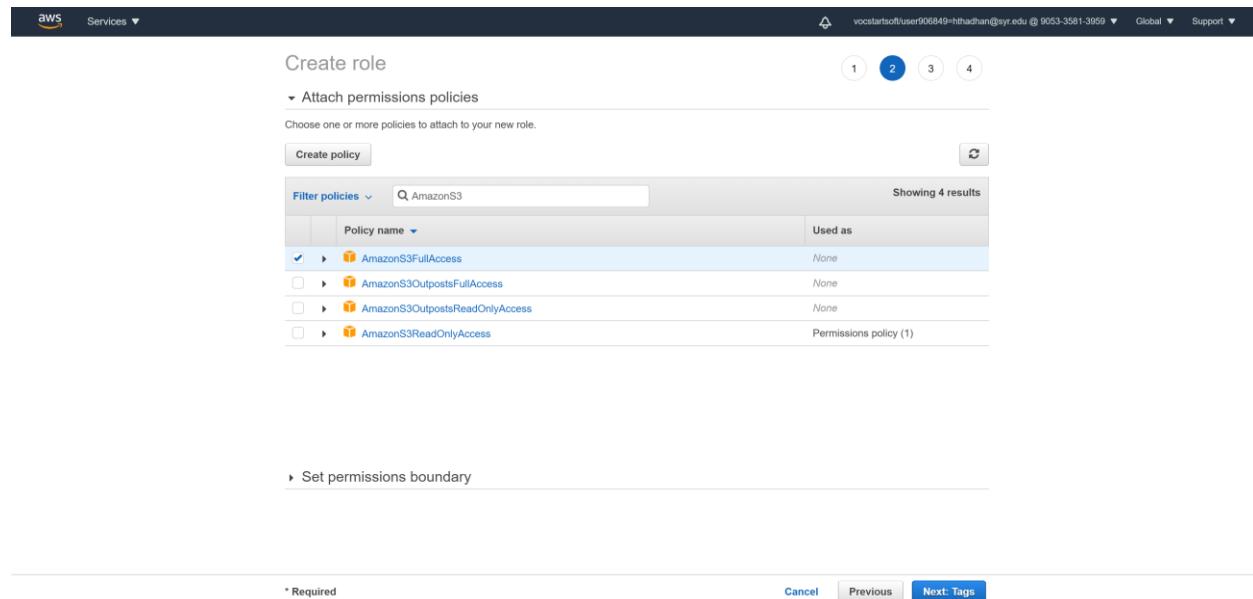


Section 2.1. Creating IAM Roles:

We are creating 2 IAM roles. IAM roles are used to provide specific permissions for access. These help in controlling the access to the AWS resources like EC2 and S3 with only the required permissions so that one cannot do anything or access unwanted information. AWS Identity and Access Management Roles (IAM) Roles set specific permissions for access control. They define the granularity level as to which the access control is provided like Read Only Access or Full Access etc.

One Role is created for the 1st EC2 instance with policy of “AmazonS3FullAccess” to allow the EC2 instance to have full access of the S3 bucket. This is due to the fact that 1st EC2 instance is running all the Machine Learning code to generate the pickle file as the model output which is copied to the S3 bucket and only this 1st EC2 where the ML code is running should be able to modify the model output so it is given Full Access to the S3. 2nd Role for the 2nd EC2 instance is created with policy of “AmazonS3ReadOnlyAccess” so that the 2nd EC2 instance where we have our Flask and Angular API code running need not modify the contents of the ML model which is not required by this 2nd EC2 instance only needs to access this S3 bucket without modifying it.



Create role

Review

Provide the required information below and review this role before you create it.

Role name*	AwsLabS3FullAccess
Use alphanumeric and '-' characters. Maximum 64 characters.	
Role description	S3 given full access through this role
Maximum 1000 characters. Use alphanumeric and '-' characters.	
Trusted entities	AWS service: ec2.amazonaws.com
Policies	AmazonS3FullAccess Edit
Permissions boundary	Permissions boundary is not set
No tags were added.	

* Required

[Cancel](#) [Previous](#) [Create role](#)

Create role

Attach permissions policies

Choose one or more policies to attach to your new role.

Create policy		Showing 4 results
Filter policies		AmazonS3
Policy name		Used as
<input type="checkbox"/>	AmazonS3FullAccess	Permissions policy (1)
<input type="checkbox"/>	AmazonS3OutpostsFullAccess	None
<input type="checkbox"/>	AmazonS3OutpostsReadOnlyAccess	None
<input checked="" type="checkbox"/>	AmazonS3ReadOnlyAccess	Permissions policy (1)

▶ Set permissions boundary

* Required

[Cancel](#) [Previous](#) [Next: Tags](#)

Create role

Review

Provide the required information below and review this role before you create it.

Role name* AwsLab2S3ReadOnlyAccess

Use alphanumeric and "+", "-", "_" characters. Maximum 64 characters.

Role description Read Only access to EC2 for S3 for the 2nd instance

Maximum 1000 characters. Use alphanumeric and "+", "-", "_" characters.

Trusted entities AWS service: ec2.amazonaws.com

Policies AmazonS3ReadOnlyAccess

Permissions boundary Permissions boundary is not set

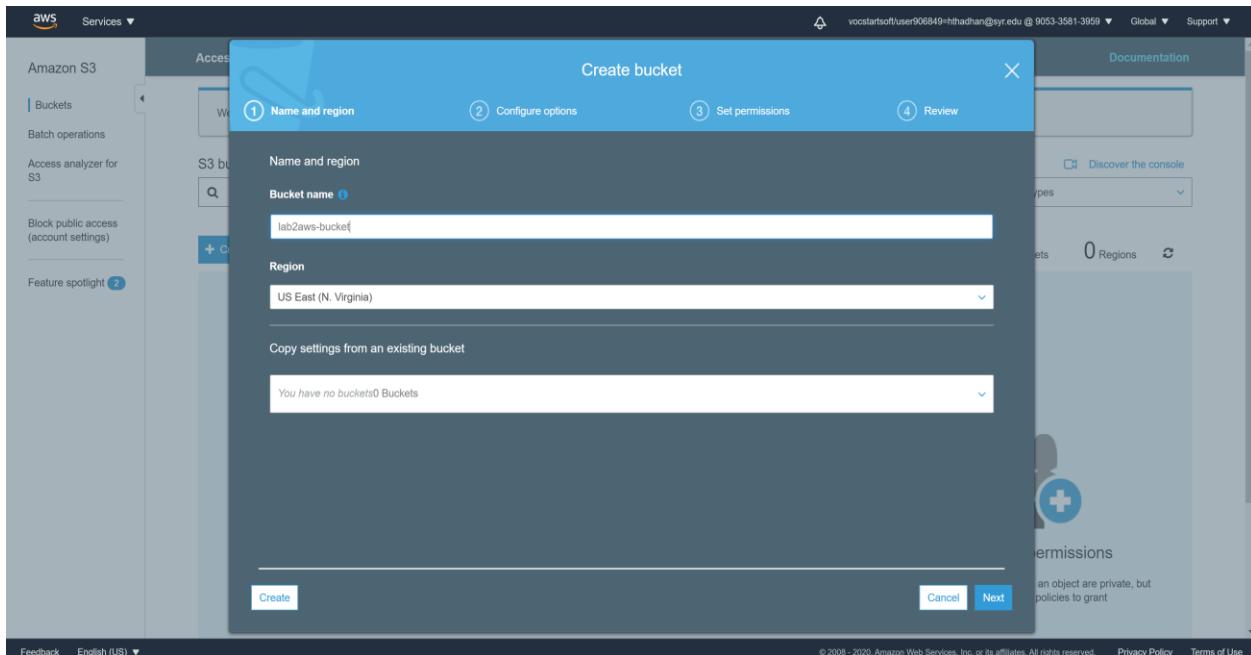
No tags were added.

