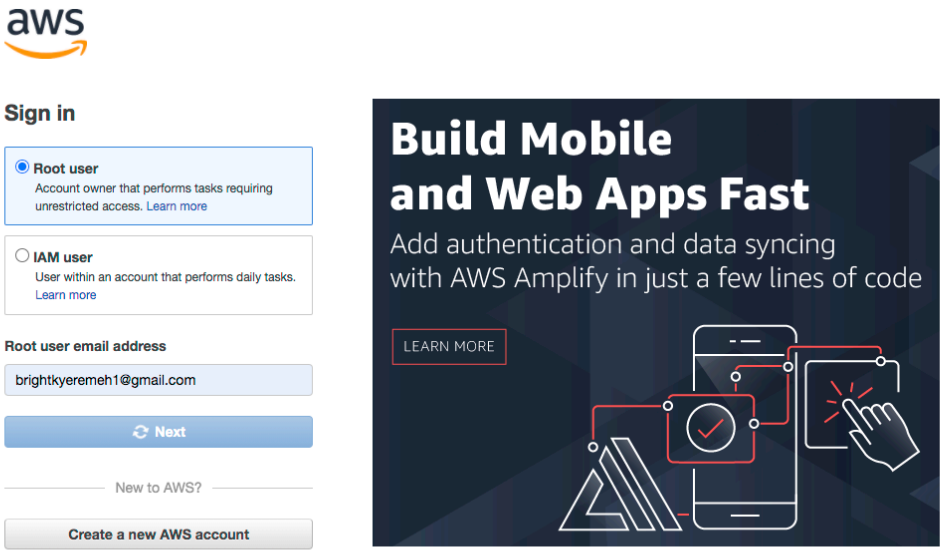
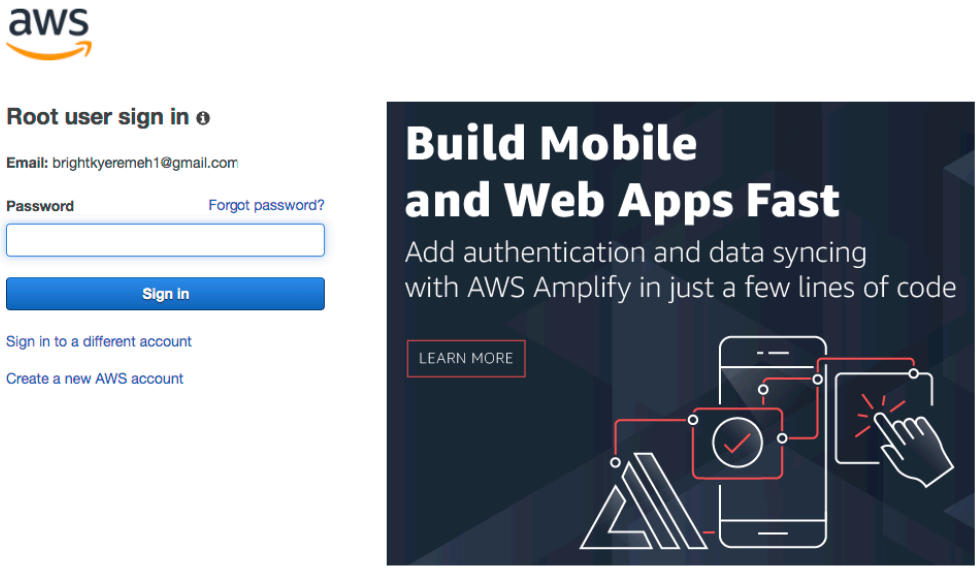
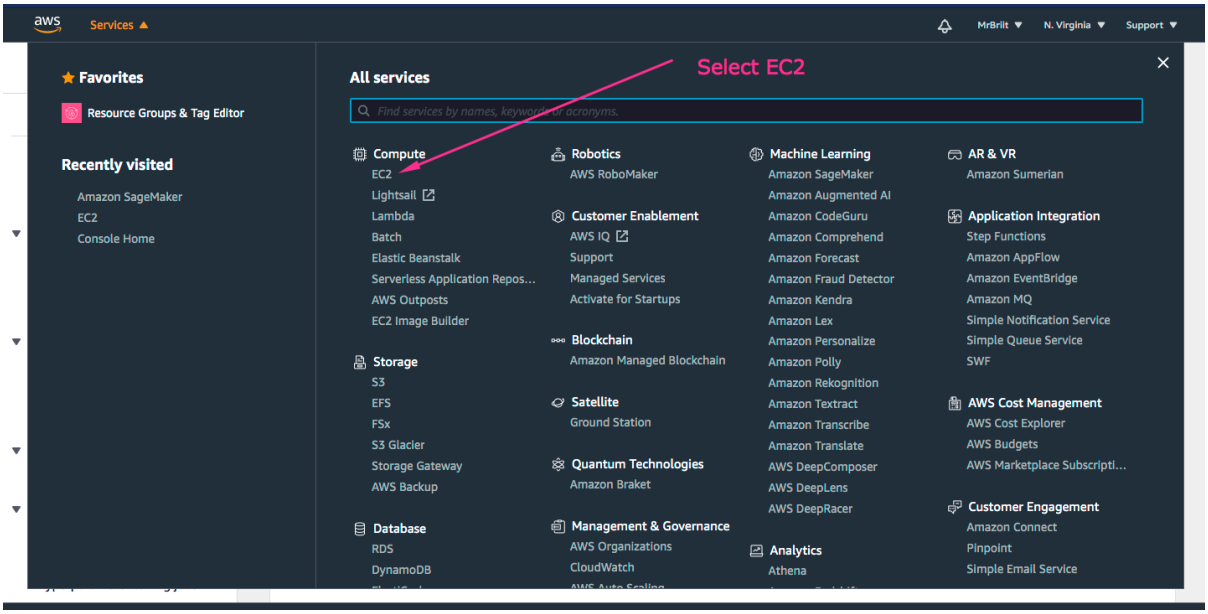
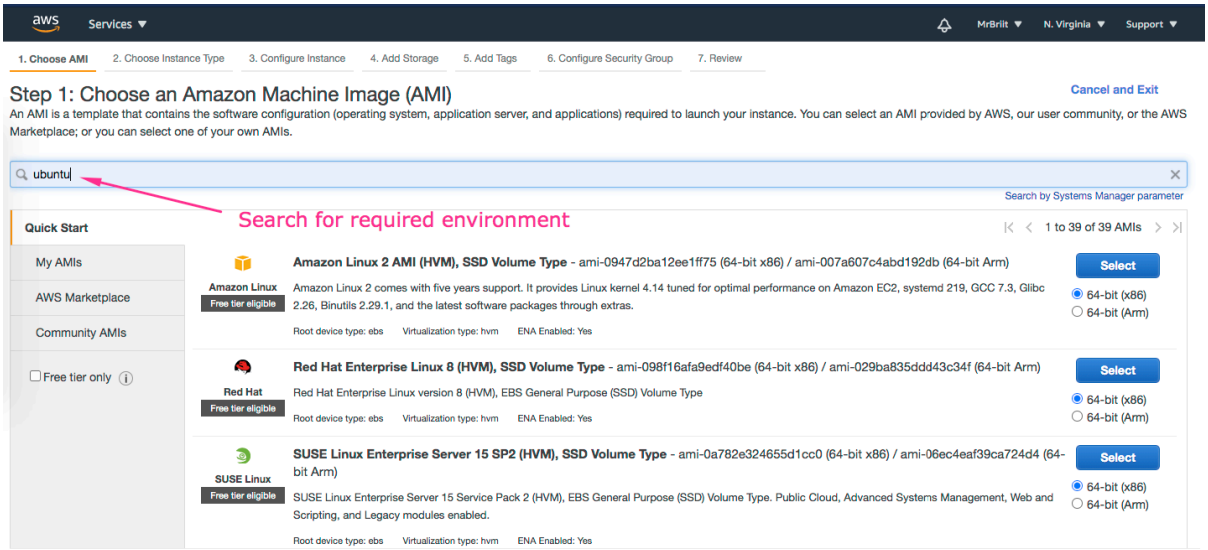
AWS DEPLOYMENT

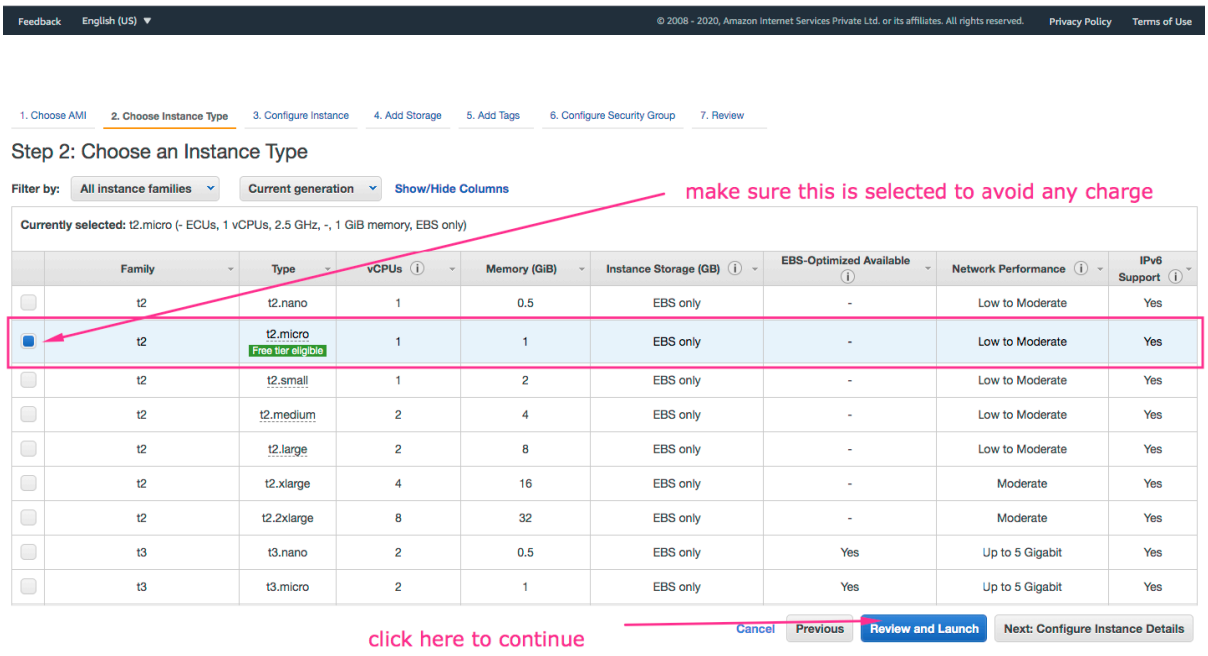
- PART-1: Model Building and hosting local API
- PART - 2 : Deploying Public API to AWS EC2 server and launch website service

