

Build and Deploy an Sentiment Analysis of Reviews [AWS and Python]

Steps:

1. Build a model on your local box (Amazon Fine Food reviews) and store the model and other key model related variables in .pkl files
2. Launch a micro instance on AWS.
3. Connect to the AWS box [ssh]
4. Move the files to an AWS EC2 instance/box [scp]
5. Install all packages needed on the AWS box.
6. Run app.py on the AWS box.
7. Check the output in the browser.

Software needed:

1. Anaconda:
 - a. **Windows 64 bit:** https://repo.continuum.io/archive/Anaconda3-5.2.0-Windows-x86_64.exe
 - b. **Windows 32 bit:** <https://repo.continuum.io/archive/Anaconda3-5.2.0-Windows-x86.exe>
 - c. **Mac :** https://repo.continuum.io/archive/Anaconda3-5.2.0-MacOSX-x86_64.sh
 - d. **Linux 64 bit:** https://repo.continuum.io/archive/Anaconda3-5.2.0-Linux-x86_64.sh
 - e. **Linux 32 bit:** <https://repo.continuum.io/archive/Anaconda3-5.2.0-Linux-x86.sh>
 - f. **Check the previous Archives of Anaconda:** <https://repo.continuum.io/archive/>

2. Packages needed:

1. pip3
2. pandas
3. numpy
4. sklearn
5. beautifulsoup4
6. lxml
7. flask
8. regex

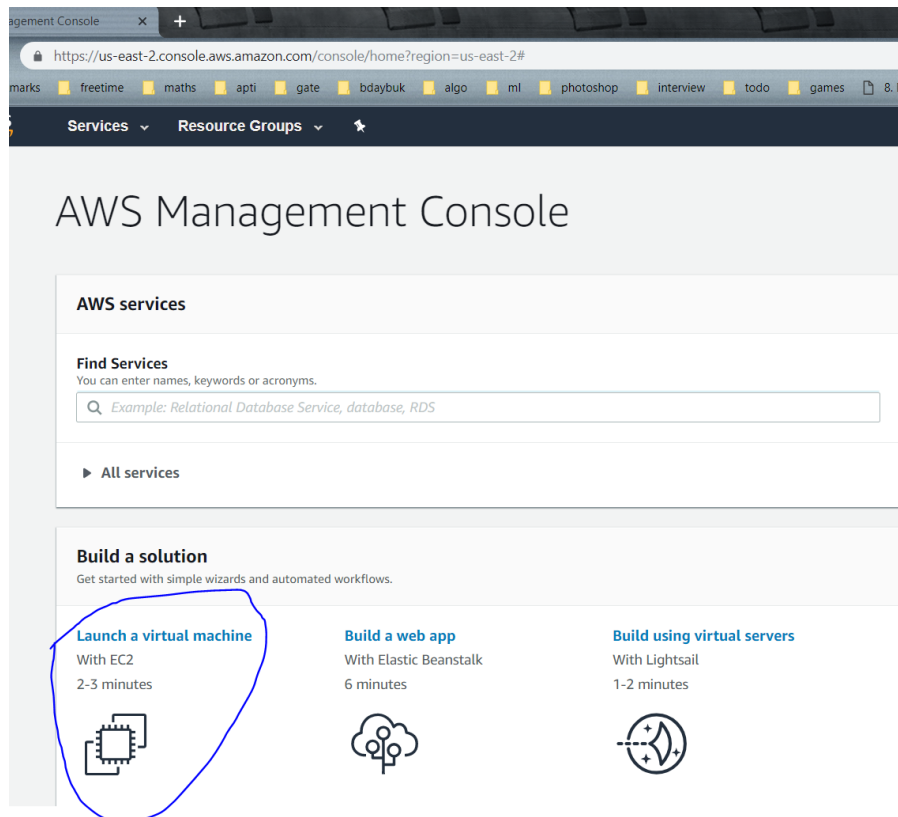
you can copy all these packages and try like this:

<https://stackoverflow.com/a/15593865/4084039>

[1] Launch a micro instance on AWS.