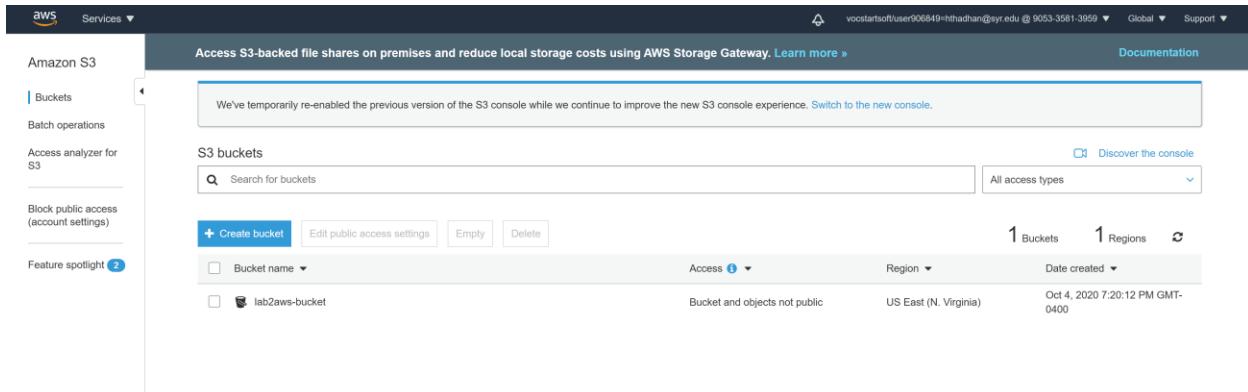
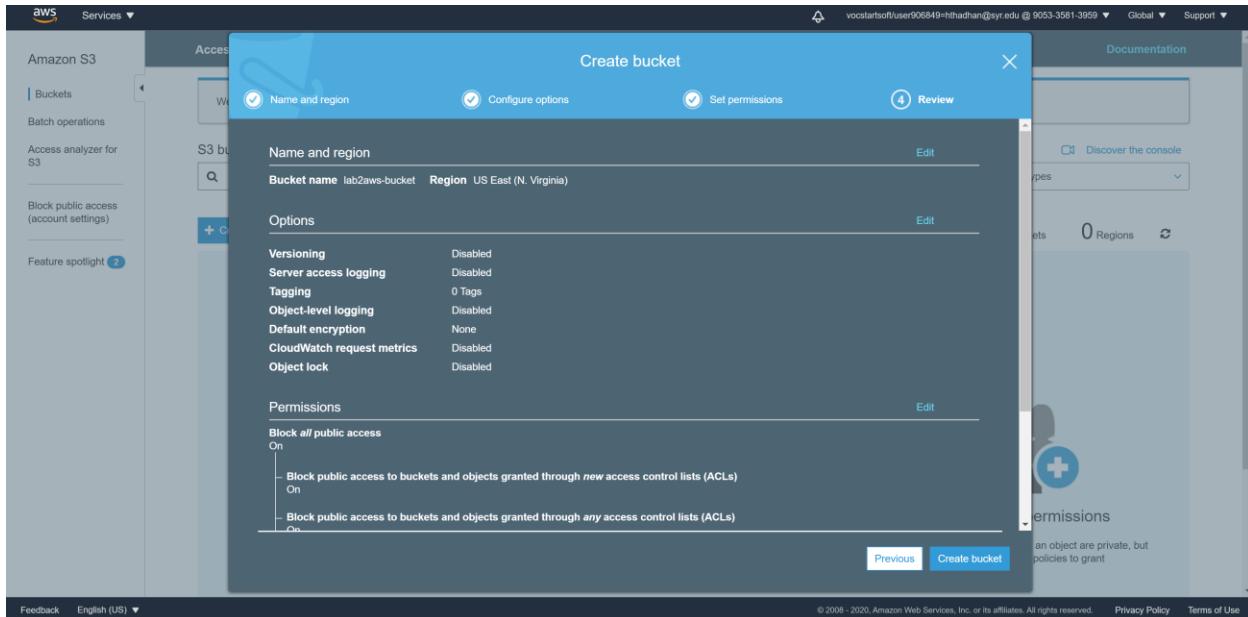
* Required

Cancel Previous Create role

Section 2.2. Creating S3 Bucket:

S3 bucket is created to hold the ML model output data. S3 is Simple Storage Service and the bucket is just like file folders inside of which we store our data. This bucket will contain the pickle file which is the output of the Linear Regression Model we are using to predict the Housing Prices using 2 input attributes. This pickle file is the compiled version of the model output after running the BostonHousingLR.py which contains the ML code to predict the housing price. This same pickle file would be accessed by the 2nd EC2 instance to be able to predict the price of the house based on the values inputted by the user for these 2 attributes.





Section 2.3. Creating First EC2 instance:

Security Groups are created one for each of the 2 EC2 instances. For the 1st EC2 instance, security group with inbound rule of SSH with type anywhere is created to allow the connection to this instance using the default port 22 of the SSH service which is named as SG_EC2.

Security groups are a virtual firewall for the instance and are applied at instance level and not subnet level. These security groups are an additional layer of security to control the incoming (inbound) and outgoing (outbound) traffic to the instance. We create rules in the security group to ensure only certain permitted traffic comes to the instance and if some unwanted traffic comes or someone tries to access something which they should not be from the instance, they are blocked from doing so. So, security groups are a kind firewall for the instances in the VPC.

The screenshot shows the AWS Security Groups configuration page for a security group named "SG_First_EC2".

Security group name: SG_First_EC2

Description: first EC2 instance aws lab 2

VPC: vpc-756b9408

Inbound rules:

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	Anywhere	0.0.0.0/0

Add rule

Outbound rules:

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	Custom	0.0.0.0/0

Add rule

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

The screenshot shows the AWS Security Groups details page for a security group named "sg-059d8dd3cd0ae16c7 - SG_First_EC2".

Details:

- Security group name: SG_First_EC2
- Security group ID: sg-059d8dd3cd0ae16c7
- Description: first EC2 instance aws lab 2
- VPC ID: vpc-756b9408
- Owner: 905335813959
- Inbound rules count: 2 Permission entries
- Outbound rules count: 1 Permission entry

Inbound rules:

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	0.0.0.0/0	-
SSH	TCP	22	:/0	-

1st EC2 instance is created with IAM role as S3FullAccess to allow this instance to make changes to the S3 bucket for the model output.

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1

Purchasing option: Request Spot instances

Network: vpc-756b9408 (default)

Subnet: No preference (default subnet in any Availability Zone)

Auto-assign Public IP: Use subnet setting (Enable)

Placement group: Add instance to placement group

Capacity Reservation: Open

Domain join directory: No directory

IAM role: AwsLab2S3FullAccess

Shutdown behavior: Stop

Stop - Hibernate behavior: Enable hibernation as an additional stop behavior

Enable termination protection: Protect against accidental termination

Monitoring: Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy: Shared - Run a shared hardware instance
Additional charges will apply for dedicated tenancy.

We have added the tag with “ML-Model” as the key and “ML_Model_EC2” as the Value.

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	(128 characters maximum)	Value	(256 characters maximum)	Instances	Volumes
ML-Model		ML_Model_EC2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

(Up to 50 tags maximum)

Security group created earlier with name SG_EC2 to allow inbound SSH traffic is assigned to the instance.