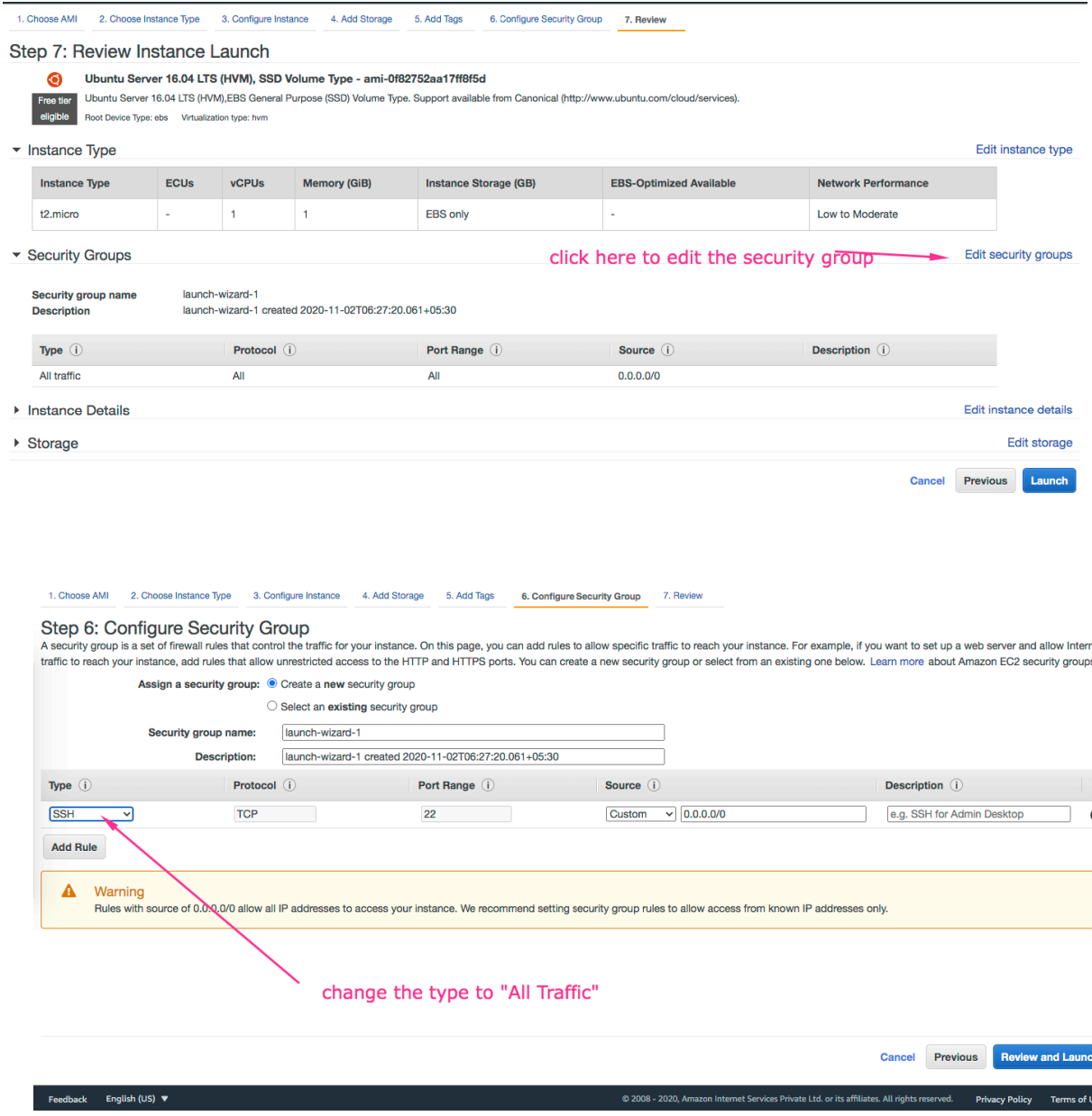




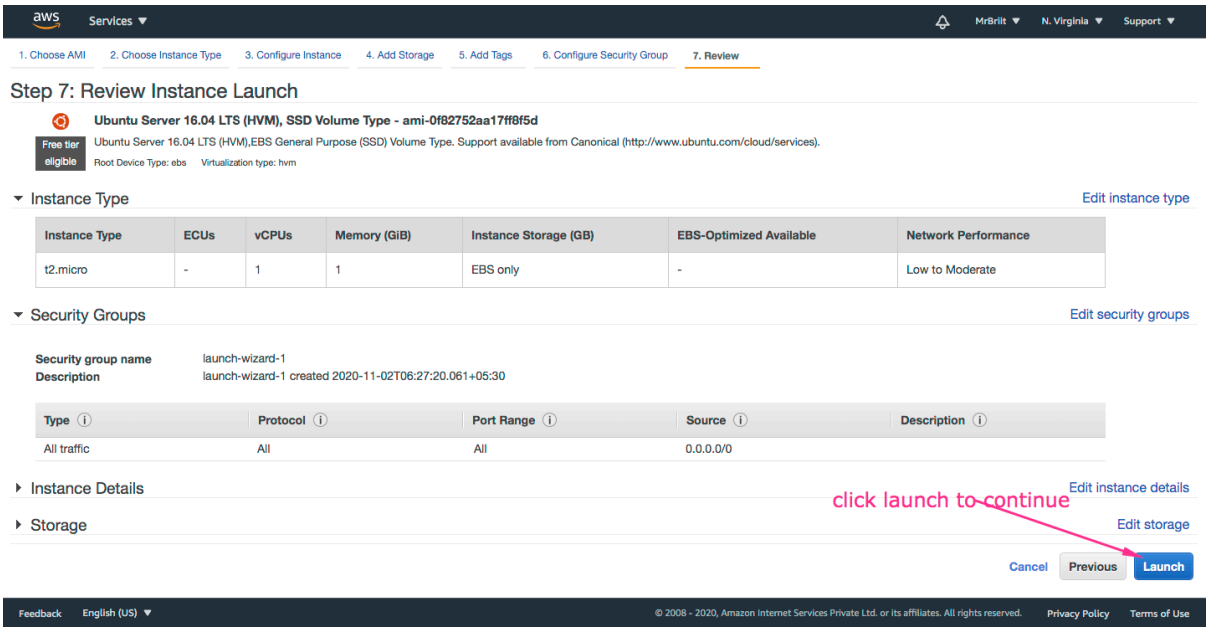

Services
MrBrit
N. Virginia
Support
Favorites
Resource Groups & Tag Editor
Recently visited
Amazon SageMaker
EC2
Console Home
All services
Select EC2
Compute
EC2
Lightsail
Lambda
Batch
Elastic Beanstalk
Serverless Application Repos...
AWS Outposts
EC2 Image Builder
Storage
S3
EFS
FSx
S3 Glacier
Storage Gateway
AWS Backup
Database
RDS
DynamoDB
Robotics
AWS RoboMaker
Customer Enablement
AWS IQ
Support
Managed Services
Activate for Startups
Blockchain
Amazon Managed Blockchain
Satellite
Ground Station
Quantum Technologies
Amazon Braket
Management & Governance
AWS Organizations
CloudWatch
Machine Learning
Amazon SageMaker
Amazon Augmented AI
Amazon CodeGuru
Amazon Comprehend
Amazon Forecast
Amazon Fraud Detector
Amazon Kendra
Amazon Lex
Amazon Personalize
Amazon Polly
Amazon Rekognition
Amazon Textract
Amazon Transcribe
Amazon Translate
AWS DeepComposer
AWS DeepLens
AWS DeepRacer
Analytics
Athena
AR & VR
Amazon Sumerian
Application Integration
Step Functions
Amazon AppFlow
Amazon EventBridge
Amazon MQ
Simple Notification Service
Simple Queue Service
SWF
AWS Cost Management
AWS Cost Explorer
AWS Budgets
AWS Marketplace Subscripti...
Customer Engagement
Amazon Connect
Pinpoint
Simple Email Service


Services
MrBrit
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 1: Choose an Amazon Machine Image (AMI)
Cancel and Exit
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.
ubuntu
Search by Systems Manager parameter
Search for required environment
Quick Start
My AMIs
AWS Marketplace
Community AMIs
Free tier only
1 to 39 of 39 AMIs
Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0947d2ba12ee1ff75 (64-bit x86) / ami-007a607c4abd192db (64-bit Arm)
Amazon Linux
Free tier eligible
Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.
Root device type: ebs
Virtualization type: hvm
ENA Enabled: Yes
Select
64-bit (x86)
64-bit (Arm)
Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-098f16afa9edf40be (64-bit x86) / ami-029ba835ddd43c34f (64-bit Arm)
Red Hat
Free tier eligible
Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type
Root device type: ebs
Virtualization type: hvm
ENA Enabled: Yes
Select
64-bit (x86)
64-bit (Arm)
SUSE Linux Enterprise Server 15 SP2 (HVM), SSD Volume Type - ami-0a782e324655d1cc0 (64-bit x86) / ami-06ec4eaf39ca724d4 (64-bit Arm)
SUSE Linux
Free tier eligible
SUSE Linux Enterprise Server 15 Service Pack 2 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.
Root device type: ebs
Virtualization type: hvm
ENA Enabled: Yes
Select
64-bit (x86)
64-bit (Arm)

