Rashika Pramod Singh
IST 615

Lab- 3

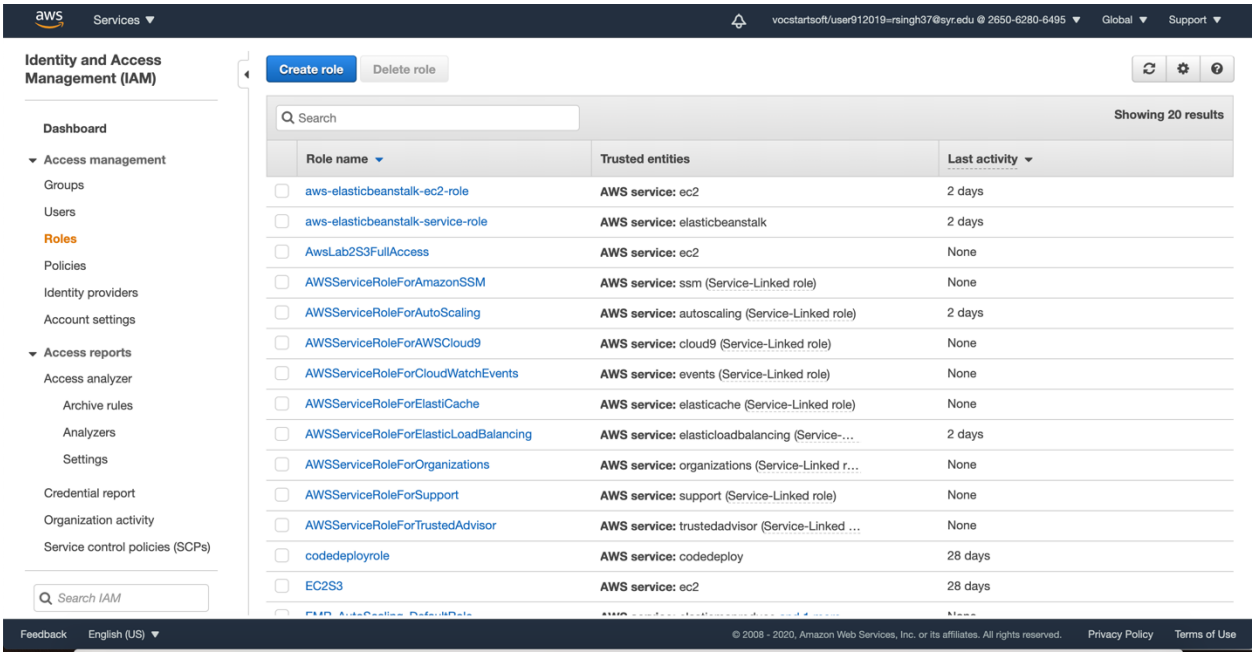
Flask Application on ML Model and Angular UI Deployment on AWS

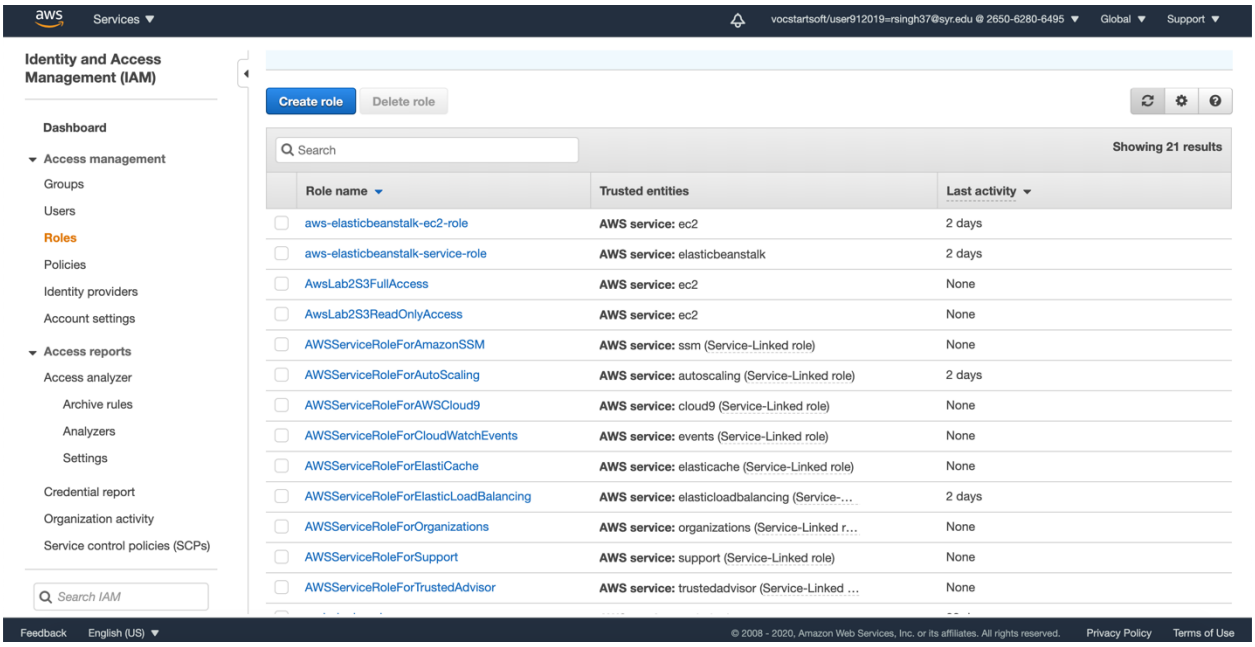
1. Setting up the Infrastructure

We need to create IAM roles for the two instances. The first instance is to pull the code from Github and run python code. This instance will be used to save the model to S3 bucket. Then we clone the Flask application code and Angular UI code from Github. The second instance starts the Flask and Angular UI on the second instance.

- **Creating IAM Roles**

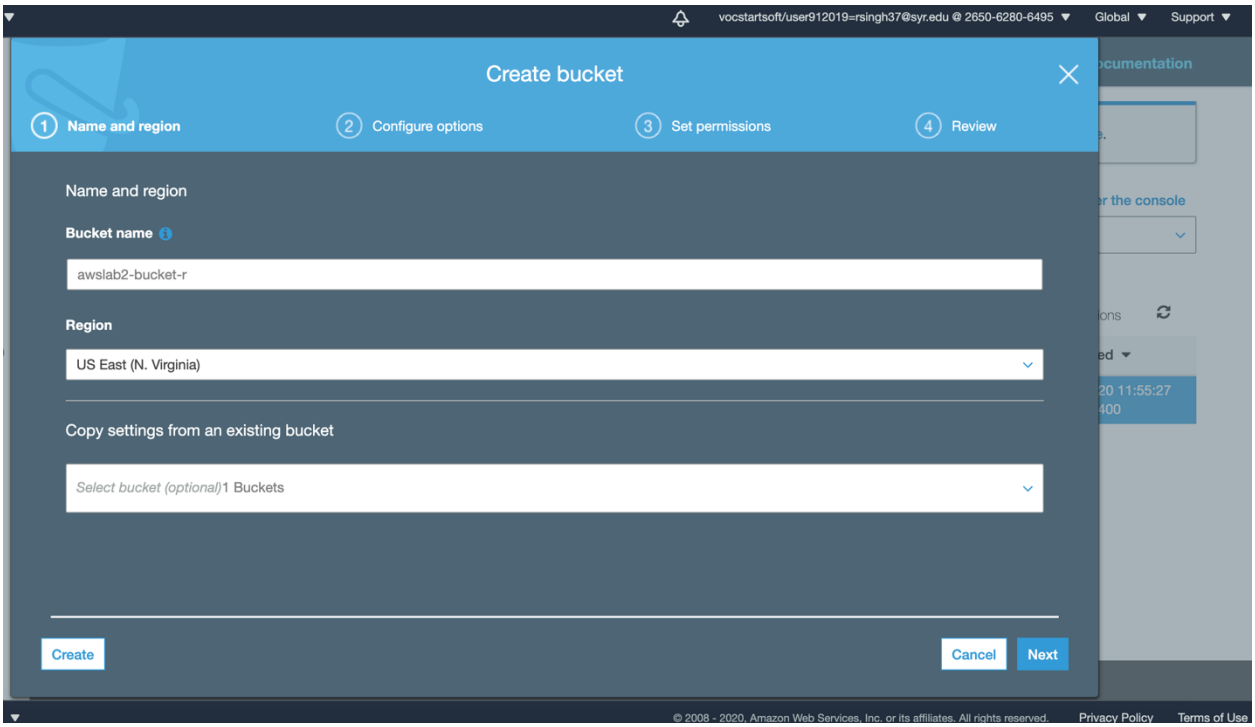
Identity and Access Management enables you to manage access to AWS services and resources securely. Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources. The roles will give the necessary policies and permissions for the EC2 instances. An EC2 instance provides the scalability computing capacity in the cloud. We create the first role to allow full access to the S3 bucket. The second role will enable EC2 to have read access to S3 bucket. The read access can perform read only access while full access allows access to all actions within the IAM service. The roles are created and can be seen in the snippet as "AwsLab2S3FullAccess" and "AmazonS3ReadOnlyAccess."

