64 0:2.6.9-2.92.amzn1
  python26-libs.x86_64 0:2.6.9-2.92.amzn1

Complete!
[ec2-user@ip-172-31-95-249 ~]$
```

4. Creating Second EC2 instance

- Similarly, in second EC2 instance, TCP traffic on port 22 (SSH) and port 8000 (Flask API)to and from the instance is not tracked, because both the inbound and outbound rules allow all traffic (0.0.0.0/0)
- Attaching IAM role Awslab2S3ReadOnlyAccess to SG_Second_EC2 is to allow everyone to only read the objects in the bucket / output in our App.
- I added the key as “”Flask_and_UI”. Used same key pair which was generated in First EC2 instance.
- After that, SSH into EC2 instance using public DNS or IPv4 public IP. Download all the required packages for Flask and Angular UI, executed the following commands on ec2.





```
akshaybhala@MacBook-Pro Downloads % chmod 400 lab2keypair.pem
akshaybhala@MacBook-Pro Downloads % ssh -i lab2keypair.pem ec2-user@ec2-54-167-105-205.compute-1.amazonaws.com
The authenticity of host 'ec2-54-167-105-205.compute-1.amazonaws.com (54.167.105.205)' can't be established.
ECDSA key fingerprint is SHA256:OVb5AUL2L6IqdoHbOrMEC38BfCYdpA7W1sp42M+4DRo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-167-105-205.compute-1.amazonaws.com,54.167.105.205' (ECDSA) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
[ec2-user@ip-172-31-92-143 ~]$
```

- Our Second is instance is ready for hosting Flask API and Angular AI

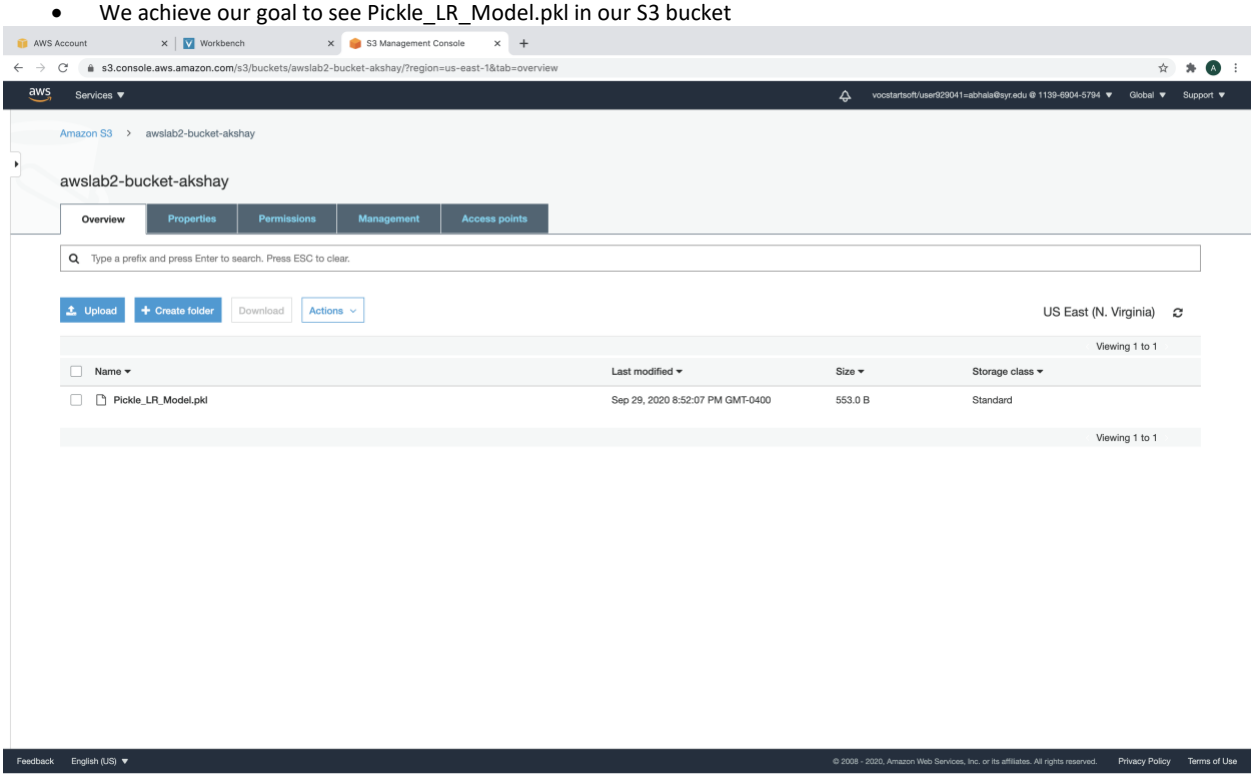
```
Install  1 Package (+4 Dependent packages)