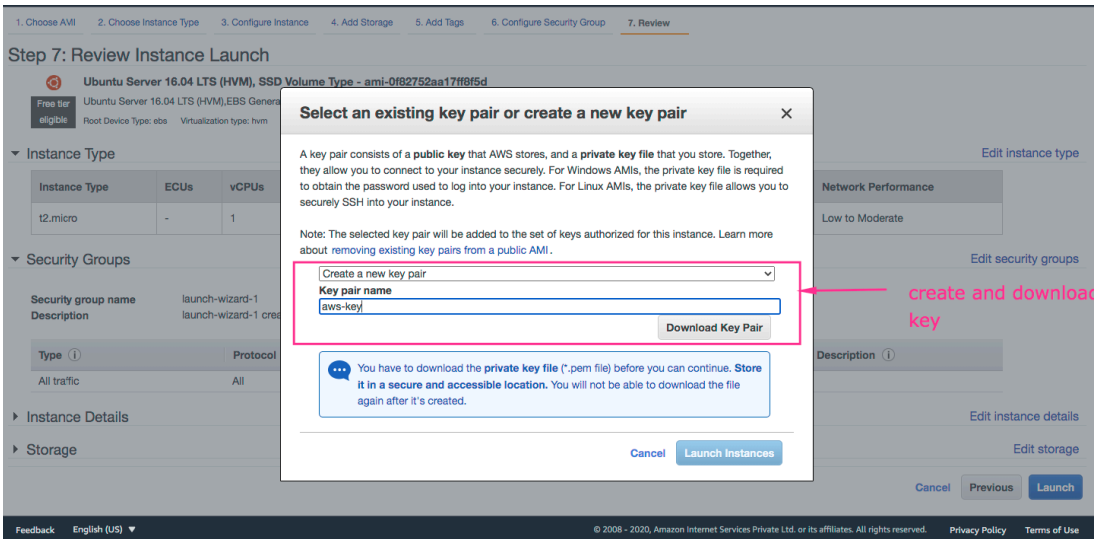

Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 2: Choose an Instance Type
Filter by:
All instance families
Current generation
Show/Hide Columns
make sure this is selected to avoid any charge
Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)
Family
Type
vCPUs
Memory (GiB)
Instance Storage (GB)
EBS-Optimized Available
Network Performance
IPv6 Support
t2 t2.nano 1 0.5 EBS only - Low to Moderate Yes
t2 t2.micro Free tier eligible 1 1 EBS only - Low to Moderate Yes
t2 t2.small 1 2 EBS only - Low to Moderate Yes
t2 t2.medium 2 4 EBS only - Low to Moderate Yes
t2 t2.large 2 8 EBS only - Low to Moderate Yes
t2 t2.xlarge 4 16 EBS only - Moderate Yes
t2 t2.2xlarge 8 32 EBS only - Moderate Yes
t3 t3.nano 2 0.5 EBS only Yes Up to 5 Gigabit Yes
t3 t3.micro 2 1 EBS only Yes Up to 5 Gigabit Yes
click here to continue
Cancel
Previous
Review and Launch
Next: Configure Instance Details


1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 7: Review Instance Launch
Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-0f82752aa17ff8f5d
Free tier eligible
Ubuntu Server 16.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Root Device Type: ebs
Virtualization type: hvm
Instance Type
Edit instance type
Instance Type | ECUs | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance
t2.micro | - | 1 | 1 | EBS only | - | Low to Moderate
Security Groups
click here to edit the security group
Edit security groups
Security group name launch-wizard-1
Description launch-wizard-1 created 2020-11-02T06:27:20.061+05:30
Type | Protocol | Port Range | Source | Description
All traffic | All | All | 0.0.0.0/0
Instance Details
Edit instance details
Storage
Edit storage
Cancel
Previous
Launch

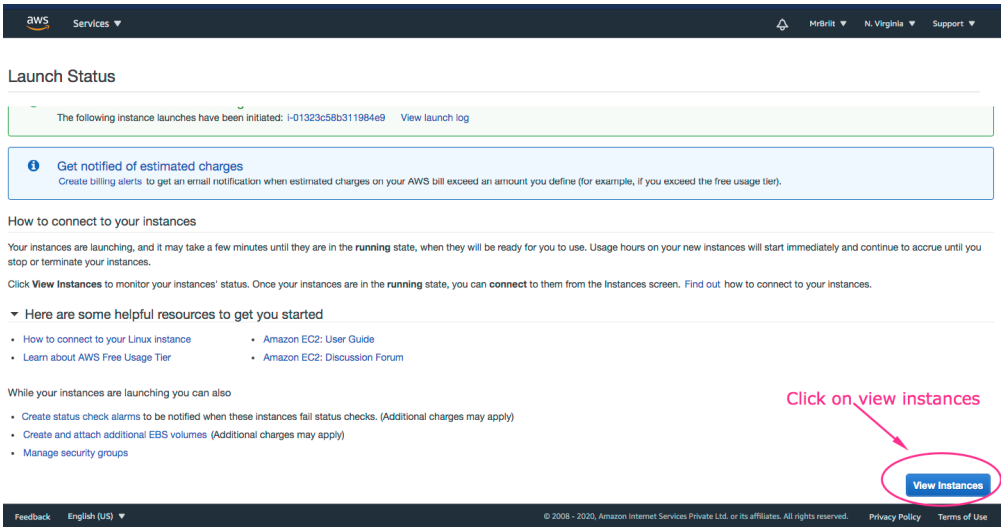
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 6: Configure Security Group
A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic 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 name: launch-wizard-1
Description: launch-wizard-1 created 2020-11-02T06:27:20.061+05:30
Type | Protocol | Port Range | Source | Description
SSH | TCP | 22 | Custom | 0.0.0.0/0 | e.g. SSH for Admin Desktop
Add Rule
Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.
change the type to "All Traffic"
Cancel
Previous
Review and Launch
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


aws
Services
MrBrilt
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 7: Review Instance Launch
Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-0f82752aa17ff8f5d
Free tier eligible
Ubuntu Server 16.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Root Device Type: ebs
Virtualization type: hvm
Instance Type
Edit instance type
Instance Type | ECUs | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance
t2.micro | - | 1 | 1 | EBS only | - | Low to Moderate
Security Groups
Edit security groups
Security group name launch-wizard-1
Description launch-wizard-1 created 2020-11-02T06:27:20.061+05:30
Type | Protocol | Port Range | Source | Description
All traffic | All | All | 0.0.0.0/0
Instance Details
Edit instance details
click launch to continue
Storage
Edit storage
Cancel
Previous
Launch
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

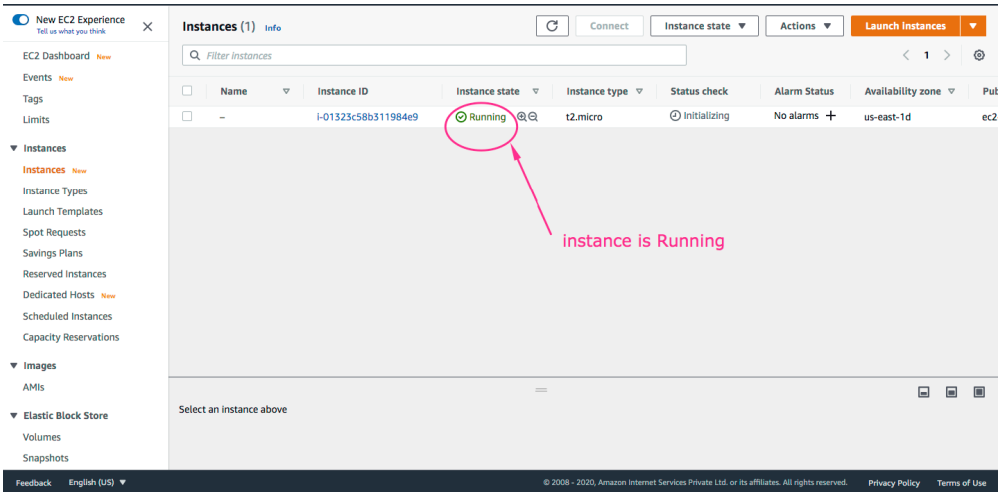
You need to create a Private Key so that only you can login to the server using SHH protocol