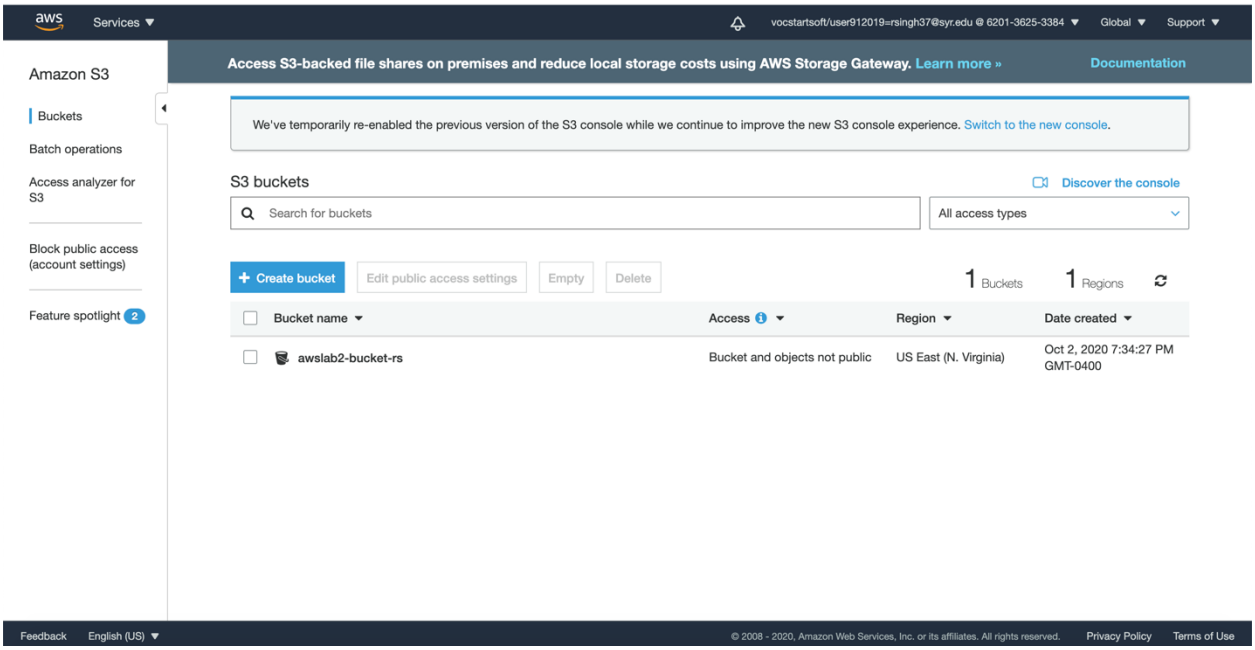




- **Creating S3 Bucket**

An Amazon S3 bucket is a public cloud storage available in AWS to store and retrieve any amount of data from anywhere. It offers scalability, data availability, security and performance which allows customers from all sizes and industries to store data and secure the data for various applications. The S3 bucket which we create will store the ML model output. First, I select S3 service and then create a new bucket with North Virginia as region. You need to give the bucket a unique name, my bucket name is “awslab2-bucket-rs.”