Creating an instance:

1. Create an AWS account <https://aws.amazon.com>,
<https://portal.aws.amazon.com/billing/signup#/start>
2. Login: <https://console.aws.amazon.com>
After login:




Launch the EC2 instance

3. Choose the ubuntu free tier


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 E

**Red Hat**
Free tier eligible

Red Hat Enterprise Linux version 7.6 (HVM), EBS General Purpose (SSD) Volume Type
Root device type: ebs Virtualization type: hvm

**SUSE Linux**
Free tier eligible

SUSE Linux Enterprise Server 15 (HVM), SSD Volume Type - ami-0eb9f58db22854f8f
Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.
Root device type: ebs Virtualization type: hvm

**Ubuntu Server 18.04 LTS (HVM), SSD Volume Type**
Free tier eligible

ami-0c55b159cbf1f0 (64-bit x86) / ami-0f2057f28f0a44d06 (64-bit Arm)
Ubuntu Server 18.04 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

64-bit (x86)
64-bit (Arm)

Select

64-bit (x86)
64-bit (Arm)

Select

Click on select

4. Choose t2.micro free tier eligible

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes

Cancel Previous **Review and Launch** Next: Configure Instance Details


Click on review and launch

5.

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 [Edit AMI](#)

**Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0c55b159cbf1f0**
Free tier eligible
Ubuntu Server 18.04 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	Variable	1	1	EBS only	-	Low to Moderate

Security Groups [Edit security groups](#)

Security group name: launch-wizard-2
Description: launch-wizard-2 created 2019-03-23T17:21:15.794+05:30

Type	Protocol	Port Range	Source	Description
This security group has no rules				

Cancel Previous **Launch**

Click on launch

6.

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

my_key_pair

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

Click on “Download Key Pair” and save the .pem file then click on “Launch Instance”

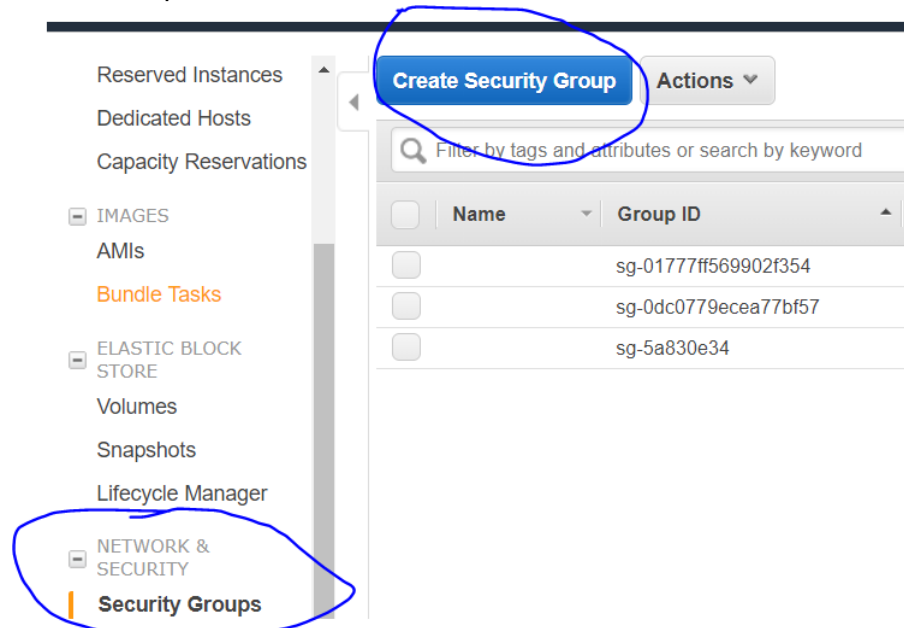
7.

The screenshot shows the AWS Management Console with the EC2 Dashboard selected. The 'Instances' tab is active, displaying a table with one instance. The instance is named 'i-06fe95c371927683f', has an 'Instance ID' of 'i-06fe95c371927683f', is of type 't2.micro', in the 'us-east-2b' Availability Zone, and is in the 'running' state. The 'Status Checks' column shows '2/2 checks ...', and the 'Alarm Status' is 'None'. The 'Public DNS (IPv4)' is 'i-06fe95c371927683f.us-east-2.compute.amazonaws.com' and the 'IPv4 Public IP' is '13.59.191.237'.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 Public IP
	i-06fe95c371927683f	t2.micro	us-east-2b	running	2/2 checks ...	None	i-06fe95c371927683f.us-east-2.compute.amazonaws.com	13.59.191.237	

You will see this screen, you have successfully launched the an EC2 instance, now we need to launch an flask api in it

8. Final step:



Select the "Network & security" -> Security groups and then click "Create Security Group"

The 'Create Security Group' dialog box is shown. It contains the following fields and options:

- Security group name:** anywhere
- Description:** anywhere
- VPC:** vpc-26713942 (default)
- Security group rules:**
 - Inbound:** Selected tab.
 - Type:** All traffic
 - Protocol:** All
 - Port Range:** 0 - 65535
 - Source:** Anywhere (highlighted with a red box), 0.0.0.0/0, ::/0
- Add Rule:** Button to add more rules.
- Buttons:** Cancel and Create at the bottom right.