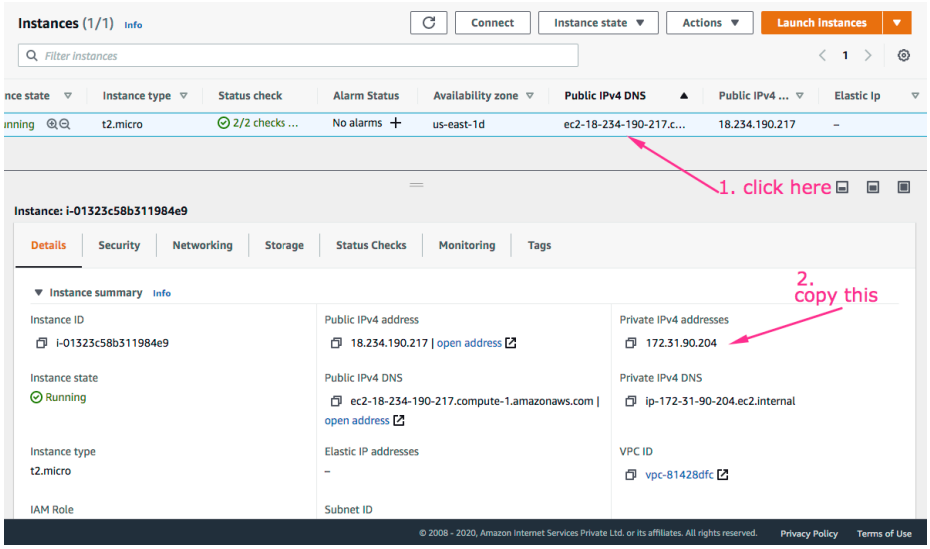
Instance Launched



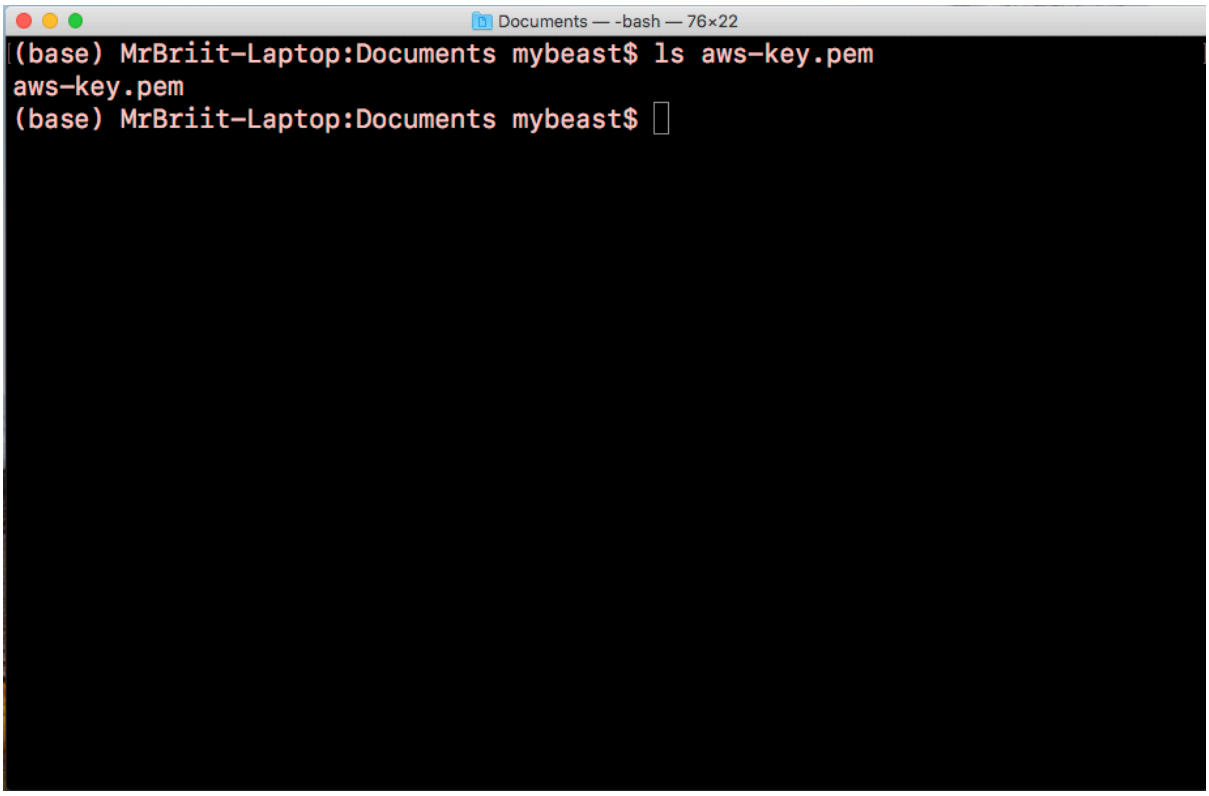
Instance is Running



You can copy your public key for later use



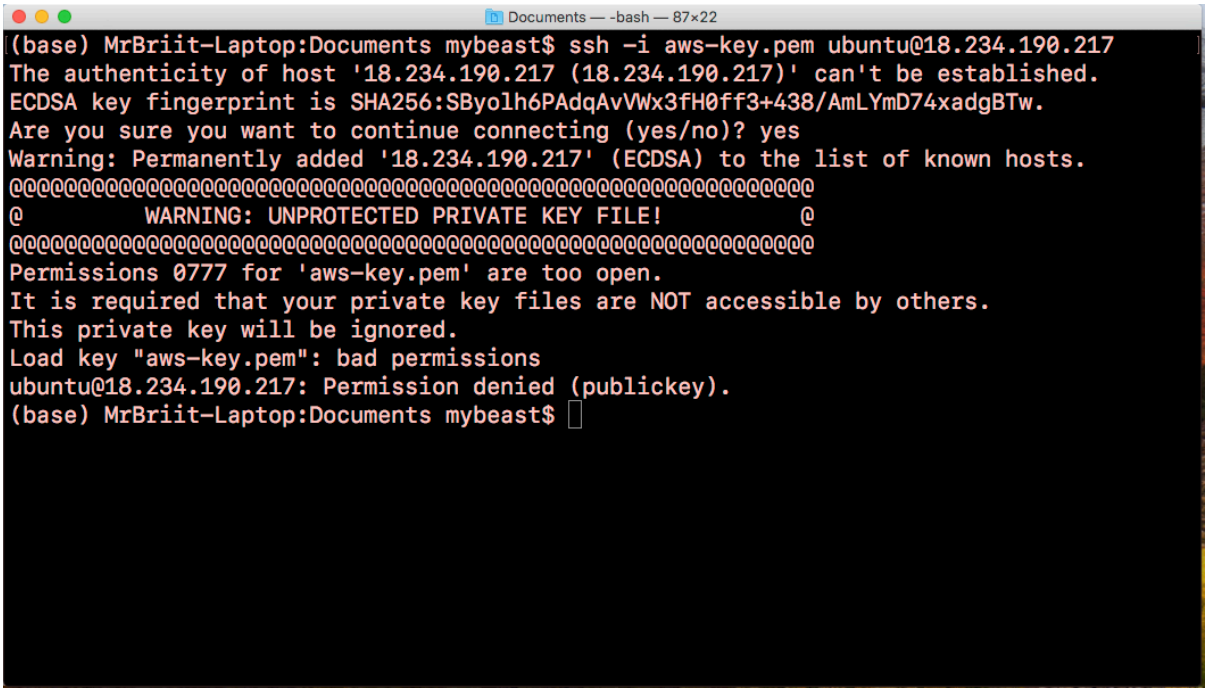
Go to Terminal on Mac or Command Prompt on Windows. Make sure your key is in your Present Working Directory



Type the command ssh -i aws-key.pem ubuntu@(add your Private Private IPv4 addresses)