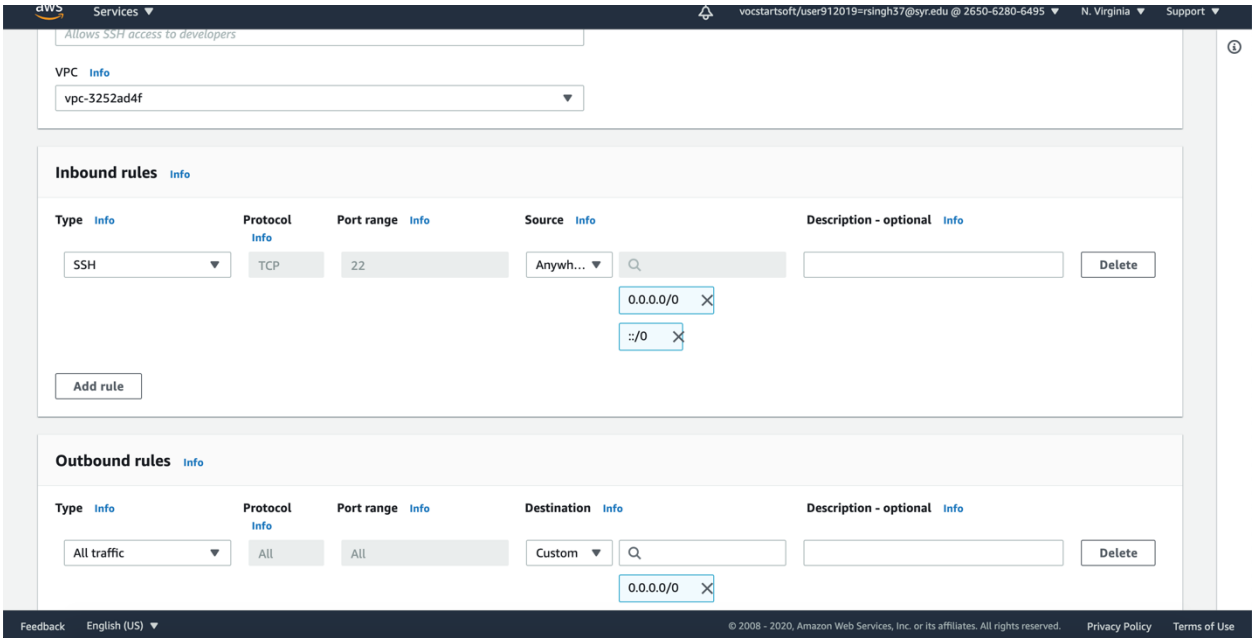


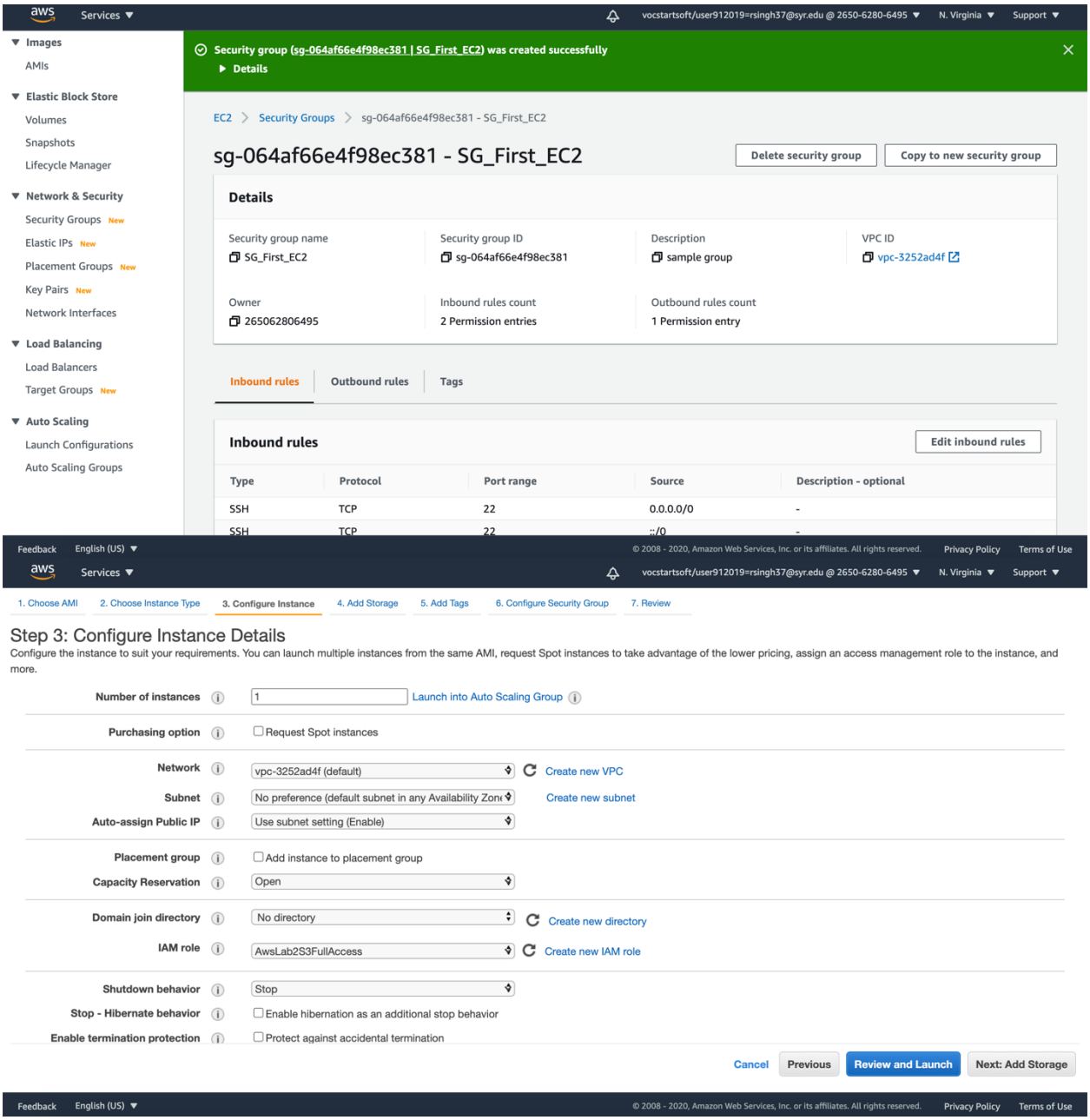


- **Creating First EC2 instance**

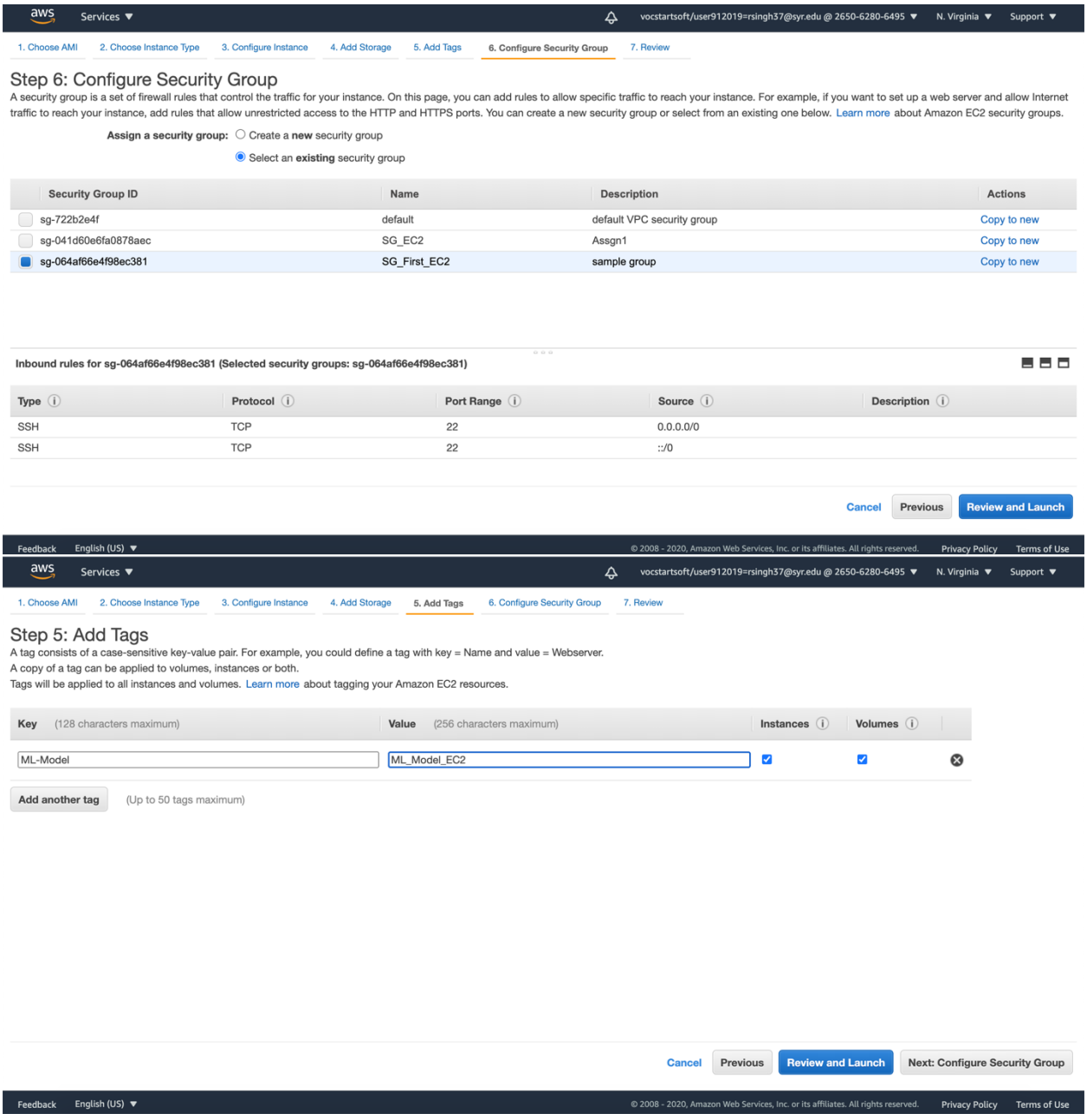
Amazon Elastic Compute Cloud (Amazon EC2) is a web service for providing secure, scalable compute capacity in cloud. It provides the necessary scalability computing to the user and eliminate the need for an infrastructure which offers processor, storage networking, operating system, etc. We create two instances, one for pulling the code from github and run

the python code to generate a pickle file. The pickle file will be pushed into S3 bucket for the storage. The second instance will be used to start the Flask and Angular UI application with the Flask and Angular code from github.
Security groups enable you to control traffic to your instance, including the traffic that can reach your instance. A security group is a virtual firewall for the EC2 instance to control incoming and outgoing traffic. SSH allows you to secure remote login from one computer to another. Inbound rules the incoming traffic into the EC2 instance while outgoing traffic is the outgoing traffic from EC2.
I create the first instance for running the ML code. First, I created a security group named "SG_First_EC2" and added inbound rules by selecting 'SSH' as type and source selected as 'anywhere'. We choose anywhere as we do not require to secure our network range currently. In the step 3, we configure the instance details and attach IAM role Full access which we created. Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. We add a key as "ML-Model" and value as "ML_Model_EC2". I selected "SG_First_EC2" that I created and launched the instance. This will generate an EC2 keypair called "lab2keypair" which I download into my PC and this will be used to SSH into our instance. A keypair is a private and public key that is stored in .pem file which is used to SSH into the instance. The




Services
vocstartsoft/user912019=rsingh37@syr.edu @ 2650-6280-6495
N. Virginia
Support
Images
AMIs
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups New
Elastic IPs New
Placement Groups New
Key Pairs New
Network Interfaces
Load Balancing
Load Balancers
Target Groups New
Auto Scaling
Launch Configurations
Auto Scaling Groups
Security group (sg-064af66e4f98ec381 | SG_First_EC2) was created successfully
Details
EC2 > Security Groups > sg-064af66e4f98ec381 - SG_First_EC2
sg-064af66e4f98ec381 - SG_First_EC2
Delete security group
Copy to new security group
Details
Security group name
SG_First_EC2
Security group ID
sg-064af66e4f98ec381
Description
sample group
VPC ID
vpc-3252ad4f
Owner
265062806495
Inbound rules count
2 Permission entries
Outbound rules count
1 Permission entry
Inbound rules
Outbound rules
Tags
Inbound rules
Edit inbound rules
Type
Protocol
Port range
Source
Description - optional
SSH
TCP
22
0.0.0.0/0
SSH
TCP
22
::/0
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Services
vocstartsoft/user912019=rsingh37@syr.edu @ 2650-6280-6495
N. Virginia
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
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-3252ad4f (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
AwsLab2S3FullAccess
Create new IAM role
Shutdown behavior
Stop
Stop - Hibernate behavior
Enable hibernation as an additional stop behavior
Enable termination protection
Protect against accidental termination
Cancel
Previous
Review and Launch
Next: Add Storage
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



Then I SSH into the EC2 instance using IPv4 public IP using the command “ssh -I ‘name of file’ @ec2-user <public DNS name>”. Once this done, I download the required packages and execute the commands on EC2 instance. I used the chmod to change the privacy setting for the key. We use the sudo yum command to check if there are any updates for our installed packages. The sudo yum install is used to install the python version and curl is used to transfer data to or from a server here it is used to install pip which allows you to install packages. After that I install python libraries such as numpy, pandas, scikit learn and install git.

```
Rashikas-Air:Downloads rashikasingh$ chmod 400 lab2keypair.pem
Rashikas-Air:Downloads rashikasingh$ ssh -i lab2keypair.pem ec2-user@ec2-52-90-223-110.compute-1.amazonaws.com
The authenticity of host 'ec2-52-90-223-110.compute-1.amazonaws.com (52.90.223.110)' can't be established.
ECDSA key fingerprint is SHA256:wHsU8P330wchha+FwJtZ68z8Mo9KnpfB2KM98Yl0xN0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-90-223-110.compute-1.amazonaws.com,52.90.223.110' (ECDSA) to the list of known hosts.

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

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
[ec2-user@ip-172-31-28-90 ~]$
```