Total download size: 1.5 M
Installed size: 3.6 M
Downloading packages:
(1/5): apr-util-ldap-1.5.4-6.18.amzn1.x86_64.rpm           |  19 kB   00:00
(2/5): apr-util-1.5.4-6.18.amzn1.x86_64.rpm                |  99 kB   00:00
(3/5): httpd-tools-2.2.34-1.16.amzn1.x86_64.rpm            |  80 kB   00:00
(4/5): apr-1.5.2-5.13.amzn1.x86_64.rpm                     | 118 kB   00:00
(5/5): httpd-2.2.34-1.16.amzn1.x86_64.rpm                  | 1.2 MB   00:00
--------------------------------------------------------------------------------
Total                                           1.8 MB/s | 1.5 MB  00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : apr-1.5.2-5.13.amzn1.x86_64                                  1/5
  Installing : apr-util-1.5.4-6.18.amzn1.x86_64                             2/5
  Installing : httpd-tools-2.2.34-1.16.amzn1.x86_64                         3/5
  Installing : apr-util-ldap-1.5.4-6.18.amzn1.x86_64                        4/5
  Installing : httpd-2.2.34-1.16.amzn1.x86_64                               5/5
  Verifying  : httpd-tools-2.2.34-1.16.amzn1.x86_64                         1/5
  Verifying  : apr-util-1.5.4-6.18.amzn1.x86_64                             2/5
  Verifying  : httpd-2.2.34-1.16.amzn1.x86_64                               3/5
  Verifying  : apr-1.5.2-5.13.amzn1.x86_64                                  4/5
  Verifying  : apr-util-ldap-1.5.4-6.18.amzn1.x86_64                        5/5

Installed:
  httpd.x86_64 0:2.2.34-1.16.amzn1

Dependency Installed:
  apr.x86_64 0:1.5.2-5.13.amzn1
  apr-util.x86_64 0:1.5.4-6.18.amzn1
  apr-util-ldap.x86_64 0:1.5.4-6.18.amzn1
  httpd-tools.x86_64 0:2.2.34-1.16.amzn1

Complete!
[ec2-user@ip-172-31-92-143 ~]$ sudo service httpd start
Starting httpd:                                            [  OK  ]
[ec2-user@ip-172-31-92-143 ~]$ npm install -g @angular/cl
npm ERR! code E404
npm ERR! 404 Not Found - GET https://registry.npmjs.org/@angular%2fcl - Not found
npm ERR! 404
npm ERR! 404  '@angular/cl@latest' is not in the npm registry.
npm ERR! 404 You should bug the author to publish it (or use the name yourself!)
npm ERR! 404
npm ERR! 404 Note that you can also install from a
npm ERR! 404 tarball, folder, http url, or git url.

npm ERR! A complete log of this run can be found in:
npm ERR!     /home/ec2-user/.npm/_logs/2020-09-30T00_47_00_814Z-debug.log
[ec2-user@ip-172-31-92-143 ~]$ npm install -g @angular/cli
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
/home/ec2-user/.nvm/versions/node/v14.13.0/bin/ng -> /home/ec2-user/.nvm/versions/node/v14.13.0/lib/node_modules/@angular/cli/bin/ng

> @angular/cli@10.1.3 postinstall /home/ec2-user/.nvm/versions/node/v14.13.0/lib/node_modules/@angular/cli
> node ./bin/postinstall/script.js