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group Select an existing security group

Security Group ID	Name	Description	Actions
sg-851125b8	default	default VPC security group	Copy to new
sg-0e91cbe52ddff8432	SG_EC2	Security Group EC2 SSH	Copy to new
sg-059d8dd3cd0ae16c7	SG_First_EC2	first EC2 instance aws lab 2	Copy to new

Inbound rules for sg-059d8dd3cd0ae16c7 (Selected security groups: sg-059d8dd3cd0ae16c7)

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0	
SSH	TCP	22	::/0	

[Cancel](#) [Previous](#) [Review and Launch](#)

To access the instance from outside using Putty, we will require the authentication to be in place to authenticate the user with which we are going to login to the instance. For this authentication, we use public/private key pairs to authenticate our login. Public and Private key pairs are created for this instance using the Create a new key pair to be used later on for authentication. We will download the key with .pem extension and use this .pem key to generate the private key using the PuttyGen. So, now the instance has the public key and we have generated the private key with .ppk extension which will serve as pair to authenticate when we login to the instance. These 2 keys are in pair so we are able to authenticate only the connections with this private key using the public key present in the instance. So, we now have only secured connections to the instance using these key pairs which do not allow anyone without private key to login to the instance.

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

⚠ Improve your instances' security. Your security group, SG_First_EC2, is open to the world.
Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.
You can also open additional ports in your security group to facilitate this.

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

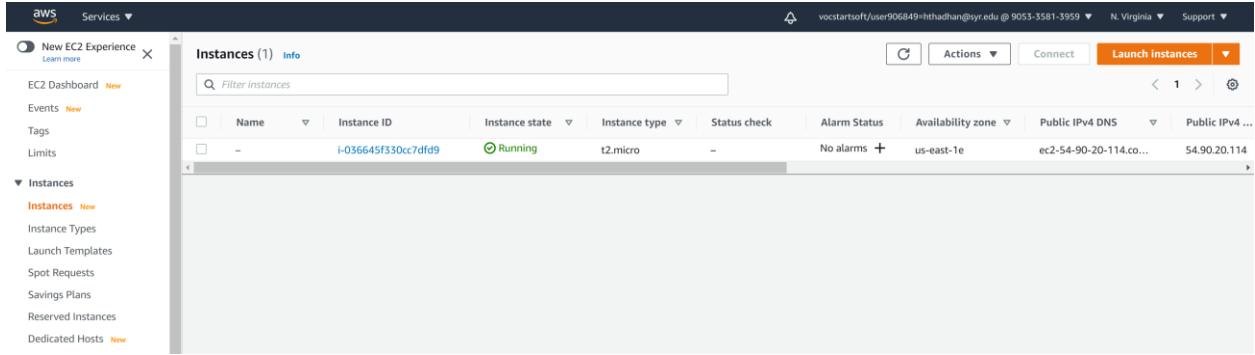
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair
Key pair name: lab2keypair
[Download Key Pair](#)

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

[Cancel](#) [Launch Instances](#)

The 1st EC2 instance is up and running as shown below.