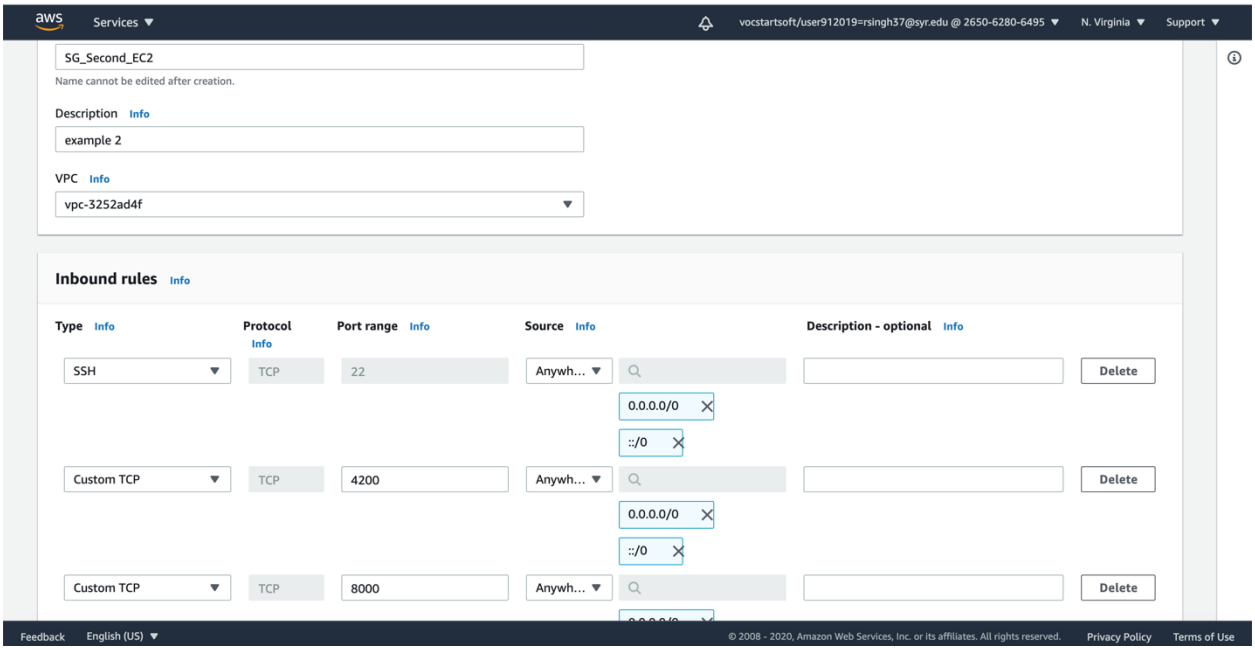
```
https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
[ec2-user@ip-172-31-28-90 ~]$ sudo yum update -y
Loaded plugins: priorities, update-motd, upgrade-helper
amzn-main                                                | 2.1 kB     00:00
amzn-updates                                             | 3.8 kB     00:00
No packages marked for update
[ec2-user@ip-172-31-28-90 ~]$ sudo yum install python35 -y
Loaded plugins: priorities, update-motd, 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

================================================================================
 Package          Arch          Version                 Repository         Size
================================================================================
Installing:
 python35         x86_64        3.5.9-1.28.amzn1        amzn-updates       56 k
Installing for dependencies:
 python35-libs    x86_64        3.5.9-1.28.amzn1        amzn-updates       12 M

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
(2/2): python35-libs-3.5.9-1.28.amzn1.x86_64.rpm          |  12 MB   00:02
--------------------------------------------------------------------------------
Total                                          4.9 MB/s |  12 MB  00:02
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : python35-3.5.9-1.28.amzn1.x86_64                             1/2
  Installing : python35-libs-3.5.9-1.28.amzn1.x86_64                        2/2
  Verifying  : python35-libs-3.5.9-1.28.amzn1.x86_64                        1/2
  Verifying  : python35-3.5.9-1.28.amzn1.x86_64                             2/2

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-172-31-28-90 ~]$
```

```
---> 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

================================================================================
 Package            Arch         Version                  Repository       Size
================================================================================
Installing:
 python35           x86_64       3.5.9-1.28.amzn1         amzn-updates     56 k
Installing for dependencies:
 python35-libs      x86_64       3.5.9-1.28.amzn1         amzn-updates     12 M

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
(2/2): python35-libs-3.5.9-1.28.amzn1.x86_64.rpm           |  12 MB   00:02
--------------------------------------------------------------------------------
Total                                          4.9 MB/s |  12 MB  00:02
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : python35-3.5.9-1.28.amzn1.x86_64                             1/2
  Installing : python35-libs-3.5.9-1.28.amzn1.x86_64                        2/2
  Verifying  : python35-libs-3.5.9-1.28.amzn1.x86_64                        1/2
  Verifying  : python35-3.5.9-1.28.amzn1.x86_64                             2/2

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-172-31-28-90 ~]$ curl -O https://bootstrap.pypa.io/get-pip.py
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 1841k  100 1841k    0     0  17.6M      0 --:--:-- --:--:-- --:--:-- 17.6M
[ec2-user@ip-172-31-28-90 ~]$ 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.
Defaulting to user installation because normal site-packages is not writeable
Collecting pip
  Downloading pip-20.2.3-py2.py3-none-any.whl (1.5 MB)
     |████████████████████████████████| 1.5 MB 11.2 MB/s
Collecting setuptools
  Downloading setuptools-50.3.0-py3-none-any.whl (785 kB)
     |████████████████████████████████| 785 kB 24.7 MB/s
Collecting wheel
  Downloading wheel-0.35.1-py2.py3-none-any.whl (33 kB)
Installing collected packages: pip, setuptools, wheel
Successfully installed pip-20.2.3 setuptools-50.3.0 wheel-0.35.1
[ec2-user@ip-172-31-28-90 ~]$



Successfully installed pip-20.2.3 setuptools-50.3.0 wheel-0.35.1
[ec2-user@ip-172-31-28-90 ~]$ 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.
Defaulting 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)
     |████████████████████████████████| 19.9 MB 14.2 MB/s
Collecting pandas
  Downloading pandas-0.25.3-cp35-cp35m-manylinux1_x86_64.whl (10.3 MB)
     |████████████████████████████████| 10.3 MB 6.7 MB/s
Collecting sklearn
  Downloading sklearn-0.0.tar.gz (1.1 kB)
Collecting python-dateutil>=2.6.1
  Downloading python_dateutil-2.8.1-py2.py3-none-any.whl (227 kB)
     |████████████████████████████████| 227 kB 36.7 MB/s
Collecting pytz>=2017.2
  Downloading pytz-2020.1-py2.py3-none-any.whl (510 kB)
     |████████████████████████████████| 510 kB 38.1 MB/s
Collecting scikit-learn
  Downloading scikit_learn-0.22.2.post1-cp35-cp35m-manylinux1_x86_64.whl (7.0 MB)
     |████████████████████████████████| 7.0 MB 5.3 MB/s
Collecting six>=1.5
  Downloading six-1.15.0-py2.py3-none-any.whl (10 kB)
Collecting scipy>=0.17.0
  Downloading scipy-1.4.1-cp35-cp35m-manylinux1_x86_64.whl (26.0 MB)
     |████████████████████████████████| 26.0 MB 145 kB/s
Collecting joblib>=0.11
  Downloading joblib-0.14.1-py2.py3-none-any.whl (294 kB)
     |████████████████████████████████| 294 kB 38.4 MB/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=1316 sha256=562a3e3b4bc286472a10e7a3177f3fd36355cb3a93334931a8e363baa7813d92
  Stored in directory: /home/ec2-user/.cache/pip/wheels/9e/ec/a6/33cdb5605b0b150074213e154792654a1006e6e6807dc7ca6f
Successfully built sklearn
Installing collected packages: numpy, six, python-dateutil, pytz, pandas, scipy, joblib, 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-172-31-28-90 ~]$ sudo yum install git -y
Loaded plugins: priorities, update-motd, upgrade-helper
Resolving Dependencies
--> Running transaction check
---> 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: git-core-doc = 2.18.4-2.71.amzn1 for package: git-2.18.4-2.71.amzn1.x86_64
--> Processing Dependency: git-core = 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::I18N) 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: /usr/bin/python2.6 for package: git-2.18.4-2.71.amzn1.x86_64
--> Running transaction check
---> 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 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
```

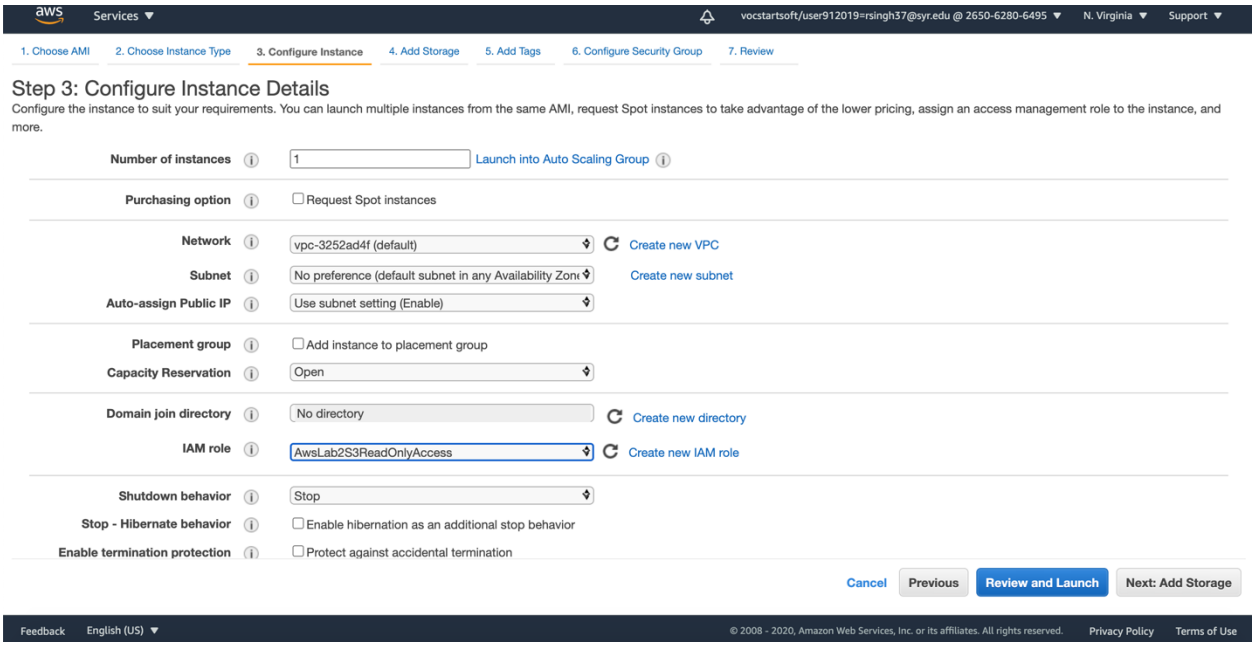
```
Transaction Summary
================================================================================
Install  1 Package (+7 Dependent packages)

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-Git-2.18.4-2.71.amzn1.noarch.rpm                |  77 kB   00:00
(3/8): perl-TermReadKey-2.30-20.9.amzn1.x86_64.rpm          |  33 kB   00:00
(4/8): git-2.18.4-2.71.amzn1.x86_64.rpm                     | 183 kB   00:00
(5/8): python26-libs-2.6.9-2.92.amzn1.x86_64.rpm            | 697 kB   00:00
(6/8): git-core-doc-2.18.4-2.71.amzn1.noarch.rpm            | 3.1 MB   00:01
(7/8): python26-2.6.9-2.92.amzn1.x86_64.rpm                                 | 5.8 MB  00:00:01
(8/8): git-core-2.18.4-2.71.amzn1.x86_64.rpm                                |  10 MB  00:00:02
--------------------------------------------------------------------------------
Total                                                       7.5 MB/s |  20 MB  00:00:02
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-172-31-28-90 ~]$
```

- **Creating second EC2 instance**

We create a second EC2 instance for hosting the Flask API and Angular UI. Security group acts as a virtual firewall to the instance to control inbound and outbound traffic. We can assign upto five security groups to any instance. I created a security group named "SG_Second_EC2". I added an inbound rule by selecting "SSH" and source as "anywhere" as we do not need to secure at this stage. We add inbound rules by "Custom TCP" as type and ports as 4200 for UI and 8000 for flask API and source as "anywhere." The rules for UI and flask API allow the incoming traffic into our EC2 instance and the port numbers are default for UI and flask API.