e.g. *ssh -i aws-key.pem ubuntu@172.31.90.204*

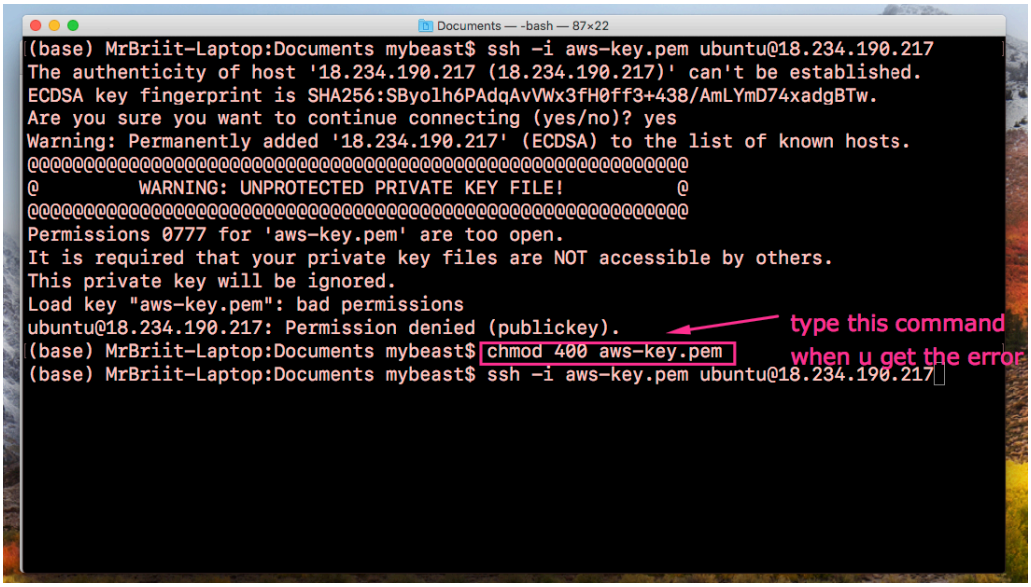
You might run into the below error

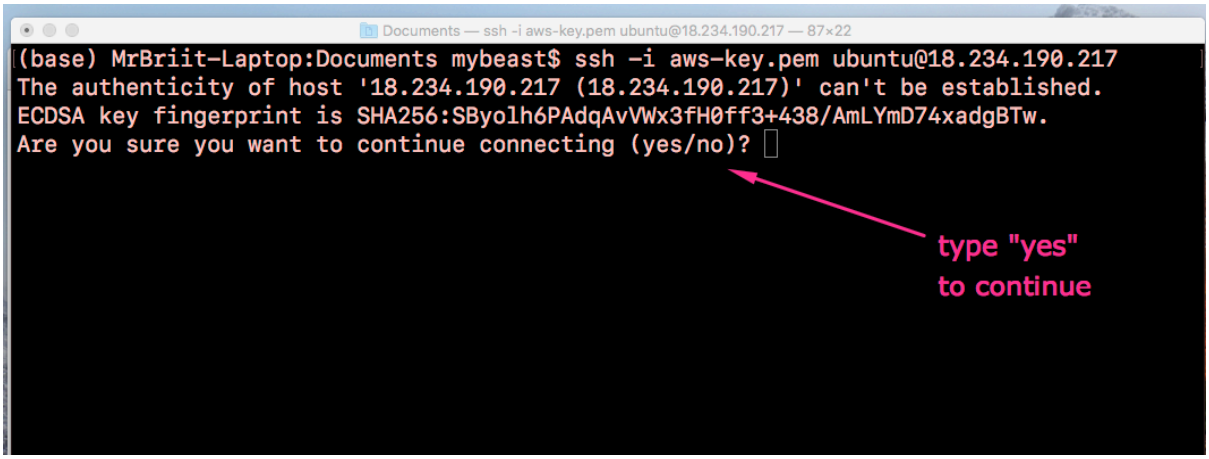


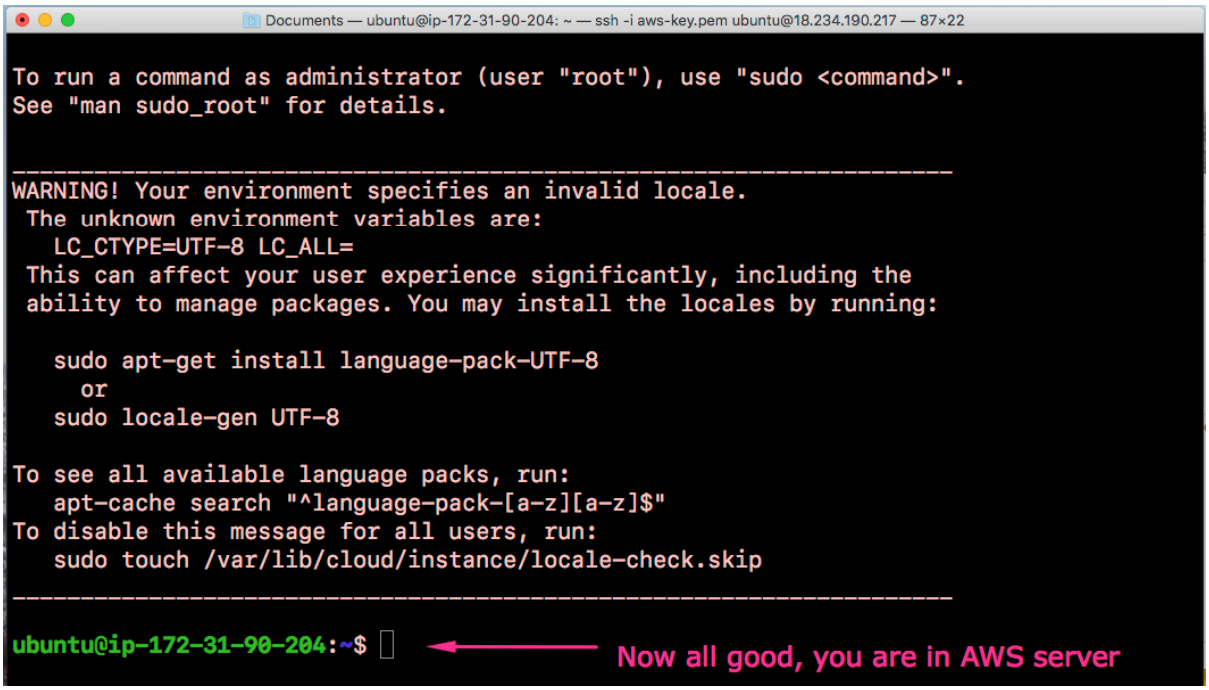
Type *chmod 400 aws-key.pem* to get rid of the error.

Then again type the command *ssh -i aws-key.pem ubuntu@(add your Private Private IPv4 addresses)*

e.g. ssh -i aws-key.pem ubuntu@172.31.90.204







Let's check few stuff.

1. Check to see if you have Python 3 installed: *python3 -V*

```
ubuntu@ip-172-31-90-204:~$ python3 -V
Python 3.5.2
ubuntu@ip-172-31-90-204:~$
```

2. Let's update all the existing packages: *sudo apt-get update*

3. Check if pip is installed, else install it. *pip*

```
ubuntu@ip-172-31-90-204:~$ pip
The program 'pip' is currently not installed. You can install it by typing:
sudo apt install python-pip
ubuntu@ip-172-31-90-204:~$
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial/universe amd64 Packages [7532 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial/universe Translation-en [4354 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial/multiverse amd64 Packages [144 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial/multiverse Translation-en [106 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages [1880 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/main Translation-en [454 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/restricted amd64 Packages [10.2 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 Packages [1195 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/universe Translation-en [348 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/multiverse amd64 Packages [23.0 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/multiverse Translation-en [8632 B]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports/main amd64 Packages [9812 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports/main Translation-en [4456 B]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports/universe amd64 Packages [11.3 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports/universe Translation-en [4476 B]
Get:20 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [1476 kB]
Get:21 http://security.ubuntu.com/ubuntu xenial-security/main Translation-en [350 kB]
Get:22 http://security.ubuntu.com/ubuntu xenial-security/restricted amd64 Packages [9824 B]
Get:23 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 Packages [773 kB]
Get:24 http://security.ubuntu.com/ubuntu xenial-security/universe Translation-en [218 kB]
Get:25 http://security.ubuntu.com/ubuntu xenial-security/multiverse amd64 Packages [8236 B]
Get:26 http://security.ubuntu.com/ubuntu xenial-security/multiverse Translation-en [2888 B]
Fetched 19.2 MB in 3s (4864 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-90-204:~$
```

3.1 If pip is not installed, *d*ownload pip: *curl -O* https://bootstrap.pypa.io/get-pip.py

3.2 Update

```
ubuntu@ip-172-31-90-204:~$ sudo apt-get update
Hit:1 http://security.ubuntu.com/ubuntu xenial-security InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports InReleas
e
Reading package lists... Done
ubuntu@ip-172-31-90-204:~$
```

3.3 Install pip: sudo python3 get-pip.py

```
Hit:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports InReleas
e
Reading package lists... Done
ubuntu@ip-172-31-90-204:~$ sudo python3 get-pip.py
DEPRECATION: Python 3.5 reached the end of its life on September 13th, 2020. P
lease 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.
WARNING: The directory '/home/ubuntu/.cache/pip' or its parent directory is no
t owned or is not writable by the current user. The cache has been disabled. C
heck the permissions and owner of that directory. If executing pip with sudo,
you may want sudo's -H flag.
Collecting pip
  Downloading pip-20.2.4-py2.py3-none-any.whl (1.5 MB)
     |################################| 1.5 MB 19.9 MB/s
Collecting setuptools
  Downloading setuptools-50.3.2-py3-none-any.whl (785 kB)
     |################################| 785 kB 35.2 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-20.2.4 setuptools-50.3.2 wheel-0.35.1
ubuntu@ip-172-31-90-204:~$
```

3.4 Check if pip is working: *pip*

4. Install Flask: *sudo pip install flask*

4.1 Install the following as well

sudo pip install flask_cors

sudo apt-get install apache2

sudo pip install sklearn

sudo apt-get install libapache2-mod-wsgi-py3

5. Let's do some configurations: *sudo vi /etc/apache2/sites-enabled/000-default.conf*
Copy and paste the following as shown in the video:

```
DocumentRoot /home/ubuntu/mlapp
WSGIDaemonProcess flaskapp threads=5 python-home=/usr/local/lib/python3.5/site-packages/ user=ubuntu
    WSGIScriptAlias / /home/ubuntu/mlapp/flaskapp.wsgi
<Directory /home/ubuntu/mlapp>
        WSGIProcessGroup flaskapp
        WSGIApplicationGroup %{GLOBAL}
        Require all granted
    </Directory>
```

5.1 Now press ***escape(esc)*** followed by *:wq!* then press **enter** to exit

6. Create file flaskapp.wsgi at mlapp directory with content below

The Web Server Gateway(wsgi) Interface is a simple calling convention for web servers to forward requests to web applications

Steps:

6.1. first create a directory: *mkdir mlapp*

6.2 cd to the directory: *cd mlapp/*

6.3 vi flaskapp.wsgi

6.4 Copy and paste the following as shown in the video

```
import sys
import site
site.addsitedir('/home/ubuntu/.local/lib/python3.5/site-packages')
sys.path.insert(0, '/home/ubuntu/mlapp')
from app import app as application
```

7. Make sure you have saved your model and app.py file

Move your files to AWS

NB: do this from a new Terminal or Command Prompt

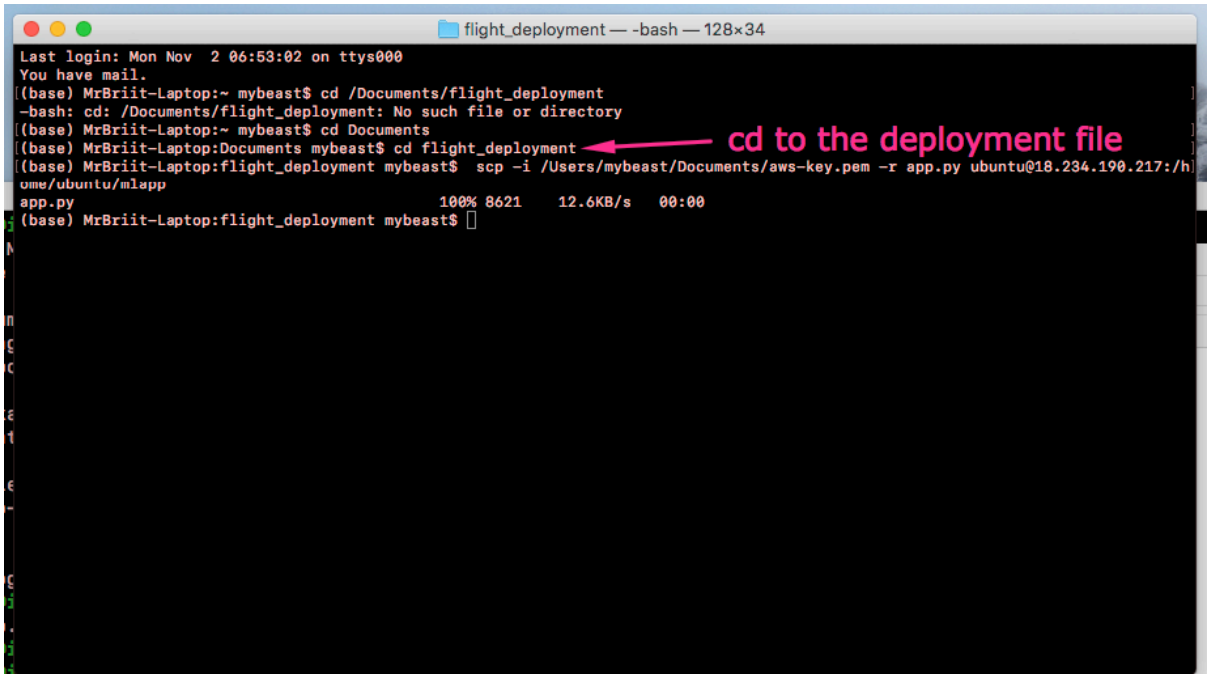
7.1. cd to the deployment folder

Then type: *scp -i (path to was key) -r app.py ubuntu@(add your public key here):/home/ubuntu/mlapp*