Now that the instance is running and we have setup the key pairs for authentication, we logged in to the instance using the ec2-user to install python and the required packages to be able to run the ML model code in this instance. We have installed python 3.5, pip, numpy, pandas and sklearn ML library in the Python necessary for running the ML code. If these libraries are not installed, our ML model wont be able to run giving the module not found error when running the ML code.

```
ec2-user@ip-172-31-60-182:~$ ssh -i ec2-user.pem ec2-user
Authenticating with public key "imported-openssh-key"
[ec2-user@ip-172-31-60-182 ~]$ sudo yum update -y
Loaded plugins: priorities, update-mtd, upgrade-helper
amzn-main
amzn-updates
No packages marked for updates
[ec2-user@ip-172-31-60-182 ~]$ sudo yum install python35 -y
Loaded plugins: priorities, update-mtd, upgrade-helper
Resolving Dependencies
--> Running transaction check
--> Package python35.x86_64 0:3.5.9-1.28.amzn1 will be installed
--> Processing Dependency: python35-libs(x86-64) = 3.5.9-1.28.amzn1 for package: python35-3.5.9-1.28.amzn1.x86_64
--> Processing Dependency: libpython3.5m.so.1.0() (64bit) for package: python35-3.5.9-1.28.amzn1.x86_64
--> Running transaction check
--> Package python35-libs.x86_64 0:3.5.9-1.28.amzn1 will be installed
--> Finished Dependency Resolution
Dependencies Resolved

Transaction Summary
=====
Install 1 Package (+1 Dependent package)

Total download size: 12 M
Installed size: 38 M
Downloading packages:
(1/2) python35-3.5.9-1.28.amzn1.x86_64.rpm | 56 kB 00:00:00
(2/2) python35-libs-3.5.9-1.28.amzn1.x86_64.rpm | 12 MB 00:00:02
5.9 MB/s | 12 MB 00:00:02
Total
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : python35-3.5.9-1.28.amzn1.x86_64
  Verifying  : python35-3.5.9-1.28.amzn1.x86_64
  Verifying  : python35-3.5.9-1.28.amzn1.x86_64
  Verifying  : python35-3.5.9-1.28.amzn1.x86_64
Installed:
  python35.x86_64 0:3.5.9-1.28.amzn1

Dependency Installed:
  python35-libs.x86_64 0:3.5.9-1.28.amzn1

Complete!
```

```
[ec2-user@ip-172-31-60-182 ~]
[ec2-user@ip-172-31-60-182 ~]$ curl -O https://bootstrap.pypa.io/get-pip.py
  % Total    % Received   Xferd  Average Speed   Time   Time  Current
                                 Dload  Upload   Total   Spent  Left  Speed
000 1041K  100 1041K    0     0      0:--:--:--:--:--:--:--:--:--: 10.9M
[ec2-user@ip-172-31-60-182 ~]$ python3 get-pip.py
DEPRECATION: Python 3.5 reached the end of its life on September 13th, 2020. Please upgrade your Python as Python 3.5 is no longer maintained. pip 21.0 will drop support for Python 3.5 in January 2021. pip 21.0
will remove support for this functionality.
Deferring to user installation because normal site-packages is not writeable
Resolving dependencies...
  Downloading pip-20.2.3-py3-none-any.whl (1.5 MB)
[ec2-user@ip-172-31-60-182 ~]$ pip-20.2.3-py3-none-any.whl 1.5 MB 11.7 MB/s
Collecting setuptools
  Downloading setuptools-50.3.0-py3-none-any.whl (785 KB)
[ec2-user@ip-172-31-60-182 ~]$ setuptools-50.3.0-py3-none-any.whl 785 KB 30.8 MB/s
Collecting wheel
  Downloading wheel-0.35.1-py2.py3-none-any.whl (33 kB)
Resolving dependencies...
Successfully installed pip-20.2.3 setuptools-50.3.0 wheel-0.35.1
[ec2-user@ip-172-31-60-182 ~]$ pip3 install numpy pandas sklearn
DEPRECATION: Python 3.5 reached the end of its life on September 13th, 2020. Please upgrade your Python as Python 3.5 is no longer maintained. pip 21.0 will drop support for Python 3.5 in January 2021. pip 21.0
will remove support for this functionality.
Resolving to user installation because normal site-packages is not writeable
Collecting numpy
  Downloading numpy-1.18.5-cp35-cp35m-manylinux1_x86_64.whl (19.9 MB)
[ec2-user@ip-172-31-60-182 ~]$ numpy-1.18.5-cp35-cp35m-manylinux1_x86_64.whl 19.9 MB 11.4 MB/s
Collecting pandas
  Downloading pandas-0.25.3-cp35-cp35m-manylinux1_x86_64.whl (10.3 MB)
[ec2-user@ip-172-31-60-182 ~]$ pandas-0.25.3-cp35-cp35m-manylinux1_x86_64.whl 10.3 MB 9.7 MB/s
Collecting sklearn
  Downloading sklearn-0.0.tar.gz (1.1 kB)
[ec2-user@ip-172-31-60-182 ~]$ sklearn-0.0.tar.gz 1.1 kB
Collecting pytz>=2017.2
  Downloading pytz->2020.1-py2.py3-none-any.whl (510 kB)
[ec2-user@ip-172-31-60-182 ~]$ pytz->2020.1-py2.py3-none-any.whl 510 kB 31.1 MB/s
Collecting python-dateutil<2.6.1
  Downloading python_dateutil-2.8.1-py2.py3-none-any.whl (227 kB)
[ec2-user@ip-172-31-60-182 ~]$ python_dateutil-2.8.1-py2.py3-none-any.whl 227 kB 36.3 MB/s
Collecting scikit-learn
  Downloading scikit_learn-0.22.2.post1-cp35-cp35m-manylinux1_x86_64.whl (7.0 MB)
[ec2-user@ip-172-31-60-182 ~]$ scikit_learn-0.22.2.post1-cp35-cp35m-manylinux1_x86_64.whl 7.0 MB 11.7 MB/s
Collecting six>1.5
  Downloading six-1.15.0-py3-none-any.whl (10 kB)
[ec2-user@ip-172-31-60-182 ~]$ six-1.15.0-py3-none-any.whl 10 kB
Collecting joblib<0.14
  Downloading joblib-0.14.1-py2.py3-none-any.whl (294 kB)
[ec2-user@ip-172-31-60-182 ~]$ joblib-0.14.1-py2.py3-none-any.whl 294 kB 35.2 MB/s
Collecting scipy>0.17.0
  Downloading scipy-1.4.1-cp35-cp35m-manylinux1_x86_64.whl (26.0 MB)
[ec2-user@ip-172-31-60-182 ~]$ scipy-1.4.1-cp35-cp35m-manylinux1_x86_64.whl 26.0 MB 94 kB/s
Building wheels for collected packages: sklearn
  Building wheel for sklearn (setup.py) ... done
  Created wheel for sklearn: filename=sklearn-0.0-py2.py3-none-any.whl size=2116 sha256=d2e4c398d32alc8a32fb7bd9dd1ef0b28fc9d448c9bef8b57dbb3e255421ae0c
  Stored in directory: /root/.cache/pip/wheeldir/0ec2-a07f3c80b150074231e15479238e100066e607dc75a6f
Successfully built sklearn
Installing collected packages: numpy, pytz, six, python-dateutil, pandas, joblib, scipy, scikit-learn, sklearn
Successfully installed joblib-0.14.1 numpy-1.18.5 pandas-0.25.3 python-dateutil-2.8.1 pytz-2020.1 scikit-learn-0.22.2.post1 scipy-1.4.1 six-1.15.0 sklearn-0.0
[ec2-user@ip-172-31-60-182 ~]$ sudo yum install git -y
Resolving Dependencies
--> Running transaction check
--> Package git.x86_64 0:2.18.4-2.71.amzn1 will be installed
--> Processing Dependency: git-p11k = 2.18.4-2.71.amzn1 for package: git-2.18.4-2.71.amzn1.x86_64
--> Processing Dependency: git-core-doc = 2.18.4-2.71.amzn1 for package: git-2.18.4-2.71.amzn1.x86_64
```

```
ec2-user@ip-172-31-60-182: ~
```

```
Installing collected packages: numpy, pytz, six, python-dateutil, pandas, joblib, scipy, scikit-learn, sklearn
  Found existing installation: numpy 1.14.1
  Overwriting existing installation: numpy (from https://s3.amazonaws.com/ami-mirror/python/2.6/amzn/python-2.6.1/python-2.6.1/scipy-1.4.1/six-1.15.0/scikit-learn-0.0)
  Found existing installation: joblib 0.14.1
  Overwriting existing installation: joblib (from https://s3.amazonaws.com/ami-mirror/python/2.6/amzn/python-2.6.1/python-2.6.1/scipy-1.4.1/six-1.15.0/scikit-learn-0.0)
[ec2-user@ip-172-31-60-182 ~]$ sudo yum install git -y
Loaded plugins: priorities, update-motd, upgrade-helper
Resolving dependencies
Transaction check
  Transaction check error on package git[x86_64] 0:2.18.4-2.71.amzn1
    Package git[x86_64] 0:2.18.4-2.71.amzn1 will be installed
-> Processing Dependency: perl-Git = 2.18.4-2.71.amzn1 for package: git-2.18.4-2.71.amzn1.x86_64
-> Processing Dependency: perl-Git::Term::ReadKey = 2.18.4-2.71.amzn1 for package: git-2.18.4-2.71.amzn1.x86_64
-> Processing Dependency: perl-Term::ReadKey for package: git-2.18.4-2.71.amzn1.x86_64
-> Processing Dependency: perl(GIT) for package: git-2.18.4-2.71.amzn1.x86_64
-> Processing Dependency: perl(GIT) for package: git-2.18.4-2.71.amzn1.x86_64
Running transaction check
  Transaction check error on package git-core[x86_64] 0:2.18.4-2.71.amzn1
    Package git-core[x86_64] 0:2.18.4-2.71.amzn1 will be installed
-> Package git-core-doc.noarch 0:2.18.4-2.71.amzn1 will be installed
-> Package perl-Git.noarch 1:0.1.20-2.9.amanz1 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
-> Package perl2-Error.noarch 1:0.17020-2.9.amzn1 will be installed
-> Package python26-langs.x86_64 0:2.6.9-2.92.amzn1 will be installed
-> Finished Dependency Resolution

Dependencies Resolved

Available Packages
```

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-main	3.1 M
perl-Error	noarch	1:0.1.20-2.9.amanz1	amzn-updates	33 k
perl-TermReadKey	noarch	2.30-20.9.amzn1	amzn-updates	77 k
python26	x86_64	2.6.9-2.92.amzn1	amzn-updates	33 k
python26-langs	x86_64	2.6.9-2.92.amzn1	amzn-updates	5.8 M
python26-langs	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-TermReadKey-2.30-20.9.amzn1.noarch.rpm | 33 kB 00:00:00
  [2/8]: perl-TermReadKey-2.30-20.9.amzn1.noarch.rpm | 33 kB 00:00:00
  [3/8]: git-C-2.18.4-2.71.amzn1.noarch.rpm | 77 kB 00:00:00
  [4/8]: git-2.18.4-2.71.amzn1.x86_64.rpm | 183 kB 00:00:00
  [5/8]: python26-langs-2.6.9-2.92.amzn1.x86_64.rpm | 697 kB 00:00:00
  [6/8]: python26-2.6.9-2.92.amzn1.x86_64.rpm | 1 MB 00:00:01
  [7/8]: python26-2.6.9-2.92.amzn1.x86_64.rpm | 5.8 MB 00:00:00
  [8/8]: git-core-2.18.4-2.71.amzn1.x86_64.rpm | 10 MB 00:00:01
```

```
Total
Running transaction check
  Transaction check error on package git[x86_64] 0:2.18.4-2.71.amzn1
    Package git[x86_64] 0:2.18.4-2.71.amzn1 will be installed
-> Processing Dependency: perl-Git = 2.18.4-2.71.amzn1 for package: git-2.18.4-2.71.amzn1.x86_64
-> Processing Dependency: perl-Git::Term::ReadKey = 2.18.4-2.71.amzn1 for package: git-2.18.4-2.71.amzn1.x86_64
-> Processing Dependency: perl-Term::ReadKey for package: git-2.18.4-2.71.amzn1.x86_64
-> Processing Dependency: perl(GIT) for package: git-2.18.4-2.71.amzn1.x86_64
-> Processing Dependency: perl(GIT) for package: git-2.18.4-2.71.amzn1.x86_64
Running transaction check
  Transaction check error on package git-core[x86_64] 0:2.18.4-2.71.amzn1
    Package git-core[x86_64] 0:2.18.4-2.71.amzn1 will be installed
-> Package git-core-doc.noarch 0:2.18.4-2.71.amzn1 will be installed
-> Package perl-Git.noarch 1:0.1.20-2.9.amanz1 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
-> Package perl2-Error.noarch 1:0.17020-2.9.amzn1 will be installed
-> Package python26-langs.x86_64 0:2.6.9-2.92.amzn1 will be installed
-> Finished Dependency Resolution