I added the IAM role with Read only access. For this instance, we add a key as “App” and value as “Flask_and_UI”. The keypair is already created which I use to SSH into the instance using IPv4 public IP using the command “ssh -I ‘name of file’ @ec2-user <public DNS name>”

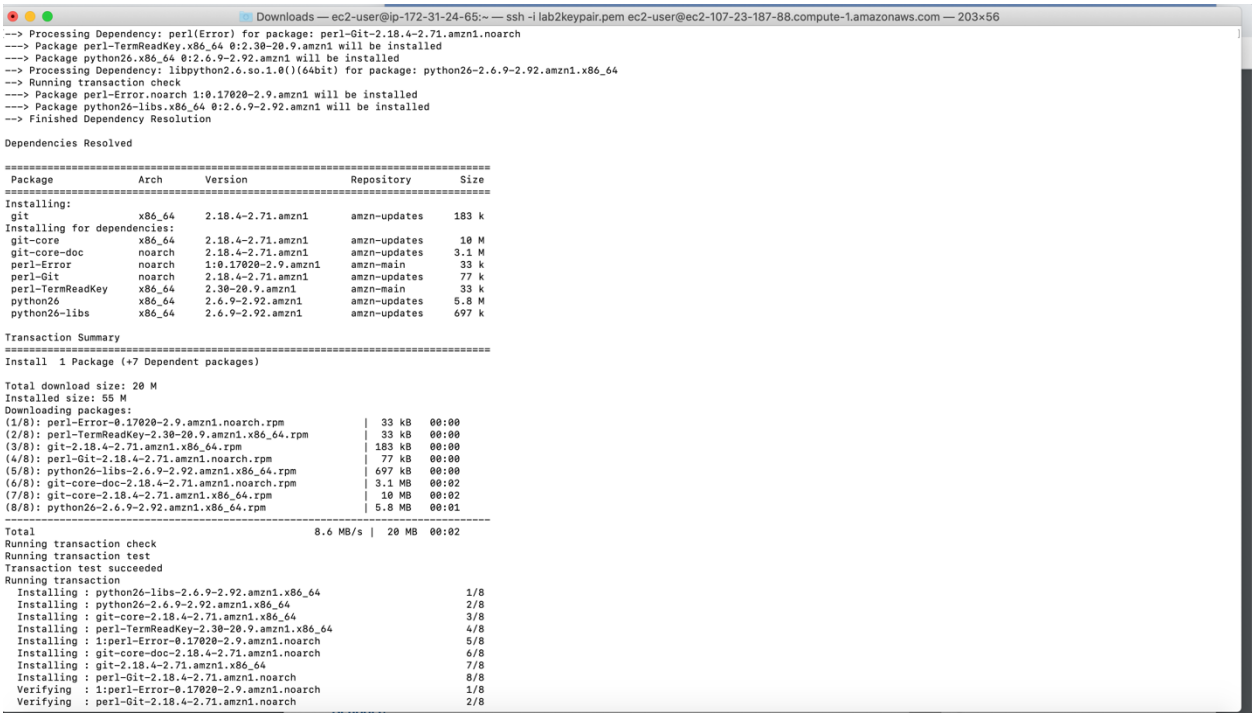


Next, I downloaded the required packages by the commands. Sudo yum upgrade should pull the updates. Then, I install the python version and use Curl to connect to the server. Similarly, to the first instance, I install pip and install numpy, pandas, skikit learn, node js, node package manager, flask packages, angular commands and git. I also use the node version manager and install node js, followed by installing httpd and start its service and command line interface.

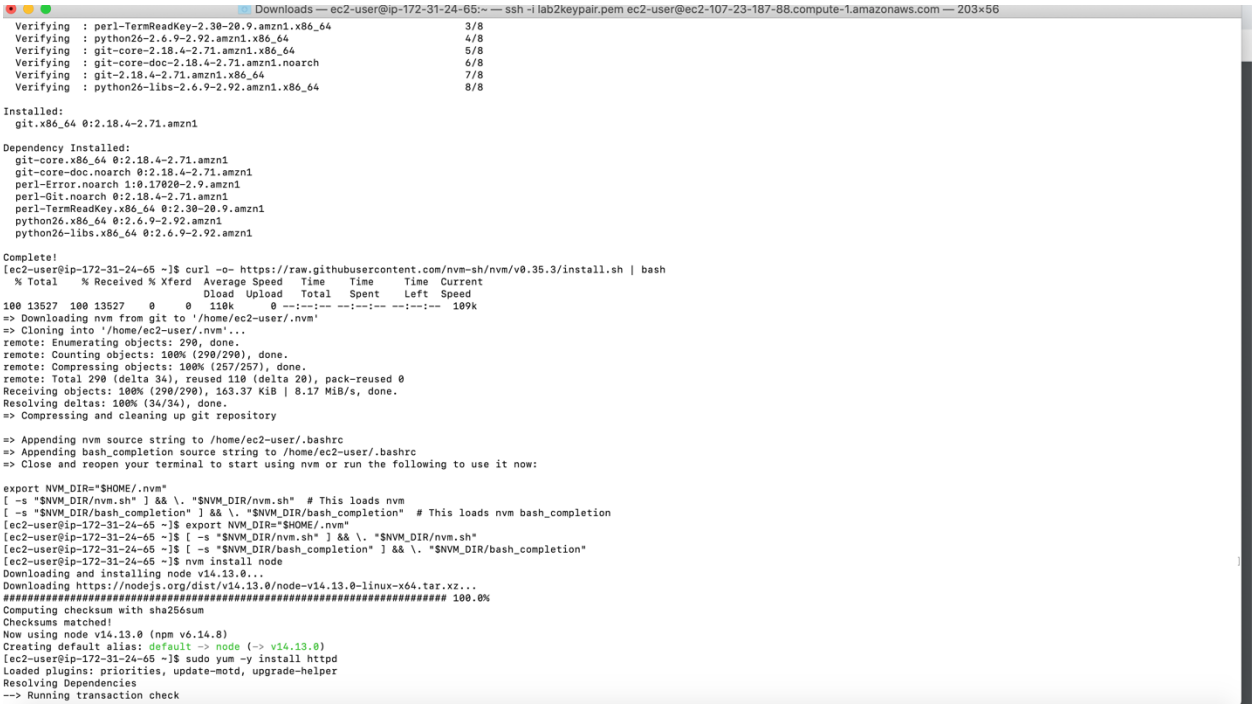
```
Last login: Fri Oct  2 19:41:10 on ttys001

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Rashikas-Air:~ rashikasingh$ cd Downloads
Rashikas-Air:Downloads rashikasingh$ chmod 400 lab2keypair.pem
Rashikas-Air:Downloads rashikasingh$ ssh -i lab2keypair.pem ec2-user@ec2-107-23-187-88.compute-1.amazonaws.com
The authenticity of host 'ec2-107-23-187-88.compute-1.amazonaws.com (107.23.187.88)' can't be established.
ECDSA key fingerprint is SHA256:6V+OSC1EEUJZzeH1y5jIOSZUOKYz78+nvpeiqlb+U38.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-107-23-187-88.compute-1.amazonaws.com,107.23.187.88' (ECDSA) to the list of known hosts.

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

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
[ec2-user@ip-172-31-24-65 ~]$ sudo yum update -y
Loaded plugins: priorities, update-motd, upgrade-helper
No packages marked for update
[ec2-user@ip-172-31-24-65 ~]$ sudo yum install python35 -y
Loaded plugins: priorities, update-motd, 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

================================================================================
 Package            Arch        Version                Repository         Size
================================================================================
Installing:
 python35           x86_64      3.5.9-1.28.amzn1       amzn-updates       56 k
Installing for dependencies:
 python35-libs      x86_64      3.5.9-1.28.amzn1       amzn-updates       12 M

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
(2/2): python35-libs-3.5.9-1.28.amzn1.x86_64.rpm          |  12 MB   00:01
--------------------------------------------------------------------------------
Total                                          7.3 MB/s |  12 MB  00:01
Running transaction check
Running transaction test
Transaction test succeeded
```

```
Running transaction test
Transaction test succeeded
Running transaction
  Installing : python35-3.5.9-1.28.amzn1.x86_64                             1/2
  Installing : python35-libs-3.5.9-1.28.amzn1.x86_64                        2/2
  Verifying  : python35-libs-3.5.9-1.28.amzn1.x86_64                        1/2
  Verifying  : python35-3.5.9-1.28.amzn1.x86_64                             2/2

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-172-31-24-65 ~]$ curl -O https://bootstrap.pypa.io/get-pip.py
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 1841k  100 1841k    0     0  17.1M      0 --:--:-- --:--:-- --:--:-- 17.1M
[ec2-user@ip-172-31-24-65 ~]$ 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.
Defaulting to user installation because normal site-packages is not writeable
Collecting pip
  Downloading pip-20.2.3-py2.py3-none-any.whl (1.5 MB)
     |████████████████████████████████| 1.5 MB 14.7 MB/s
Collecting setuptools
  Downloading setuptools-50.3.0-py3-none-any.whl (785 kB)
     |████████████████████████████████| 785 kB 40.6 MB/s
Collecting wheel
  Downloading wheel-0.35.1-py2.py3-none-any.whl (33 kB)
Installing collected packages: pip, setuptools, wheel
Successfully installed pip-20.2.3 setuptools-50.3.0 wheel-0.35.1
[ec2-user@ip-172-31-24-65 ~]$ pip3 install numpy pandas sklearn flask flask_restful flask_cors
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.
Defaulting 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)
     |████████████████████████████████| 19.9 MB 16.3 MB/s
Collecting pandas
  Downloading pandas-0.25.3-cp35-cp35m-manylinux1_x86_64.whl (10.3 MB)
     |████████████████████████████████| 10.3 MB 13.0 MB/s
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)
     |████████████████████████████████| 94 kB 3.8 MB/s
Collecting flask_restful
  Downloading Flask_RESTful-0.3.8-py2.py3-none-any.whl (25 kB)
Collecting flask_cors
  Downloading Flask_Cors-3.0.9-py2.py3-none-any.whl (14 kB)
Collecting pytz>=2017.2
  Downloading pytz-2020.1-py2.py3-none-any.whl (510 kB)
     |████████████████████████████████| 510 kB 47.8 MB/s
Collecting python-dateutil>=2.6.1
```