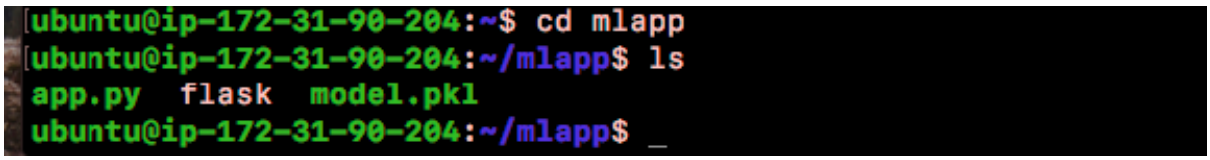
e.g. *scp -i /Users/mybeast/Documents/aws-key.pem -r **app.py** ubuntu@18.234.190.217:/home/ubuntu/mlapp*

*scp -i /Users/mybeast/Documents/aws-key.pem -r **model.pkl** ubuntu@18.234.190.217:/home/ubuntu/mlapp*

NB: You can use ***winscp*** on ***Windows*** to move the files instead of ***scp(which is for Mac)***

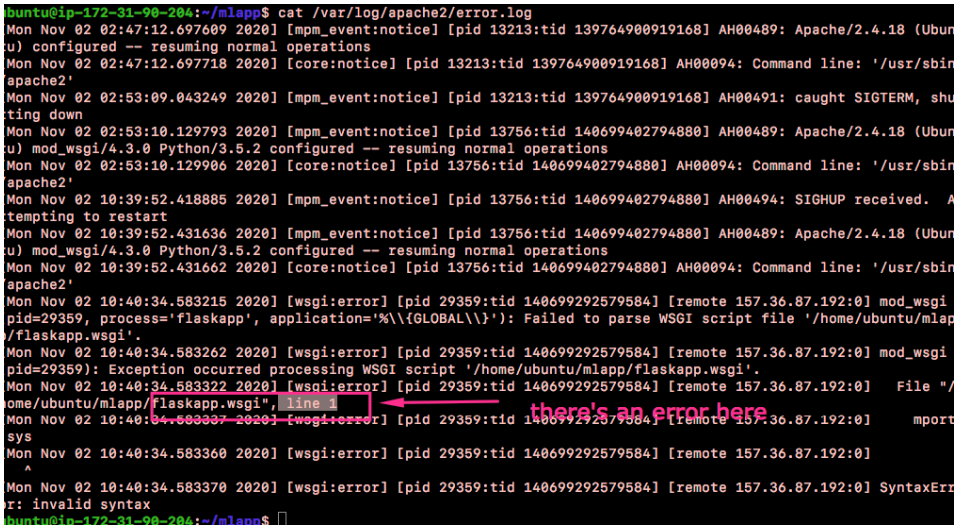


7. 2 Confirm to see if you have all files: *ls*



8. Restart the server: *sudo apachectl restart*

8.1 Check log incase you run into any error: *cat /var/log/apache2/error.log*



8.2 Use *vi (File Name).* to see the error and fix it.

e.g vi flaskapp.wsgi

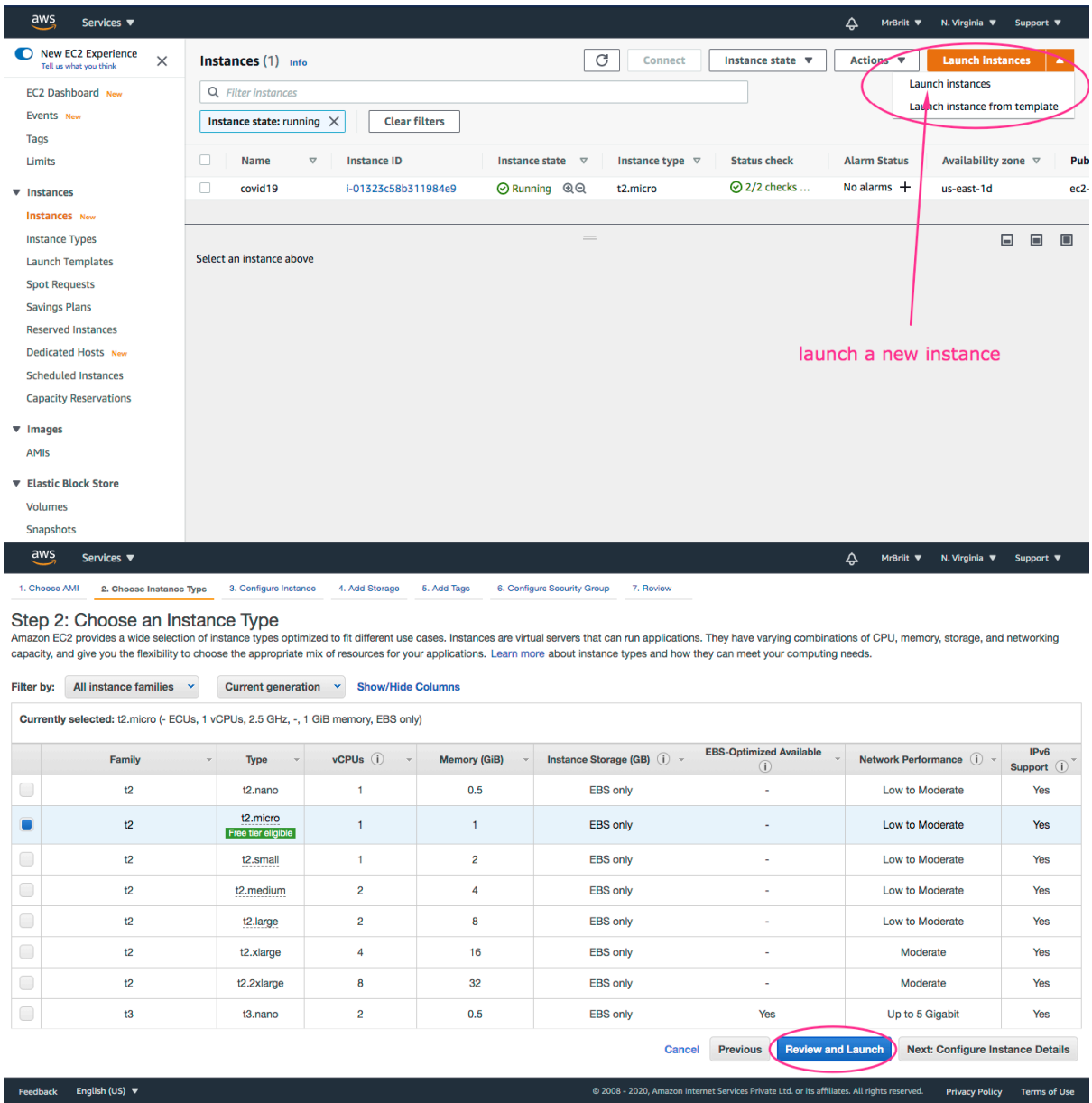
8.3.3 restart the server again

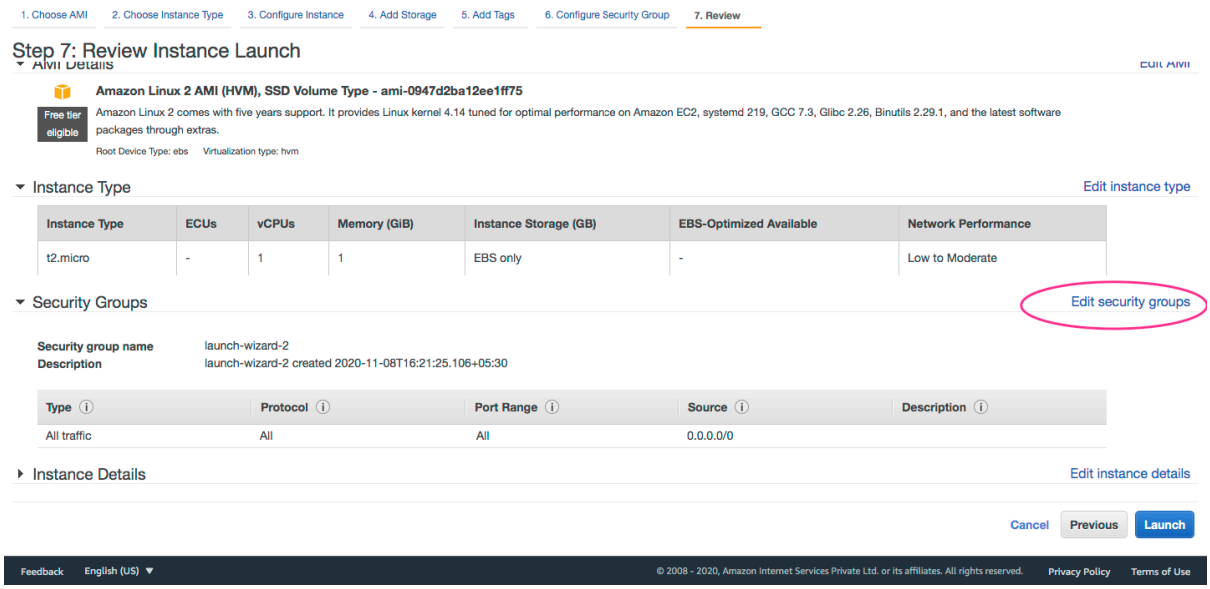
9. Test your App by copying and pasting your public key (e.g. 157.36.87.192) into your browser