Dependencies Resolved

Available Packages
```

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-main	3.1 M
perl-Error	noarch	1:0.1.20-2.9.amanz1	amzn-updates	33 k
perl-TermReadKey	noarch	2.30-20.9.amzn1	amzn-updates	77 k
python26	x86_64	2.6.9-2.92.amzn1	amzn-updates	33 k
python26-langs	x86_64	2.6.9-2.92.amzn1	amzn-updates	5.8 M
python26-langs	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-TermReadKey-2.30-20.9.amzn1.noarch.rpm | 33 kB 00:00:00
  [2/8]: perl-TermReadKey-2.30-20.9.amzn1.noarch.rpm | 33 kB 00:00:00
  [3/8]: git-C-2.18.4-2.71.amzn1.noarch.rpm | 77 kB 00:00:00
  [4/8]: git-2.18.4-2.71.amzn1.x86_64.rpm | 183 kB 00:00:00
  [5/8]: python26-langs-2.6.9-2.92.amzn1.x86_64.rpm | 697 kB 00:00:00
  [6/8]: python26-2.6.9-2.92.amzn1.x86_64.rpm | 1 MB 00:00:01
  [7/8]: python26-2.6.9-2.92.amzn1.x86_64.rpm | 5.8 MB 00:00:00
  [8/8]: git-core-2.18.4-2.71.amzn1.x86_64.rpm | 10 MB 00:00:01
```

```
Total
Running transaction check
  Transaction check error on package git[x86_64] 0:2.18.4-2.71.amzn1
    Package git[x86_64] 0:2.18.4-2.71.amzn1 will be installed
-> Processing Dependency: perl-Git = 2.18.4-2.71.amzn1 for package: git-2.18.4-2.71.amzn1.x86_64
-> Processing Dependency: perl-Git::Term::ReadKey = 2.18.4-2.71.amzn1 for package: git-2.18.4-2.71.amzn1.x86_64
-> Processing Dependency: perl-Term::ReadKey for package: git-2.18.4-2.71.amzn1.x86_64
-> Processing Dependency: perl(GIT) for package: git-2.18.4-2.71.amzn1.x86_64
-> Processing Dependency: perl(GIT) for package: git-2.18.4-2.71.amzn1.x86_64
Running transaction check
  Transaction check error on package git-core[x86_64] 0:2.18.4-2.71.amzn1
    Package git-core[x86_64] 0:2.18.4-2.71.amzn1 will be installed
-> Package git-core-doc.noarch 0:2.18.4-2.71.amzn1 will be installed
-> Package perl-Git.noarch 1:0.1.20-2.9.amanz1 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
-> Package perl2-Error.noarch 1:0.17020-2.9.amzn1 will be installed
-> Package python26-langs.x86_64 0:2.6.9-2.92.amzn1 will be installed
-> Finished Dependency Resolution

Dependencies Resolved

Available Packages
```

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-main	3.1 M
perl-Error	noarch	1:0.1.20-2.9.amanz1	amzn-updates	33 k
perl-TermReadKey	noarch	2.30-20.9.amzn1	amzn-updates	77 k
python26	x86_64	2.6.9-2.92.amzn1	amzn-updates	33 k
python26-langs	x86_64	2.6.9-2.92.amzn1	amzn-updates	5.8 M
python26-langs	x86_64	2.6.9-2.92.amzn1	amzn-updates	697 k

```

[root@ip-172-31-60-182 ~]# yum install git perl-Git perl-TermReadKey python26 python26-libs git-core git-core-doc perl-Error perl-termreadkey python26-termreadkey python26-libperl
Last metadata update on Fri Jul 28 09:00:00 2017.
Dependencies resolved.
Transaction Summary
Install 1 Package (+7 Dependent packages)

Total download size: 20 M
Installed size: 55 M
(1/8) perl-Error-0.17020~2.9.amzn1.noarch.rpm | 33 kB 00:00:00
(2/8) perl-Git-2.18.4-2.71.amzn1.noarch.rpm | 33 kB 00:00:00
(3/8) perl-Git-2.18.4-2.71.amzn1.noarch.rpm | 77 kB 00:00:00
(4/8) git-2.18.4-2.71.amzn1.x86_64.rpm | 183 kB 00:00:00
(5/8) python26-libs-2.6.9-2.92.amzn1.x86_64.rpm | 697 kB 00:00:00
(6/8) git-core-2.18.4-2.71.amzn1.noarch.rpm | 1.1 MB 00:00:00
(7/8) python26-2.6.9-2.92.amzn1.x86_64.rpm | 5.8 MB 00:00:00
(8/8) git-core-2.18.4-2.71.amzn1.x86_64.rpm | 10 MB 00:00:01
Total download size: 20 MB 00:00:01
Local transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : perl-Git-2.18.4-2.71.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 : perl-termreadkey-2.30-20.9.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 : perl-Git-2.18.4-2.71.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-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

Complete!
[root@ip-172-31-60-182 ~]#

```

Section 2.4. Creating second EC2 instance:

The 2nd EC2 instance is created to host the Flask and Angular API. Flask forms the backend and Angular handles the frontend UI presented to the user.

New Security Group named SG_Second_EC2 is created to serve as an extra layer of the firewall to allow only SSH traffic along with some custom rules for allowing the Flask and Angular API traffic coming to the instance. For the UI, port 4200 is white listed to allow inbound traffic to the UI default port and port 8000 is white listed to allow the inbound traffic to the Flask API port. So, we are allowing only the SSH access to connect to the instance along with Flask and Angular API connections incoming requests to this instance using this security group settings/rules created.

Basic details

Security group name: SG_Second_EC2
 Description: Security group for second EC2 instance for Flask and Angular API
 VPC: vpc-756b9408

Inbound rules

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	Anywhere	0.0.0.0/0 :/0 UI API port
Custom TCP	TCP	4200	Anywhere	0.0.0.0/0 :/0 Flask API port
Custom TCP	TCP	8000	Anywhere	0.0.0.0/0 :/0

Add rule

The Security group with SSH, Flask and Angular API ports allowed for the incoming traffic is shown below.

Details

sg-0a543bbd288d98d29 - SG_Second_EC2

Details

Security group name: SG_Second_EC2
 Security group ID: sg-0a543bbd288d98d29
 Description: Security group for second EC2 instance for Flask and Angular API
 VPC ID: vpc-756b9408
 Owner: 905335813959
 Inbound rules count: 6 Permission entries
 Outbound rules count: 1 Permission entry

Inbound rules

Type	Protocol	Port range	Source	Description - optional
Custom TCP	TCP	8000	0.0.0.0/0	Flask API port
Custom TCP	TCP	8000	::/0	Flask API port
SSH	TCP	22	0.0.0.0/0	-
SSH	TCP	22	::/0	-
Custom TCP	TCP	4200	0.0.0.0/0	UI API port
Custom TCP	TCP	4200	::/0	UI API port

This 2nd instance is now created using the IAM role as “S3ReadOnlyAccess” as the instance doesn’t and shouldn’t have to modify any of the data of the model.