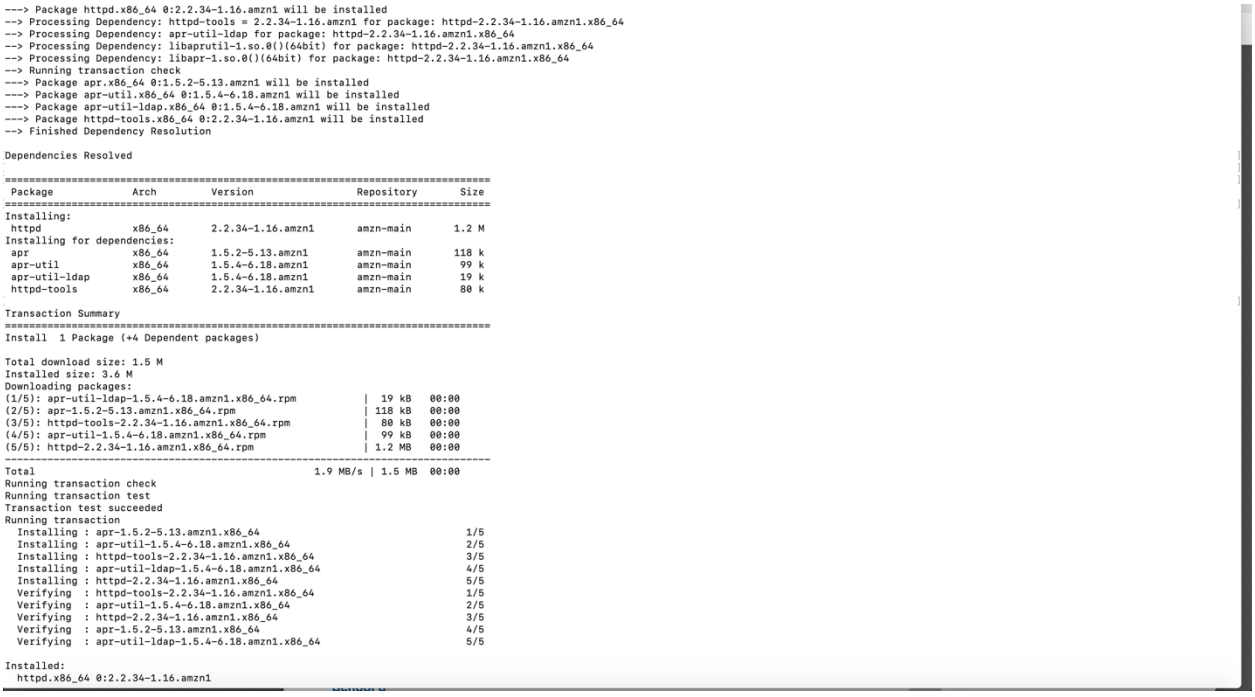

```
Collecting python-dateutil>=2.6.1
  Downloading python_dateutil-2.8.1-py2.py3-none-any.whl (227 kB)
     |████████████████████████████████| 227 kB 44.4 MB/s
Collecting scikit-learn
  Downloading scikit_learn-0.22.2.post1-cp35-cp35m-manylinux1_x86_64.whl (7.0 MB)
     |████████████████████████████████| 7.0 MB 6.8 MB/s
Collecting itsdangerous>=0.24
  Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting Jinja2>=2.10.1
  Downloading Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)
     |████████████████████████████████| 125 kB 39.2 MB/s
Collecting Werkzeug>=0.15
  Downloading Werkzeug-1.0.1-py2.py3-none-any.whl (298 kB)
     |████████████████████████████████| 298 kB 41.1 MB/s
Collecting click>=5.1
  Downloading click-7.1.2-py2.py3-none-any.whl (82 kB)
     |████████████████████████████████| 82 kB 825 kB/s
Collecting six>=1.3.0
  Downloading six-1.15.0-py2.py3-none-any.whl (10 kB)
Collecting aniso8601>=0.82
  Downloading aniso8601-8.0.0-py2.py3-none-any.whl (43 kB)
     |████████████████████████████████| 43 kB 2.8 MB/s
Collecting scipy>=0.17.0
  Downloading scipy-1.4.1-cp35-cp35m-manylinux1_x86_64.whl (26.0 MB)
     |████████████████████████████████| 26.0 MB 6.8 MB/s
Collecting joblib>=0.11
  Downloading joblib-0.14.1-py2.py3-none-any.whl (294 kB)
     |████████████████████████████████| 294 kB 37.8 MB/s
Collecting MarkupSafe>=0.23
  Downloading MarkupSafe-1.1.1-cp35-cp35m-manylinux1_x86_64.whl (27 kB)
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=1316 sha256=135ee88ec88d52f6c7d698a71d0cc12f78d9e4533d307e482182f83b7e7083ce
  Stored in directory: /home/ec2-user/.cache/pip/wheels/9e/ec/a6/33cdb5605b0b150074213e154792654a1006e6e6807dc7ca6f
Successfully built sklearn
Installing collected packages: numpy, pytz, six, python-dateutil, pandas, scipy, joblib, scikit-learn, sklearn, itsdangerous, MarkupSafe, Jinja2, Werkzeug, click, flask, aniso8601, flask-restful, flask-c
ors
Successfully installed Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-1.0.1 aniso8601-8.0.0 click-7.1.2 flask-1.1.2 flask-cors-3.0.9 flask-restful-0.3.8 itsdangerous-1.1.0 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-172-31-24-65 ~]$ sudo yum install git -y
Loaded plugins: priorities, update-motd, upgrade-helper
Resolving Dependencies
--> Running transaction check
---> 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: git-core-doc = 2.18.4-2.71.amzn1 for package: git-2.18.4-2.71.amzn1.x86_64
--> Processing Dependency: git-core = 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::I18N) 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: /usr/bin/python2.6 for package: git-2.18.4-2.71.amzn1.x86_64
--> Running transaction check
---> 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 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
```



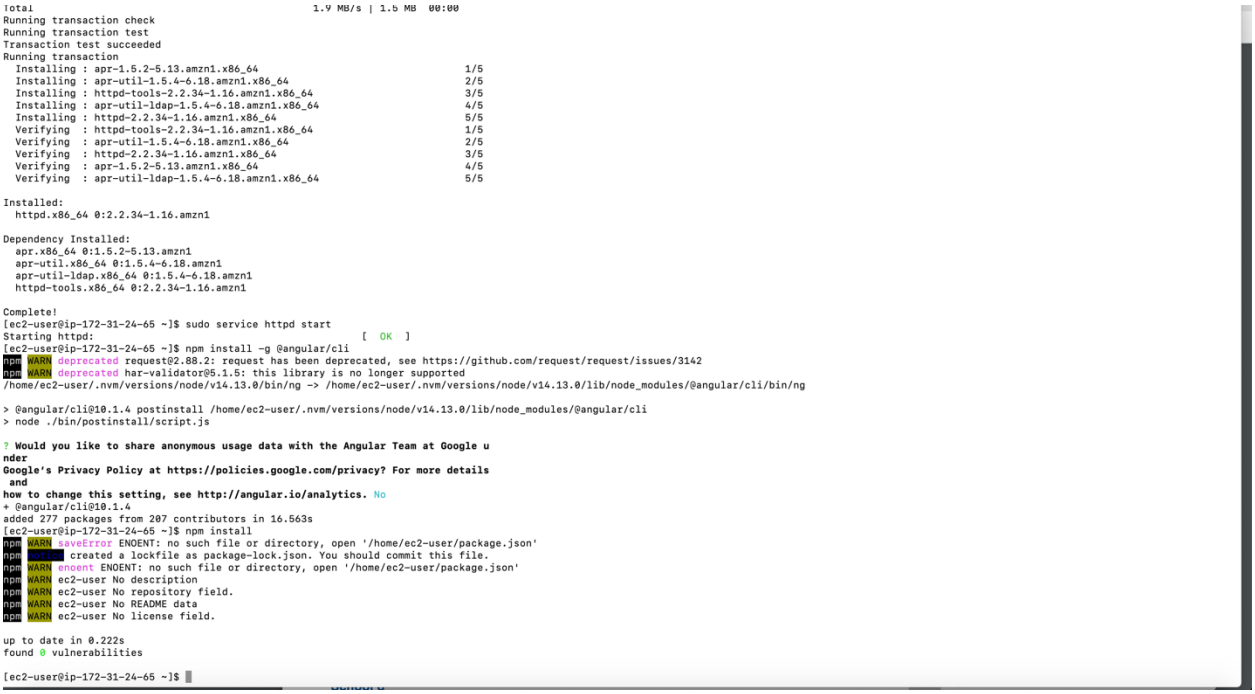

```
--> Processing Dependency: perl(Error) for package: perl-Git-2.18.4-2.71.amzn1.noarch
---> Package perl-TermReadKey.x86_64 0:2.30-20.9.amzn1 will be installed
---> Package python26.x86_64 0:2.6.9-2.92.amzn1 will be installed
--> Processing Dependency: libpython2.6.so.1.0()(64bit) for package: python26-2.6.9-2.92.amzn1.x86_64
--> 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): python26-libs-2.6.9-2.92.amzn1.x86_64.rpm           | 697 kB   00:00
(6/8): git-core-doc-2.18.4-2.71.amzn1.noarch.rpm           | 3.1 MB   00:02
(7/8): git-core-2.18.4-2.71.amzn1.x86_64.rpm               |  10 MB   00:02
(8/8): python26-2.6.9-2.92.amzn1.x86_64.rpm                | 5.8 MB   00:01
--------------------------------------------------------------------------------
Total                                           8.6 MB/s |  20 MB   00:02
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
```