? Would you like to share anonymous usage data with the Angular Team at Google u
nder
Google's Privacy Policy at https://policies.google.com/privacy? For more details
 and
how to change this setting, see http://angular.io/analytics. No
+ @angular/cli@10.1.3
added 277 packages from 207 contributors in 16.966s
[ec2-user@ip-172-31-92-143 ~]$ npm install
npm WARN saveError ENOENT: no such file or directory, open '/home/ec2-user/package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open '/home/ec2-user/package.json'
npm WARN ec2-user No description
npm WARN ec2-user No repository field.
npm WARN ec2-user No README data
npm WARN ec2-user No license field.

up to date in 0.227s
found 0 vulnerabilities

[ec2-user@ip-172-31-92-143 ~]$
```

Section 2: Run Flask and Angular Code

- **Run ML Code:**
- Successful in establishing connection to Ec2 instance through SSH
- Cloning the git repository using "git clone command" which contains AWSLab2MLcode
- After cloning we run BostonHousingLR.py file on our Ec2 instance which gives us the results stored in the form of a pickle file (A PKL file is a file created by pickle, a Python module that enables objects to be serialized to files on disk and deserialized back into the program at runtime)
- Moving ahead, we then copy our pickle file to a S3 bucket which we created (bucket-akshay) using the command shown below in red.

```
akshaybhala@MacBook-Pro Downloads % chmod 400 lab2keypair.pem
akshaybhala@MacBook-Pro Downloads % ssh -i lab2keypair.pem ec2-user@ec2-3-86-226-146.compute-1.amazonaws.com
Last login: Wed Sep 30 00:35:34 2020 from cpe-67-244-154-132.rochester.res.rr.com

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
[ec2-user@ip-172-31-95-249 ~]$
[ec2-user@ip-172-31-95-249 ~]$ git clone https://github.com/ssingh60/AWSLab2MLCode.git master
Cloning into 'master'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 11 (delta 3), reused 2 (delta 0), pack-reused 0
Unpacking objects: 100% (11/11), done.
[ec2-user@ip-172-31-95-249 ~]$ cd master
[ec2-user@ip-172-31-95-249 master]$ python3 BostonHousingLR.py
(404, 2)
(102, 2)
(404, 1)
(102, 1)
The model performance for training set
--------------------------------------
RMSE is 5.137400784702911
[ec2-user@ip-172-31-95-249 master]$ ll
total 48
-rw-rw-r-- 1 ec2-user ec2-user 35734 Sep 30 00:49 BostonHousing.csv
-rw-rw-r-- 1 ec2-user ec2-user   979 Sep 30 00:49 BostonHousingLR.py
-rw-rw-r-- 1 ec2-user ec2-user   553 Sep 30 00:49 Pickle_LR_Model.pkl
-rw-rw-r-- 1 ec2-user ec2-user    15 Sep 30 00:49 README.md
[ec2-user@ip-172-31-95-249 master]$

[ec2-user@ip-172-31-95-249 master]$ aws s3 cp Pickle_LR_Model.pkl s3://awslab2-bucket-akshay
upload: ./Pickle_LR_Model.pkl to s3://awslab2-bucket-akshay/Pickle_LR_Model.pkl
[ec2-user@ip-172-31-95-249 master]$
```

- We achieve our goal to see Pickle_LR_Model.pkl in our S3 bucket



2. Run Flask and Angular Code:

- Successful in establishing connection to second Ec2 instance through SSH
- Cloning the git repository using "git clone command" which contains AWSLab2FlaskCode
- Copying our pickle file stored in our bucket to our Second EC2 instance so that our flask service keeps on running in background. The command to run the service is shown below in red. (Nohup: short for no hang up is a command in Linux systems that keep processes running even after exiting the shell or terminal)
- This service accepts input request from UI and gives back the output predicted housing price.
-