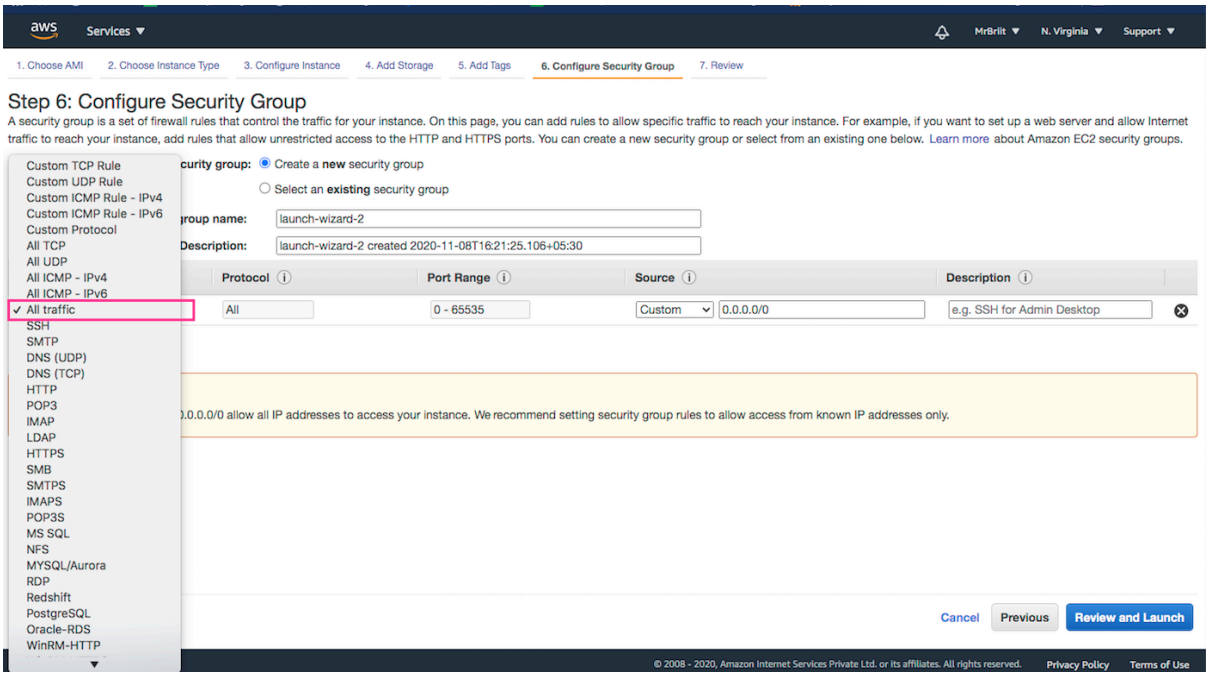
PART 2

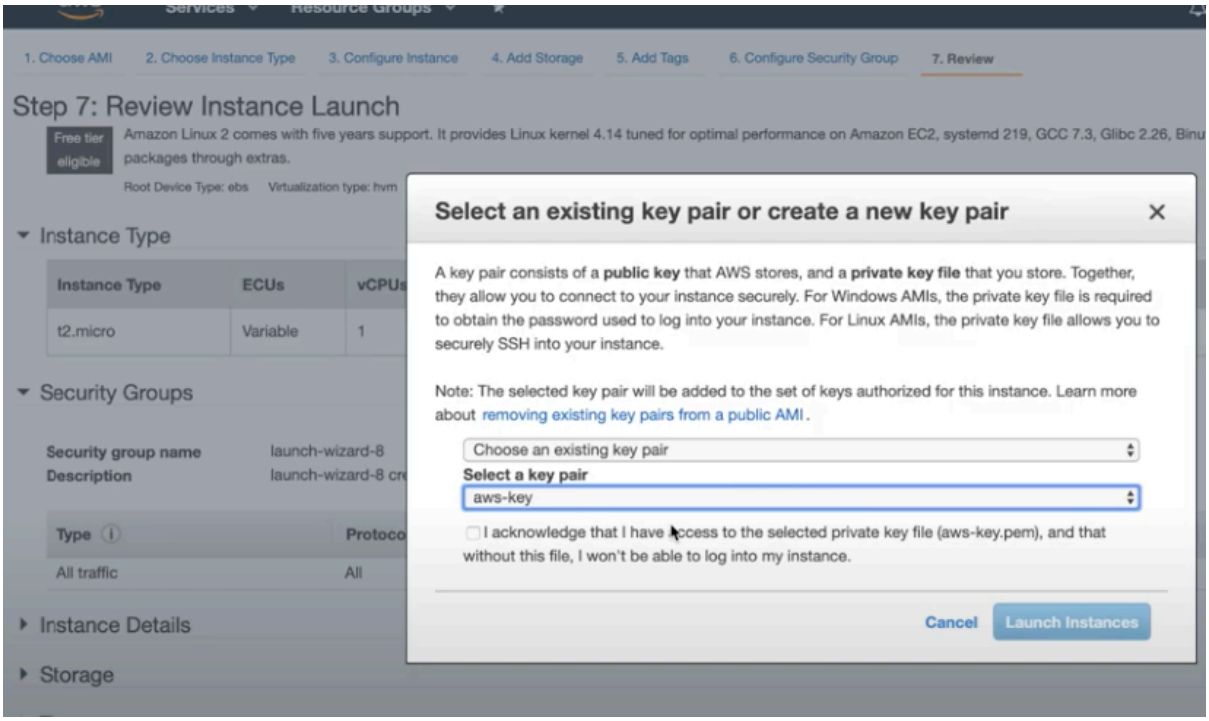
We have so far run it from localhost, let's see how we can make configure it well on AWS.

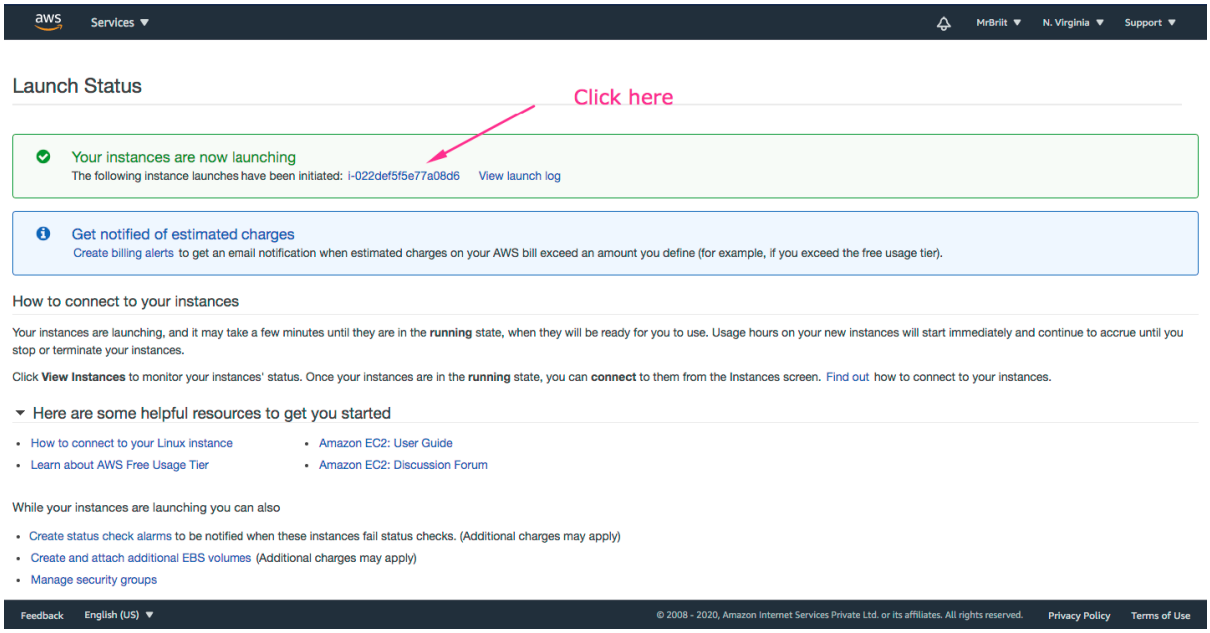
Let's create another instance, just like we did previously.