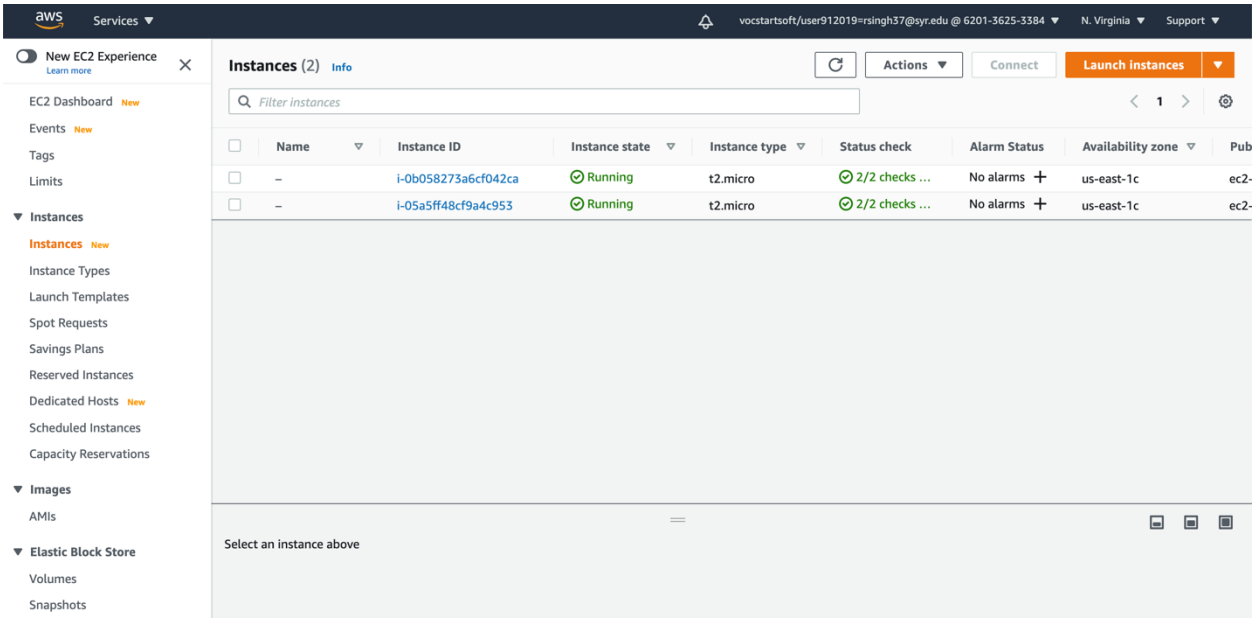


```
Downloads — ec2-user@ip-172-31-24-65:~ — ssh -i lab2keypair.pem ec2-user@ec2-107-23-187-88.compute-1.amazonaws.com — 203×56
  Verifying  : perl-TermReadKey-2.30-20.9.amzn1.x86_64                      3/8
  Verifying  : python26-2.6.9-2.92.amzn1.x86_64                             4/8
  Verifying  : 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-172-31-24-65 ~]$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.35.3/install.sh | bash
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 13527  100 13527    0     0   110k      0 --:--:-- --:--:-- --:--:--  109k
=> Downloading nvm from git to '/home/ec2-user/.nvm'
=> Cloning into '/home/ec2-user/.nvm'...
remote: Enumerating objects: 290, done.
remote: Counting objects: 100% (290/290), done.
remote: Compressing objects: 100% (257/257), done.
remote: Total 290 (delta 34), reused 110 (delta 20), pack-reused 0
Receiving objects: 100% (290/290), 163.37 KiB | 8.17 MiB/s, done.
Resolving deltas: 100% (34/34), done.
=> Compressing and cleaning up git repository

=> Appending nvm source string to /home/ec2-user/.bashrc
=> Appending bash_completion source string to /home/ec2-user/.bashrc
=> Close and reopen your terminal to start using nvm or run the following to use it now:

export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"  # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion"  # This loads nvm bash_completion
[ec2-user@ip-172-31-24-65 ~]$ export NVM_DIR="$HOME/.nvm"
[ec2-user@ip-172-31-24-65 ~]$ [ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"
[ec2-user@ip-172-31-24-65 ~]$ [ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion"
[ec2-user@ip-172-31-24-65 ~]$ nvm install node
Downloading and installing node v14.13.0...
Downloading https://nodejs.org/dist/v14.13.0/node-v14.13.0-linux-x64.tar.xz...
################################################################################ 100.0%
Computing checksum with sha256sum
Checksums matched!
Now using node v14.13.0 (npm v6.14.8)
Creating default alias: default -> node (-> v14.13.0)
[ec2-user@ip-172-31-24-65 ~]$ sudo yum -y install httpd
Loaded plugins: priorities, update-motd, upgrade-helper
Resolving Dependencies
--> Running transaction check
```



```
---> Package httpd.x86_64 0:2.2.34-1.16.amzn1 will be installed
--> Processing Dependency: httpd-tools = 2.2.34-1.16.amzn1 for package: httpd-2.2.34-1.16.amzn1.x86_64
--> Processing Dependency: apr-util-ldap for package: httpd-2.2.34-1.16.amzn1.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.2.34-1.16.amzn1.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.2.34-1.16.amzn1.x86_64
--> Running transaction check
---> Package apr.x86_64 0:1.5.2-5.13.amzn1 will be installed
---> Package apr-util.x86_64 0:1.5.4-6.18.amzn1 will be installed
---> Package apr-util-ldap.x86_64 0:1.5.4-6.18.amzn1 will be installed
---> Package httpd-tools.x86_64 0:2.2.34-1.16.amzn1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package             Arch         Version                 Repository       Size
================================================================================
Installing:
 httpd               x86_64       2.2.34-1.16.amzn1       amzn-main       1.2 M
Installing for dependencies:
 apr                 x86_64       1.5.2-5.13.amzn1        amzn-main       118 k
 apr-util            x86_64       1.5.4-6.18.amzn1        amzn-main        99 k
 apr-util-ldap       x86_64       1.5.4-6.18.amzn1        amzn-main        19 k
 httpd-tools         x86_64       2.2.34-1.16.amzn1       amzn-main        80 k

Transaction Summary
================================================================================
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-1.5.2-5.13.amzn1.x86_64.rpm                     | 118 kB   00:00
(3/5): httpd-tools-2.2.34-1.16.amzn1.x86_64.rpm            |  80 kB   00:00
(4/5): apr-util-1.5.4-6.18.amzn1.x86_64.rpm                |  99 kB   00:00
(5/5): httpd-2.2.34-1.16.amzn1.x86_64.rpm                  | 1.2 MB   00:00
--------------------------------------------------------------------------------
Total                                          1.9 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
```





2. Deploying the code

- **Run ML Code**

Once I connected to the EC2 instance through SSH in the terminal and clone the repository which has the code for the ML model and cd is used to change the directory and run the python code for Boston housing linear regression. I can see the contents of the directory using the ll command.