```
[ec2-user@ip-172-31-92-143 AWSLab2FlaskCode]$ aws s3 cp s3://awslab2-bucket-akshay/Pickle_LR_Model.pkl ./
download: s3://awslab2-bucket-akshay/Pickle_LR_Model.pkl to ./Pickle_LR_Model.pkl
[ec2-user@ip-172-31-92-143 AWSLab2FlaskCode]$
[ec2-user@ip-172-31-92-143 AWSLab2FlaskCode]$ nohup python3 modelFlask.py &
[1] 4517
[ec2-user@ip-172-31-92-143 AWSLab2FlaskCode]$ nohup: ignoring input and appending output to 'nohup.out'
[ec2-user@ip-172-31-92-143 AWSLab2FlaskCode]$
```

3. Run Angular Code:

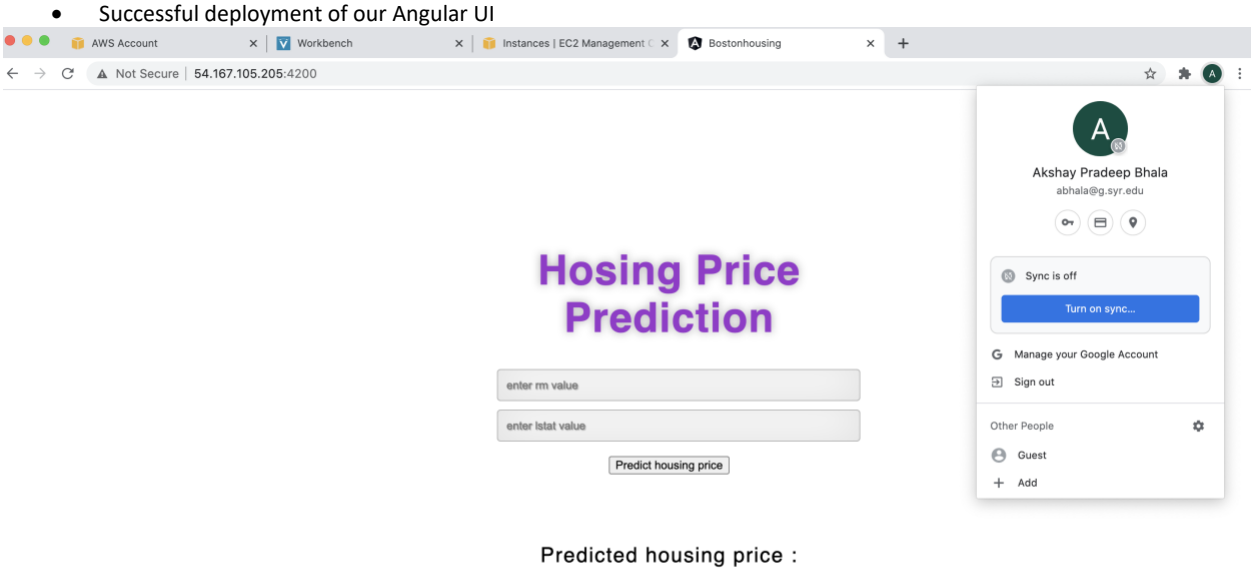
- Cloning the git repository using “git clone command” which contains AWSLab2AngularUI.git
- Our purpose is to deploy our Angular UI code for our application on EC2.
- As our Flask Service is running in background our Angular code will call Flask API for output but before that we need to change the IPV4 address present in web.service.ts file to the IPV4 address of our running instance so that we can call Flask API which is deployed on our EC2 machine.
- After building a connection between our Flask API and Angular UI we now install the dependencies in Node_modules folder of Node Package Manager using “npm install command” (It installs the packages you want to use and provides a useful interface to work with them)