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 **Launch into Auto Scaling Group:**

Purchasing option: Request Spot instances

Network: vpc-756b9408 (default) **Create new VPC:**

Subnet: No preference (default subnet in any Availability Zone) **Create new subnet:**

Auto-assign Public IP: Use subnet setting (Enable)

Placement group: Add instance to placement group

Capacity Reservation: Open

Domain join directory: No directory **Create new directory:**

IAM role: AwsLab2S3ReadOnlyAccess **Create new IAM role:**

Shutdown behavior: Stop

Stop - Hibernate behavior: Enable hibernation as an additional stop behavior

Enable termination protection: Protect against accidental termination

Monitoring: Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy: Shared - Run a shared hardware instance
Additional charges will apply for dedicated tenancy.

Plastic instances: Add as Plastic instances

Buttons: Cancel, Previous, **Review and Launch**, Next: Add Storage

Tags with Key as “App” and Value as “Flask_and_UI” is added.

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.

A copy of a tag can be applied to volumes, instances or both.

Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances (1)	Volumes (1)
App	Flask_and_UI	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add another tag (Up to 50 tags maximum)

Buttons: Cancel, Previous, **Review and Launch**, Next: Configure Security Group

Security group created before with inbound traffic allowed access to ports 22, 4200 and 8000 for the SSH, UI and Flask API respectively is assigned to the instance as a virtual firewall.

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group Select an existing security group

Security Group ID	Name	Description	Actions
sg-851125b8	default	default VPC security group	Copy to new
sg-0e91cbe52ddfd8432	SG_EC2	Security Group EC2 SSH	Copy to new
sg-059dd3c030be16c7	SG_First_EC2	first EC2 instance aws lab 2	Copy to new
sg-0a543bbd288d98d29	SG_Second_EC2	Security group for second EC2 instance for Flask and Angular API	Copy to new

Inbound rules for sg-0a543bbd288d98d29 (Selected security groups: sg-0a543bbd288d98d29)

Type	Protocol	Port Range	Source	Description
Custom TCP Rule	TCP	8000	0.0.0.0/0	Flask API port
Custom TCP Rule	TCP	8000	::/0	Flask API port
SSH	TCP	22	0.0.0.0/0	
SSH	TCP	22	::/0	
Custom TCP Rule	TCP	4200	0.0.0.0/0	UI API port

[Cancel](#) [Previous](#) [Review and Launch](#)

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

This instance would be using the same key pairs for the authentication of the connection requests created earlier during the 1st EC2 instance creation. These key pairs would be used to authenticate the login of the user acting as a security layer.

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

Instance Type

Security Groups

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

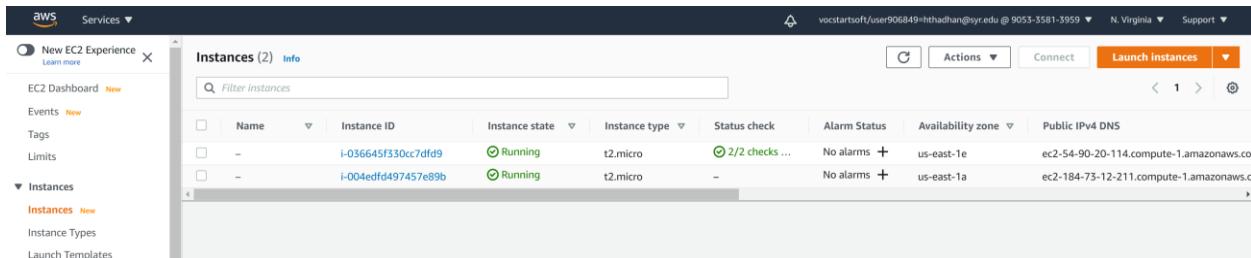
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair
Select a key pair
lab2keypair
 I acknowledge that I have access to the selected private key file (lab2keypair.pem), and that without this file, I won't be able to log into my instance.

[Cancel](#) [Launch Instances](#)

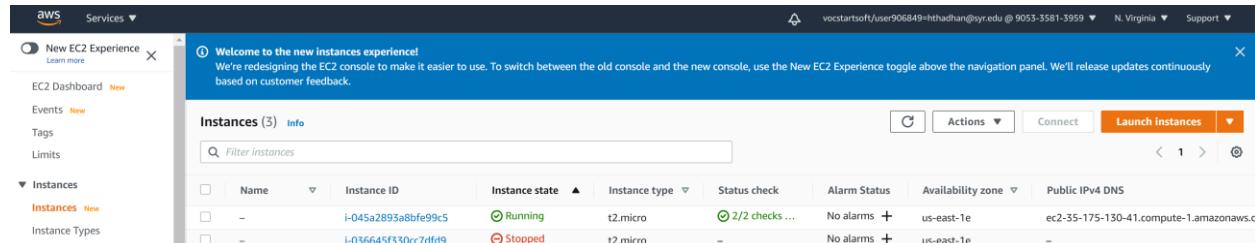
Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Both the instances are now up and running as seen in the screenshot below.



After successfully login to the 2nd instance, we would install python and the required packages to be able to run the Flask and Angular API. We installed python 3.5, pip, numpy, pandas and sklearn machine learning python library to be able make the predictions. We are installing the Nodejs and angular for setting up the Angular and Flask API.

```
[ec2-user@ip-172-31-51-229 ~]$ AWSLabAngular0
[ec2-user@ip-172-31-51-229 ~]$ sudo yum update -y
[sudo] password for ec2-user:
Loaded plugins: priorities, update-motd, upgrade-helper
ezmz-main
ezmz-updates
No packages marked for update
[ec2-user@ip-172-31-51-229 ~]$ sudo yum install python35 -y
Loaded plugins: priorities, update-motd, upgrade-helper
acknowledged python35-3.5.9-1.28.amzn1.x86_64 already installed and latest version
[ec2-user@ip-172-31-51-229 ~]$ curl -o https://bootstrap.pypa.io/get-pip.py
[ec2-user@ip-172-31-51-229 ~]$ curl -o https://bootstrap.pypa.io/get-pip.py
  % Total    % Received   % Xferd  Average Speed   Time     Time      Current
                                 Dload  Upload   Total   Spent    Left  Speed
0     0    0     0    0     0      0      0      0      0      0      0      0
  0 1841k  0 1371  0  23637  0 0:01:19 --:--:--  0:01:19 23637
curl: (23) Failed writing body (0 != 1371)
[ec2-user@ip-172-31-51-229 ~]$ python3 get-pip.py
Collecting pip
  Downloading pip-20.2.3-py3-none-any.whl (1.5 MB)
Installing collected packages: pip
  Successfully installed pip-20.2.3
Collecting numpy<1.18.5,>=1.17.3
  Downloading numpy-1.18.5-cp35-cp35m-manylinux1_x86_64.whl (19.9 MB)
Collecting pandas
  Downloading pandas-0.25.3-cp35-cp35m-manylinux1_x86_64.whl (10.3 MB)
Collecting skLearn
  Downloading skLearn-0.0.tar.gz (1.1 kB)
Collecting flask
  Downloading Flask-1.1.2-py2.py3-none-any.whl (94 kB)
Collecting flask-restful
  Downloading Flask_RESTful-0.3.8-py2.py3-none-any.whl (25 kB)
Collecting flask_cors
  Downloading flask-cors-3.0.0-py2.py3-none-any.whl (14 kB)
Collecting pytz==2017.2
  Downloading pytz-2020.1-py2.py3-none-any.whl (510 kB)
Collecting python-dateutil<2.6.1
  Downloading python_dateutil-2.8.1-py2.py3-none-any.whl (227 kB)
Collecting scikit-learn
  Downloading scikit-learn-0.22.2.post1-cp35-cp35m-manylinux1_x86_64.whl (7.0 MB)
Collecting Werkzeug<1.0.1-py2.py3-none-any.whl (298 kB)
Collecting itsdangerous<0.24
  Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting Jinja2<2.11.1
  Downloading Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)
Collecting Werkzeug<1.0.1-py2.py3-none-any.whl (298 kB)
Collecting itsdangerous<0.24
  Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting Jinja2<2.11.1
  Downloading Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)
[ec2-user@ip-172-31-51-229 ~]$ ls
[ec2-user@ip-172-31-51-229 ~]$
```