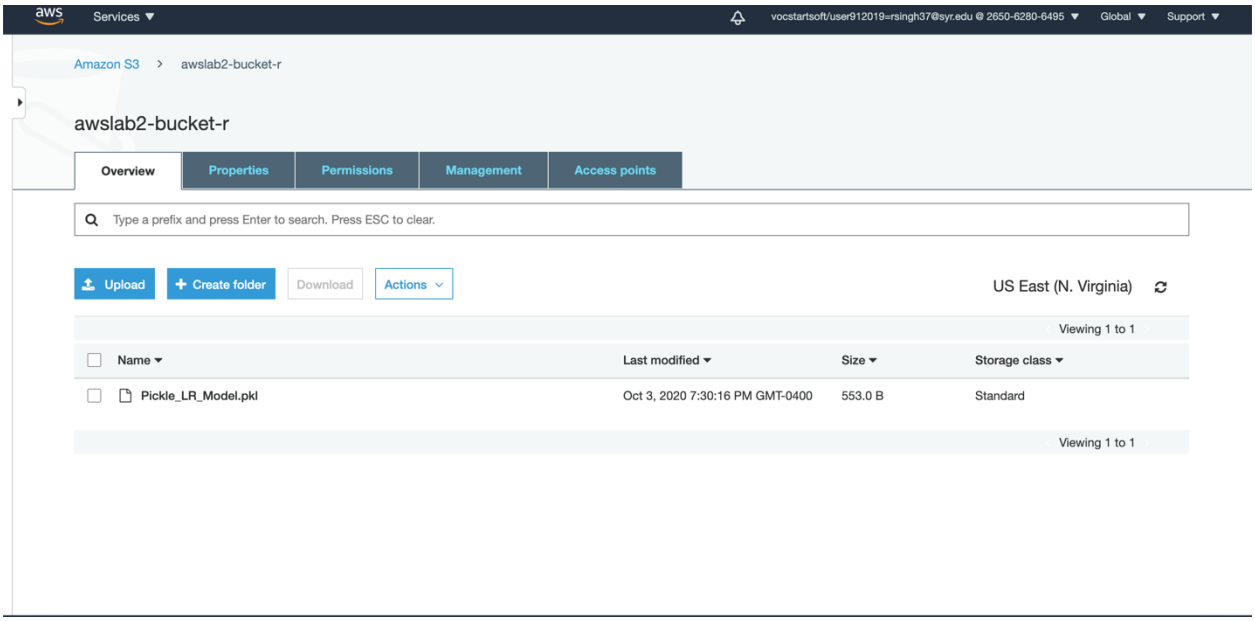
```
Complete!
[ec2-user@ip-172-31-28-90 ~]$ git clone https://github.com/ssingh60/AWSLab2MLCod
e.git master
Cloning into 'master'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 11 (delta 3), reused 2 (delta 0), pack-reused 0
Unpacking objects: 100% (11/11), done.
[ec2-user@ip-172-31-28-90 ~]$ cd master
[ec2-user@ip-172-31-28-90 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-172-31-28-90 master]$
```

```
RMSE is 5.137400784702911
[ec2-user@ip-172-31-28-90 master]$ ll
total 48
-rw-rw-r-- 1 ec2-user ec2-user 35734 Oct  2 23:56 BostonHousing.csv
-rw-rw-r-- 1 ec2-user ec2-user   979 Oct  2 23:56 BostonHousingLR.py
-rw-rw-r-- 1 ec2-user ec2-user   553 Oct  2 23:56 Pickle_LR_Model.pkl
-rw-rw-r-- 1 ec2-user ec2-user    15 Oct  2 23:56 README.md
[ec2-user@ip-172-31-28-90 master]$
```

Next I copy the contents from the EC2 instance to the S3 bucket using the command **"aws s3 cp Pickle_LR_Model.pkl s3://awslab2-bucket-r"** with the file name and the bucket location.

```
Complete!
[ec2-user@ip-172-31-31-43 ~]$ 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-172-31-31-43 ~]$ cd master
[ec2-user@ip-172-31-31-43 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-172-31-31-43 master]$ ll
total 48
-rw-rw-r-- 1 ec2-user ec2-user 35734 Oct  3 23:26 BostonHousing.csv
-rw-rw-r-- 1 ec2-user ec2-user   979 Oct  3 23:26 BostonHousingLR.py
-rw-rw-r-- 1 ec2-user ec2-user   553 Oct  3 23:26 Pickle_LR_Model.pkl
-rw-rw-r-- 1 ec2-user ec2-user    15 Oct  3 23:26 README.md
[ec2-user@ip-172-31-31-43 master]$ aws s3 ls
2020-10-03 23:20:11 awslab2-bucket-r
2020-09-30 15:55:27 elasticbeanstalk-us-east-1-265062806495
[ec2-user@ip-172-31-31-43 master]$ aws s3 ls s3://awslab2-bucket-r
[ec2-user@ip-172-31-31-43 master]$ aws s3 cp Pickle_LR_Model.pkl s3://awslab2-bucket-r
upload: ./Pickle_LR_Model.pkl to s3://awslab2-bucket-r/Pickle_LR_Model.pkl
[ec2-user@ip-172-31-31-43 master]$
```

As seen in the snippet, the Pickle file gets uploaded to the S3 bucket.



Then, I stop the EC2 instance.

- **Run Flask and Angular Code**
  - **Run Flask Code**

    Flask allows you to deploy web application framework to scale up complex applications. It has the tools, libraries and technologies to build web application. First, I establish the connection to the second instance using SSH and use the git clone command to clone the code containing the Flask code and change the directory to the location of the code.

```
up to date in 0.222s
found 0 vulnerabilities

[ec2-user@ip-172-31-24-65 ~]$ 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-172-31-24-65 ~]$ cd AWSLab2FlaskCode
[ec2-user@ip-172-31-24-65 AWSLab2FlaskCode]$ ll
total 4
-rw-rw-r-- 1 ec2-user ec2-user 642 Oct  3 00:08 modelFlask.py
[ec2-user@ip-172-31-24-65 AWSLab2FlaskCode]$
```

Then I use the command to copy the Pickle file from S3 to the instance using the command **"aws s3 cp s3://awslab2-bucket-r/Pickle_LR_Model.pkl."** The snippet shows the contents of the instance which we have copied from s3.

```
[ec2-user@ip-172-31-55-127 AWSLab2FlaskCode]$ aws s3 cp s3://awslab2-bucket-r/Pickle_LR_Model.pkl Pickle_LR_Model.pkl
download: s3://awslab2-bucket-r/Pickle_LR_Model.pkl to ./Pickle_LR_Model.pkl
[ec2-user@ip-172-31-55-127 AWSLab2FlaskCode]$ ll
total 8
-rw-rw-r-- 1 ec2-user ec2-user 642 Oct  3 23:44 modelFlask.py
-rw-rw-r-- 1 ec2-user ec2-user 553 Oct  3 23:30 Pickle_LR_Model.pkl
[ec2-user@ip-172-31-55-127 AWSLab2FlaskCode]$
```

Then I run the “nohup python3 modelFlask.py &” to start Flask App in the background to deploy the code in the front end.

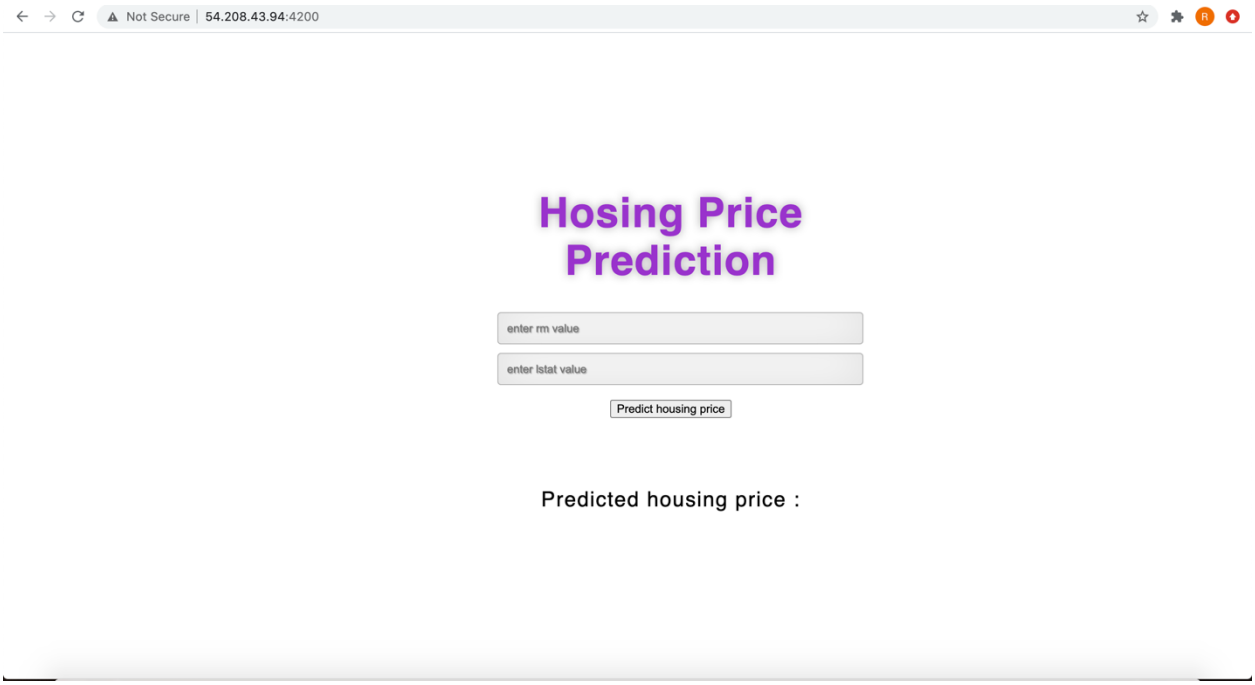
- **Run Angular Code**

AngularJS is designed for building web applications which extends HTML attributes and binds data. First, I cloned the repository which has the code for the Angular UI and our code is calling the Flask API to give the output. The IP address of instances are different we change the file “web.service.ts” where we can update the IPv4 address of our instance.

The Angular code is updated to call the Flask API and I executed the “npm install” command on the instance to install all the dependencies of node package manager. It installs all the modules listed as dependencies. Then I run “ng serve –host 0.0.0.0 –port 4200” which will start a server a local host 4200. 4200 is the default port chosen by the Angular CLI.

```
> 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/AWSLab2FlaskCode/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 36.254s

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-172-31-55-127 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: 2020-10-04T00:03:32.226Z - Hash: 378ff1220988286ceb2f - Time: 15437ms
** Angular Live Development Server is listening on 0.0.0.0:4200, open your browser on http://localhost:4200/ **
: Compiled successfully.
```

Next, I used IPv4 address of EC2 instance with the port 4200 in my browser to see the Angular UI. All the backend is being done on the cloud AWS. The snippet shows the UI shows to enter the values.



Then, I enter the values which shows the predicted values for the houses. Later, I stop the instance.