```
[ec2-user@ip-172-31-92-143 AWSLab2FlaskCode]$ nohup: ignoring input and appending output to 'nohup.out'
[ec2-user@ip-172-31-92-143 AWSLab2FlaskCode]$ git clone https://github.com/ssingh60/AWSLab2AngularUI.git
Cloning into 'AWSLab2AngularUI'...
remote: Enumerating objects: 47, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (44/44), done.
remote: Total 47 (delta 5), reused 38 (delta 1), pack-reused 0
Unpacking objects: 100% (47/47), done.
[ec2-user@ip-172-31-92-143 AWSLab2FlaskCode]$ cd AWSLab2AngularUI
[ec2-user@ip-172-31-92-143 AWSLab2AngularUI]$ visrc/app/web.service.ts
-bash: visrc/app/web.service.ts: No such file or directory
[ec2-user@ip-172-31-92-143 AWSLab2AngularUI]$ vi src/app/web.service.ts
[ec2-user@ip-172-31-92-143 AWSLab2AngularUI]$
```

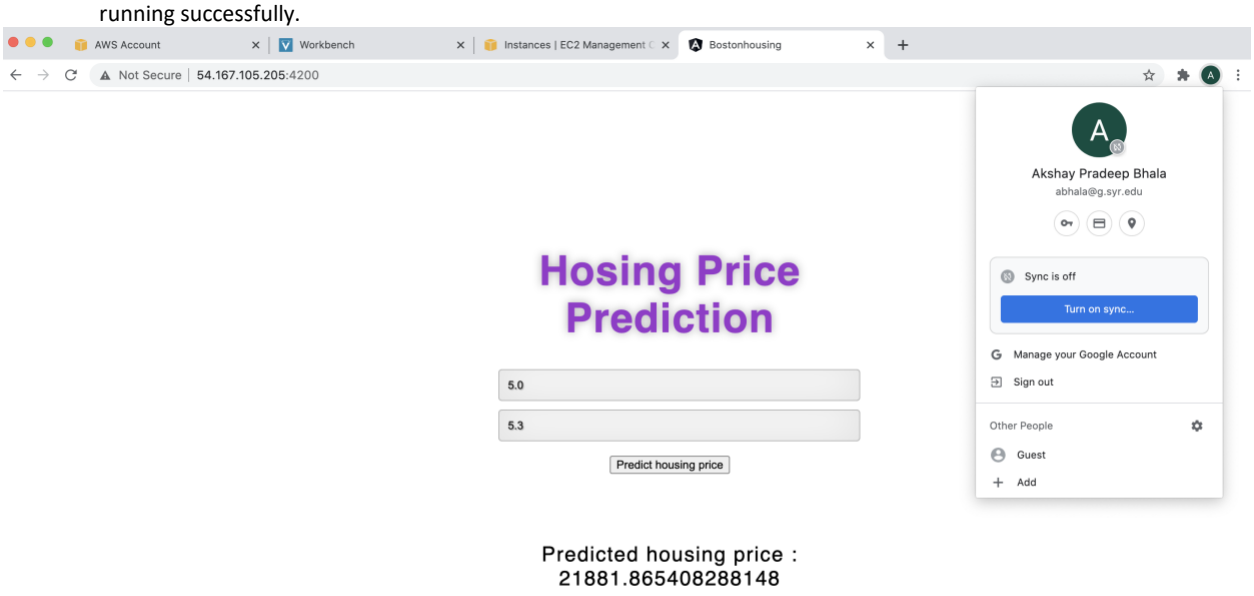
- After installing all the packages required for Interface, we finally host our application using command ng serve –host 0.0.0.0 – port 4200 (Builds and serves our app, rebuilding on file changes and host to listen on)

```
[ec2-user@ip-172-31-92-143 AWSLab2AngularUI]$ ng serve --host 0.0.0.0 --port 4200
Your global Angular CLI version (10.1.3) is greater than your local
version (10.0.5). The local Angular CLI version is used.

To disable this warning use "ng config -g cli.warnings.versionMismatch false".
WARNING: This is a simple server for use in testing or debugging Angular applications
locally. It hasn't been reviewed for security issues.

Binding this server to an open connection can result in compromising your application or
computer. Using a different host than the one passed to the "--host" flag might result in
websocket connection issues. You might need to use "--disableHostCheck" if that's the
case.
Compiling @angular/animations : es2015 as esm2015
Compiling @angular/core : es2015 as esm2015
Compiling @angular/animations/browser : es2015 as esm2015
Compiling @angular/animations/browser/testing : es2015 as esm2015
Compiling @angular/common : es2015 as esm2015
Compiling @angular/common/http : es2015 as esm2015
Compiling @angular/common/http/testing : es2015 as esm2015
Compiling @angular/forms : es2015 as esm2015
Compiling @angular/platform-browser : es2015 as esm2015
Compiling @angular/platform-browser/animations : es2015 as esm2015
Compiling @angular/core/testing : es2015 as esm2015
Compiling @angular/platform-browser-dynamic : es2015 as esm2015
Compiling @angular/platform-browser/testing : es2015 as esm2015
Compiling @angular/compiler/testing : es2015 as esm2015
Compiling @angular/platform-browser-dynamic/testing : es2015 as esm2015
Compiling @angular/common/testing : es2015 as esm2015
Compiling @angular/router : es2015 as esm2015
Compiling @angular/router/testing : es2015 as esm2015