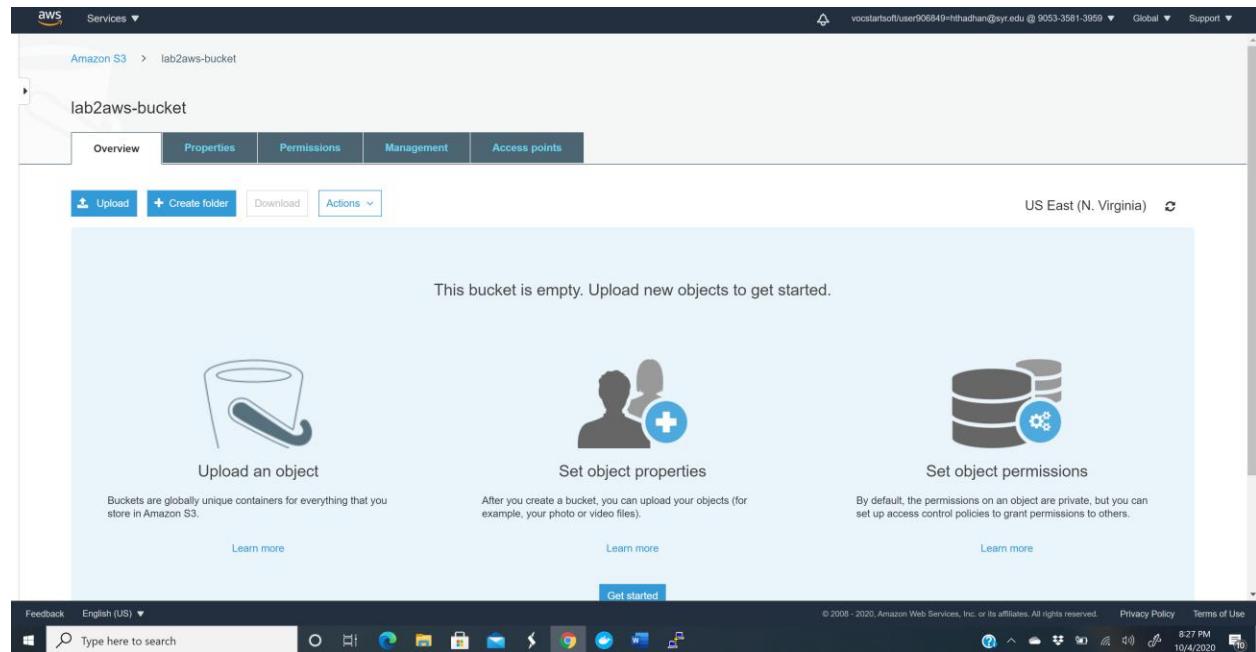
Section 3. Deploying The Code:

Section 3.1. Run ML Code:

ML code to be run to predict the housing prices is run on the 1st instance. We already have the code with the data in the .csv format in the github repository. We are cloning the github repository to get the data along with the ML code to be run. After running the python Regression Model code present on the BostonHousingLR.py, pickle file is generated which is the compiled output of the ML code. This file needs to be transferred to the S3 bucket for the 2nd instance to access the model.

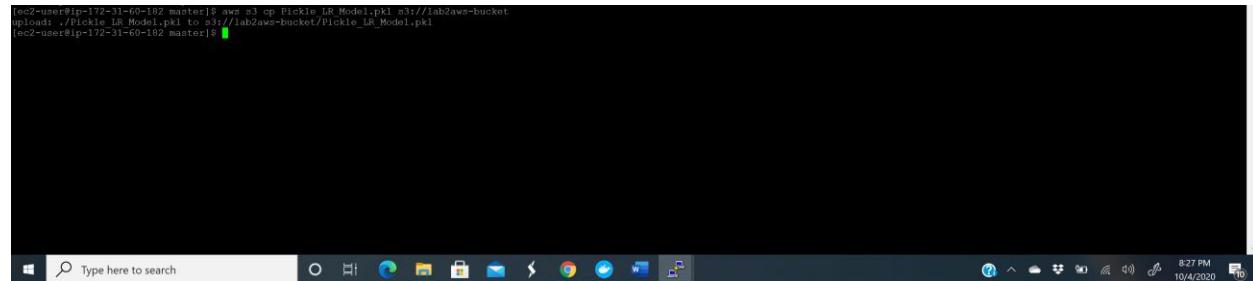
```
ec2-user@ip-172-31-60-102:~/master
$ ssh -i "Amazon Linux AMI.pem" ec2-user@ip-172-31-60-102
Last login: Sun Oct 4 23:39:39 2020 from pool-72-90-94-64.syrcony.fios.verizon.net
[ec2-user@ip-172-31-60-102 ~]$ cd master
[ec2-user@ip-172-31-60-102 master]$ git clone https://github.com/csingh60/AWSLab2MLCode.git master
Cloning into 'master'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100%, 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-172-31-60-102 ~]$ cd master
[ec2-user@ip-172-31-60-102 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-172-31-60-102 master]$ ls
BostonHousingLR.py  Pickle_LR_Model.pkl  README.md
[ec2-user@ip-172-31-60-102 master]$ ll
total 48
-rw-r--r-- 1 ec2-user ec2-user 35734 Oct 5 00:19 BostonHousing.csv
-rw-r--r-- 1 ec2-user ec2-user 979 Oct 5 00:19 BostonHousingLR.py
-rw-r--r-- 1 ec2-user ec2-user 553 Oct 5 00:19 Pickle_LR_Model.pkl
-rw-r--r-- 1 ec2-user ec2-user 15 Oct 5 00:19 README.md
[ec2-user@ip-172-31-60-102 master]$
```

The S3 bucket created earlier is now empty with no files in it as shown below.

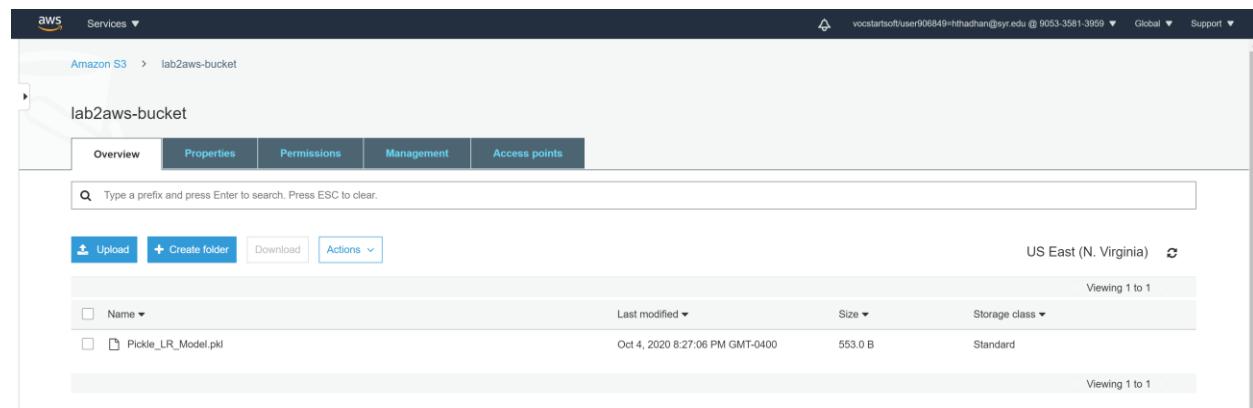


The Pickle file from the Model output is copied to the S3 bucket using the “aws s3 cp command”. The **aws s3 cp file_name s3://s3_bucket_name** is used to transfer the pickle file to the S3 bucket.

```
[ec2-user@ip-172-31-60-182 master]$ aws s3 cp Pickle_LR_Model.pkl s3://lab2aws-bucket
upload: ./Pickle_LR_Model.pkl to s3://lab2aws-bucket/Pickle_LR_Model.pkl
[ec2-user@ip-172-31-60-182 master]$
```



Pickle file is copied to the S3 bucket successfully using the above command.