Then add the specific security group to **network interface**

Services

Resource Groups

INSTANCES

Instances

Spot Requests

Reserved Instances

Dedicated Hosts

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

NETWORK & SECURITY

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

Create Network Interface

Attach

Detach

Delete

Actions

Filter by tags and attributes or search by keyword

	Name	Network interf	Subnet ID	VPC ID	Zon
<input type="checkbox"/>		eni-3013077d		942	ap-

Attach

Detach

Delete

Manage IP Addresses

Associate Address

Disassociate Address

Change Termination Behavior

Change Security Groups

Change Source/Dest. Check

Add/Edit Tags

Change Description

Create Flow Log

Network Interface: eni-3013077d

Details

Flow Logs

Tags

Network Interfaces > Change Security Groups

Change Security Groups

Network Interface eni-06d51045198e42093

Security groups*

Filter by attributes or search by keyword

<<

<

1 to 3 of 3

>

>>

	Group ID	Group name	Description
<input checked="" type="checkbox"/>	sg-01777ff569...	anywhere	anywhere
<input checked="" type="checkbox"/>	sg-0dc0779ecea...	launch-wizard-1	launch-wizard-1 created 2019-03-23T12:36:12.510+05:30
<input type="checkbox"/>	sg-5a830e34	default	default VPC security group

* Required

Cancel

Save

[2] Connect to the AWS box

Connect To Your Instance

I would like to connect with

☒ A standalone SSH client

☐ A Java SSH Client directly from my browser (Java required)

To access your instance:

1. Open an SSH client. (find out how to [connect using PuTTY](#))

2. Locate your private key file (for_live.pem). The wizard automatically detects the key you used to launch the instance.

3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

```
chmod 400 for_live.pem
```

4. Connect to your instance using its Public DNS:

```
ec2-13-59-191-237.us-east-2.compute.amazonaws.com
```

Example:

```
ssh -i "for_live.pem" ubuntu@ec2-13-59-191-237.us-east-2.compute.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our [connection documentation](#).

Close

[3] Move the files to an AWS EC2 instance/box []

Command line to copy files

C:\Users\Asus\OneDrive\Desktop>scp -i "for_live.pem" -r AFR <http://ec2-3-14-72-94.us-east-2.compute.amazonaws.com:~/>

app.py	100%	3888	16.9KB/s	00:00
count_vect.pkl	100%	31MB	2.6MB/s	00:11
model.pkl	100%	79KB	312.4KB/s	00:00
model.py	100%	4864	19.7KB/s	00:00
index.html	100%	332	1.4KB/s	00:00

```

C:\Users\Manasa>scp -i my_key.pem -r AFR ubuntu@ec2-3-14-72-94.us-east-2.compute.amazonaws.com:~/
app.py                                     100% 3888   16.9KB/s   00:00
count_vect.pkl                           100% 31MB   2.6MB/s   00:11
model.pkl                                 100% 79KB   312.4KB/s  00:00
model.py                                  100% 4864   19.7KB/s   00:00
index.html                                100% 332    1.4KB/s   00:00

C:\Users\Manasa>

```

[4] Install all packages needed on the AWS box.

```

sudo apt-get install python3-pip
pip3 install <each of the following packages>
Packages needed:
    pip3
    pandas
    numpy
    sklearn
    beautifulsoup4
    lxml
    flask
    regex

```

[5] Run app.py on the AWS box.

```

ubuntu@ip-172-31-27-97: ~/AFR
ubuntu@ip-172-31-27-97:~/AFR$ python3 app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
183.83.170.52 - - [24/Mar/2019 04:00:38] "GET /index HTTP/1.1" 200 -

```


[6] Check the output in the browser.

← → ↻ ⓘ Not secure | ec2-13-59-191-237.us-east-2.compute.amazonaws.com:8080/index

Amazon Fine Food Reviews: Sentiment Analysis

Review Text

Have been having this since years. Much better option than Bru.Nescafe still managing to do well in market with all the competitors breathing down it\'s neck. Good one!

Submit

[7] Check the result in the browser.

← → ↻ ⬆ ⓘ Not secure | ec2-3-14-72-94.us-east-2.compute.amazonaws.com:8080/predict

```
{"prediction": "Positive"}
```