chunk {main} main.js, main.js.map (main) 23.4 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 141 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.15 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 12.4 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 3.02 MB [initial] [rendered]
Date: 2020-09-30T01:03:54.014Z - Hash: 0735ae03fa8d281a9193 - Time: 14988ms
** Angular Live Development Server is listening on 0.0.0.0:4200, open your browser on http://localhost:4200/ **
: Compiled successfully.
```

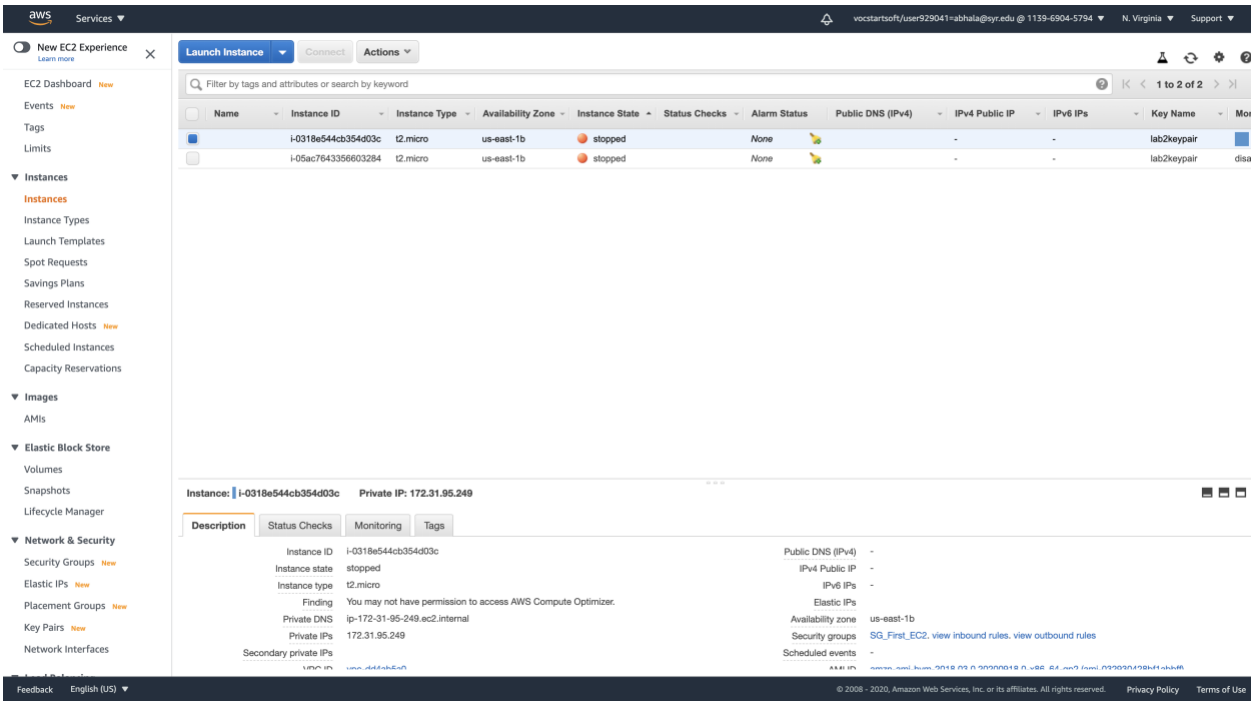
- Using Ipv4 address of EC2 with port 4200 and paste on browser to see our angular UI.
- Successful deployment of our Angular UI



- After giving input we get the predicted housing price which proves our ML model deployment and flask services are running successfully.



- Terminated instances



Citations:
https://docs.aws.amazon.com/machine-learning/latest/dg/tutorial.html