1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 7: Review Instance Launch
Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0947d2ba12ee1ff75
Free tier eligible
Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.
Root Device Type: ebs
Virtualization type: hvm
Instance Type
Edit instance type
Instance Type
ECUs
vCPUs
Memory (GiB)
Instance Storage (GB)
EBS-Optimized Available
Network Performance
t2.micro
1
1
EBS only
Low to Moderate
Security Groups
Edit security groups
Security group name
launch-wizard-2
Description
launch-wizard-2 created 2020-11-08T16:21:25.106+05:30
Type
Protocol
Port Range
Source
Description
All traffic
All
All
0.0.0.0/0
Instance Details
Edit instance details
Cancel
Previous
Launch
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


aws
Services
MrBrilt
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 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.
Create a new security group
Select an existing security group
launch-wizard-2
launch-wizard-2 created 2020-11-08T16:21:25.106+05:30
Protocol
Port Range
Source
Description
All
0 - 65535
Custom
0.0.0.0/0
e.g. SSH for Admin Desktop
Custom TCP Rule
Custom UDP Rule
Custom ICMP Rule - IPv4
Custom ICMP Rule - IPv6
Custom Protocol
All TCP
All UDP
All ICMP - IPv4
All ICMP - IPv6
All traffic
SSH
SMTP
DNS (UDP)
DNS (TCP)
HTTP
POP3
IMAP
LDAP
HTTPS
SMB
SMTPS
IMAPS
POP3S
MS SQL
NFS
MYSQL/Aurora
RDP
Redshift
PostgreSQL
Oracle-RDS
WinRM-HTTP
allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.
Cancel
Previous
Review and Launch
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 7: Review Instance Launch
Free tier eligible
Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binu
packages through extras.
Root Device Type: ebs
Virtualization type: hvm
Instance Type
Instance Type
ECUs
vCPUs
t2.micro
Variable
1
Security Groups
Security group name
launch-wizard-8
Description
launch-wizard-8 cr
Type
Protoco
All traffic
All
Instance Details
Storage
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
aws-key
I acknowledge that I have access to the selected private key file (aws-key.pem), and that without this file, I won't be able to log into my instance.
Cancel
Launch Instances

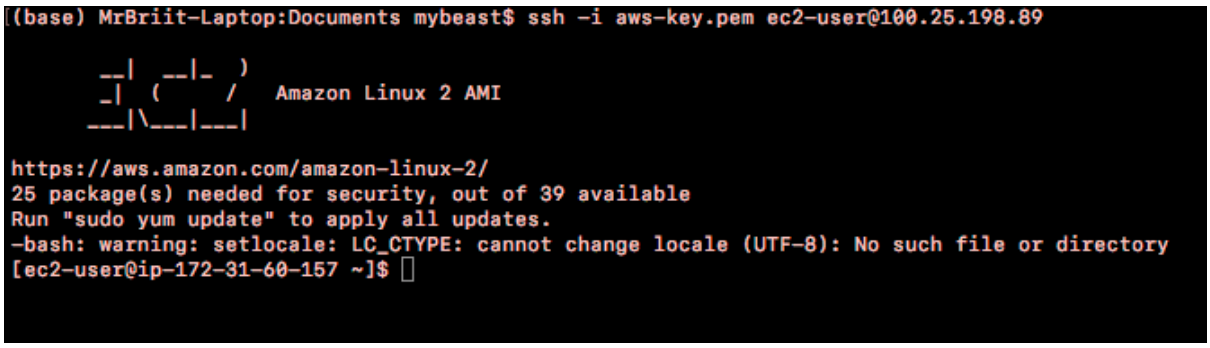


Start from a New terminal or Commandline.

STEPS.

1. Start ssh session: *ssh -i [directory where was is located]/aws-key.pem ec2-user@[add your public key]*

e.g. ssh -i Douments/aws-key.pem ec2-user@100.25.198.89



2. Update all existing packages: *sudo yum update*

NB: Press yes when prompted

3. Install apache: *sudo yum install httpd*

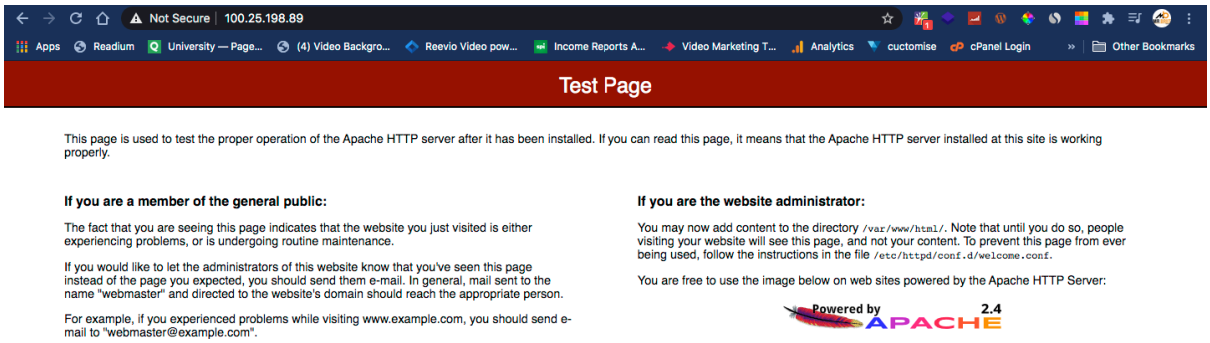
NB: Press yes when prompted

4. Start server: *sudo service httpd start*

```
[ec2-user@ip-172-31-60-157 ~]$ sudo service httpd start
Redirecting to /bin/systemctl start httpd.service
[ec2-user@ip-172-31-60-157 ~]$
```

5. Test you default page. *Copy and paste your public key in your browser*

You should see a default page like this



6. Navigate to the home folder: *cd /*

cd home

```
[ec2-user@ip-172-31-60-157 ~]$ cd /
[ec2-user@ip-172-31-60-157 /]$ cd home
[ec2-user@ip-172-31-60-157 home]$ ls
ec2-user
```

7. Copy your index.html file to your apache server.

Refer to **7.1. of PART 1** to do that

e.g. *scp -i /Users/mybeast/Documents/aws-key.pem -r* ***index.html*** *ec2-user@100.25.198.89:/home/ec2-user*

```
(base) MrBriit-Laptop:flight_deployment mybeast$ scp -i /Users/mybeast/Documents/aws-key.pem -r index.
html ec2-user@100.25.198.89:/home/ec2-user
/etc/profile.d/lang.sh: line 19: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such f
ile or directory
index.html                                                    100%   12KB  12.9KB/s   00:00
(base) MrBriit-Laptop:flight_deployment mybeast$
```

```
[ec2-user@ip-172-31-60-157 home]$ ls
ec2-user
[ec2-user@ip-172-31-60-157 home]$ cd ec2-user
[ec2-user@ip-172-31-60-157 ~]$ ls
index.html
[ec2-user@ip-172-31-60-157 ~]$
```

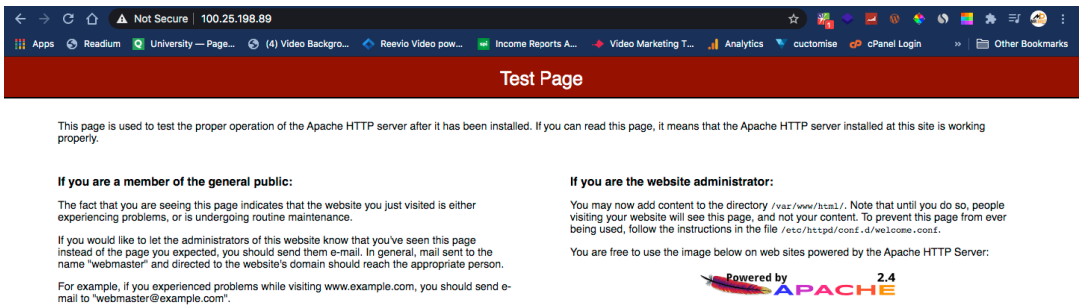
8. Copy the *index.html* file to *var* directory

sudo cp index.html /var/www/html

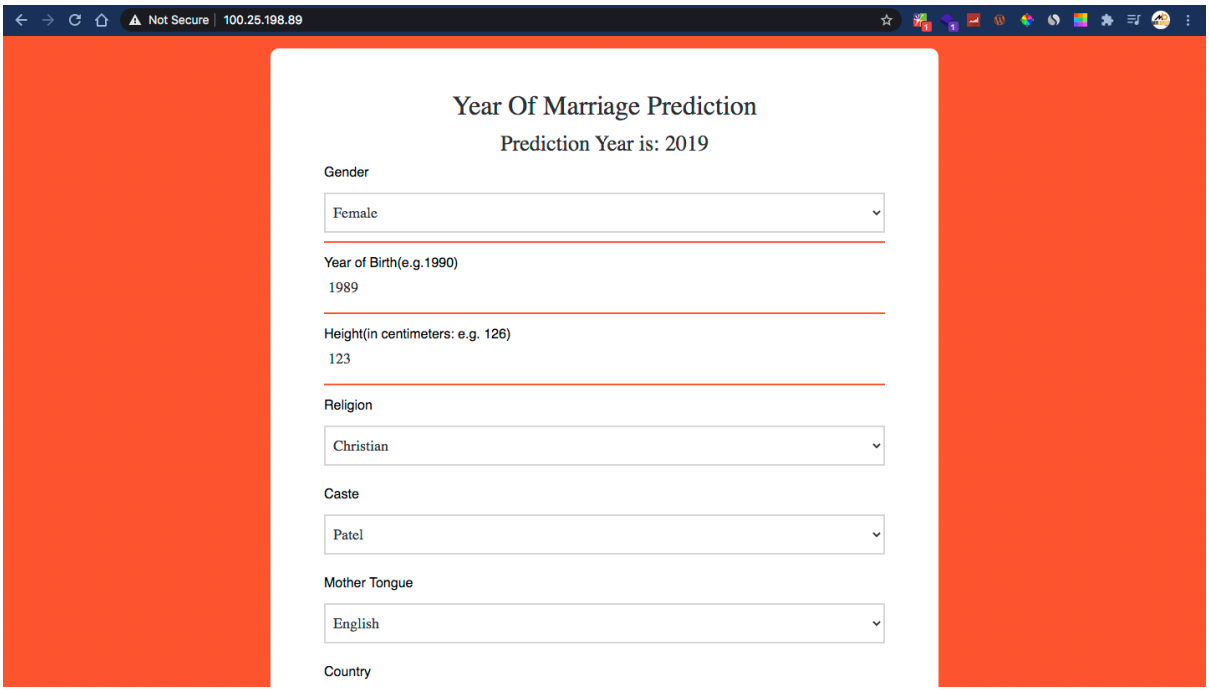
```
[ec2-user@ip-172-31-60-157 ~]$ sudo cp index.html /var/www/html
[ec2-user@ip-172-31-60-157 ~]$
```

9. Refresh your default page, or copy and paste your public key into your browser.

Refresh the below page



You should be seeing your default page now



You can get a customised URL from such places as bluest,goddady, etc.