Name	Last modified	Size	Storage class
Pickle_LR_Model.pkl	Oct 4, 2020 8:27:06 PM GMT-0400	553.0 B	Standard

Section 3.2. Run Flask and Angular Code:

Section 3.2.1. Run Flask Code:

Now we would be deploying the Flask and the Angular API on the EC2 instance. We are first deploying the Flask API code available at the github using the git clone command. Flask is the backend for our ML model and Angular would give us the frontend UI used by users to interact with the app to get housing price predictions.

```
[ec2-user@ip-172-31-51-229 ~]$ git clone https://github.com/ssingh60/AWSLab2FlaskCode.git
Cloning into 'AWSLab2FlaskCode'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 9 (delta 1), reused 2 (delta 0), pack-reused 0
Unpacking objects: 100% (9/9), done.
[ec2-user@ip-172-31-51-229 ~]$ cd AWSLab2FlaskCode/
[ec2-user@ip-172-31-51-229 AWSLab2FlaskCode]$ ll
total 4
-rw-rw-r-- 1 ec2-user ec2-user 642 Oct  6 01:04 modelFlask.py
```

Now, copying the pickle file created earlier from the S3 bucket to this instance in the AWSLab2FlaskCode folder. This file is copied using the “aws s3 cp command”. This file is needed for the Flask API to use this to run in the backend the model to give out the prediction of the housing price to the user on the UI using the Angular API as the frontend. This prediction is

made using 2 attributes which are RM and LSTAT. RM is the average number of rooms per dwelling of the house and LSTAT is the percentage of the lower status of the population.

aws s3 cp se://s3_bucket_name/file_name file_location_2nd_EC2_instance

```
[ec2-user@ip-172-31-51-229 AWSLab2FlaskCode]$ aws s3 cp s3://lab2aws-bucket/Pickle_LR_Model.pkl .
download: s3://lab2aws-bucket/Pickle_LR_Model.pkl to ./Pickle_LR_Model.pkl
[ec2-user@ip-172-31-51-229 AWSLab2FlaskCode]$ ll
total 8
-rw-rw-r-- 1 ec2-user ec2-user 642 Oct  6 01:04 modelFlask.py
-rw-rw-r-- 1 ec2-user ec2-user 553 Oct  5 03:19 Pickle_LR_Model.pkl
```

Now the Flask App for the backend is started by running the python file with .py extension using **nohup python3 modelFlask.py &**

nohup is used to run the Flask app even if the session becomes inactive or terminates so that we do not need to re-run the command again to run the Flask app.

& is used to run the Flask app in the background so that we can run other commands without having to run this in foreground as only 1 foreground process can run on a single shell session.

Below screenshot shows that Flask App is deployed and is running on the port 8000.

```
[ec2-user@ip-172-31-51-229 AWSLab2FlaskCode]$ nohup python3 modelFlask.py &
[1] 6411
[ec2-user@ip-172-31-51-229 AWSLab2FlaskCode]$ nohup: ignoring input and appending output to 'nohup.out'
[ec2-user@ip-172-31-51-229 AWSLab2FlaskCode]$ cat nohup.out
 * Serving Flask app "modelFlask" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:8000/ (Press CTRL+C to quit)
```

Section 3.2.2. Run Angular Code:

We will deploy the Angular UI code on the 2nd EC2 instance. This code is cloned from the github.

Now the Angular API is calling the Flask API to give the output to the user.

To be able to view the output, we are changing the IP address in the web.services.ts file in the src/app folder with the Public IPv4 Address of the EC2 instance as the IP address is different for the EC2 instance where it is running.

```
[ec2-user@ip-172-31-51-229 AWSLab2FlaskCode]$ cd ..
[ec2-user@ip-172-31-51-229 ~]$ 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-172-31-51-229 ~]$ cd AWSLab2AngularUI/
[ec2-user@ip-172-31-51-229 AWSLab2AngularUI]$ vi src/app/web.service.ts
```

The screenshot below shows the IP address is changed to the IPv4 Address of the EC2 instance.

```
[ec2-user@ip-172-31-51-229 AWSLab2AngularUI]$ cat src/app/web.service.ts
import { Injectable } from '@angular/core';
import { Observable, Subject } from 'rxjs';
import { HttpClient } from "@angular/common/http";

@Injectable({
  providedIn: 'root'
})
export class WebService {
  readonly ROOT_URL;

  post$: Observable<any>;
  private myMethodSubject = new Subject<any>();

  constructor(private http: HttpClient) {
    this.ROOT_URL = "http://35.175.130.41:8000/predict";
    this.post$ = this.myMethodSubject.asObservable();
  }
  arrBirds: string [];
  post(rm : any, lstat : any) {
    console.log(rm);
    console.log(lstat);

    this.http.post(`.${this.ROOT_URL}`, rm, lstat);
    return this.http.post(`.${this.ROOT_URL}`, {rm, lstat});
  }
}
[ec2-user@ip-172-31-51-229 AWSLab2AngularUI]$ █
```

npm install command is run to install all the dependencies in the node_modules folder of the Node Package Manager to run the Flask and the Angular API code smoothly without any package dependency requirement not met.

```
[ec2-user@ip-172-31-51-229 AWSLab2AngularUI]$ npm install
> core-js@3.6.4 postinstall /home/ec2-user/AWSLab2AngularUI/node_modules/core-js
> node -e "try{require('./postinstall')}catch(e){}"
Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standard library!
The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
https://opencollective.com/core-js
https://www.patreon.com/zloirock
Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)

> @angular/cli@10.0.5 postinstall /home/ec2-user/AWSLab2AngularUI/node_modules/@angular/cli
> node ./bin/postinstall/script.js

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules/webpack-dev-server/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules/watchpack-chokidar2/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.3 (node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.3: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
added 1458 packages from 1216 contributors and audited 1463 packages in 36.815s

61 packages are looking for funding
  run 'npm fund' for details

found 1 high severity vulnerability
  run 'npm audit fix' to fix them, or 'npm audit' for details
[ec2-user@ip-172-31-51-229 AWSLab2AngularUI]$ npm audit fix
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.3 (node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.3: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules/watchpack-chokidar2/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules/webpack-dev-server/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})

updated 2 packages in 9.804s

61 packages are looking for funding
  run 'npm fund' for details

fixed 1 of 1 vulnerability in 1463 scanned packages
```

Now running the ng serve command will run the Angular CLI application on the host with the port 4200 for us to be able to access the UI of the housing price prediction using the EC2 instance IP address with 4200 port number for the Angular API.

```
[ec2-user@ip-172-31-51-229 AWSLab2AngularUI]$ ng serve --host 0.0.0.0 --port 4200
Your global Angular CLI version (10.1.4) 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: 2020-10-06T01:09:01.221Z - Hash: b86a143761e08676af64 - Time: 14644ms
** Angular Live Development Server is listening on 0.0.0.0:4200, open your browser on http://localhost:4200/ **
: Compiled successfully.
```

Angular API with Flask API in the backend running the ML code to predict the housing prices. 35.175.130.41 is the public IPv4 Address of the EC2 instance with 4200 being the port number for the Angular API to access the app using the browser.

