Configuration Manual

Application of short text topic modelling techniques to Greta Thunberg discussion on Twitter

MSc Research Project
Masters in Data Analytics

Sean Dingemans
Student ID: x18199089

School of Computing
National College of Ireland

Supervisor: Dr Catherine Mulwa

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Sean Dingemans…….………………………………………………………………………

Student ID: x18199089……………………………………………………………………………

Programme: Masters in Data Analytics ……………………… **Year:** 2020……………………..

Module: MSc Research Project……………………………………………………………….………

Lecturer: Dr Catherine Mulwa…………………………………………………………………….………

Submission Due Date: 17 August 2020……………………………………………………………………….………

Project Title: Application of short text topic modelling techniques to Greta Thunberg discussion on Twitter……….…………………………….………

Word Count: …2297…………………… **Page Count:** ……….…57…

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……………………

Date: 17 August 2020………………………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

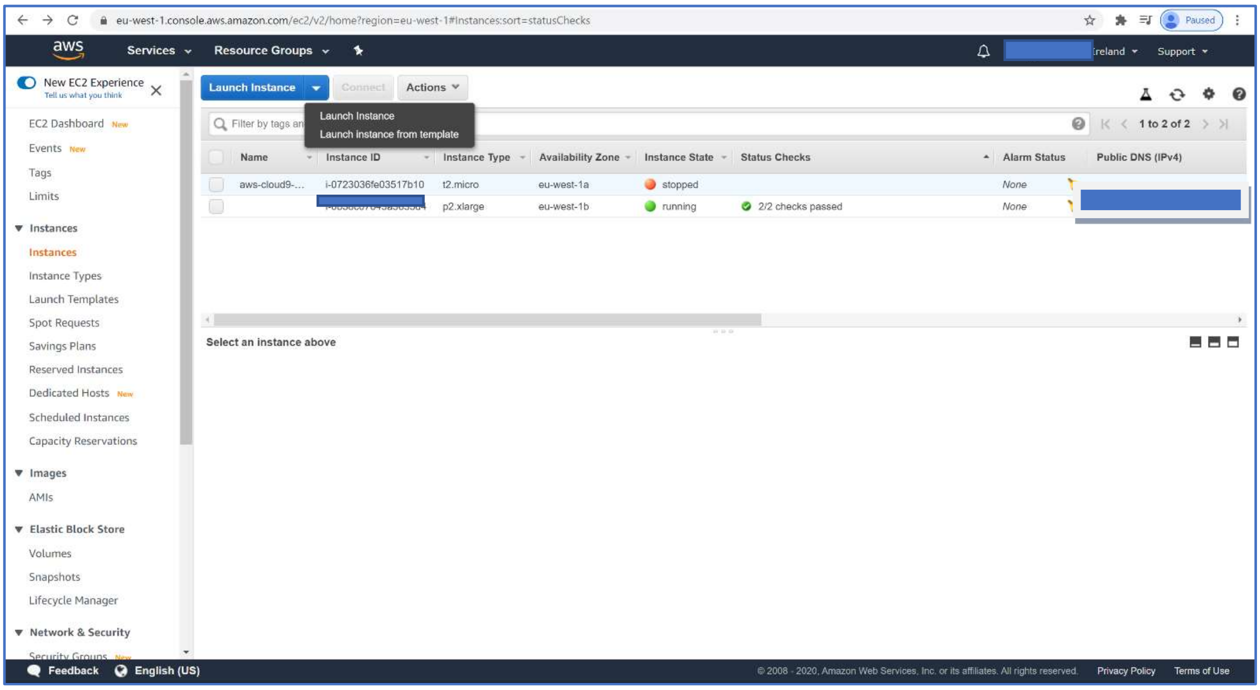
Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Sean Dingemans
Student ID: x18199089

1 Environment Setup

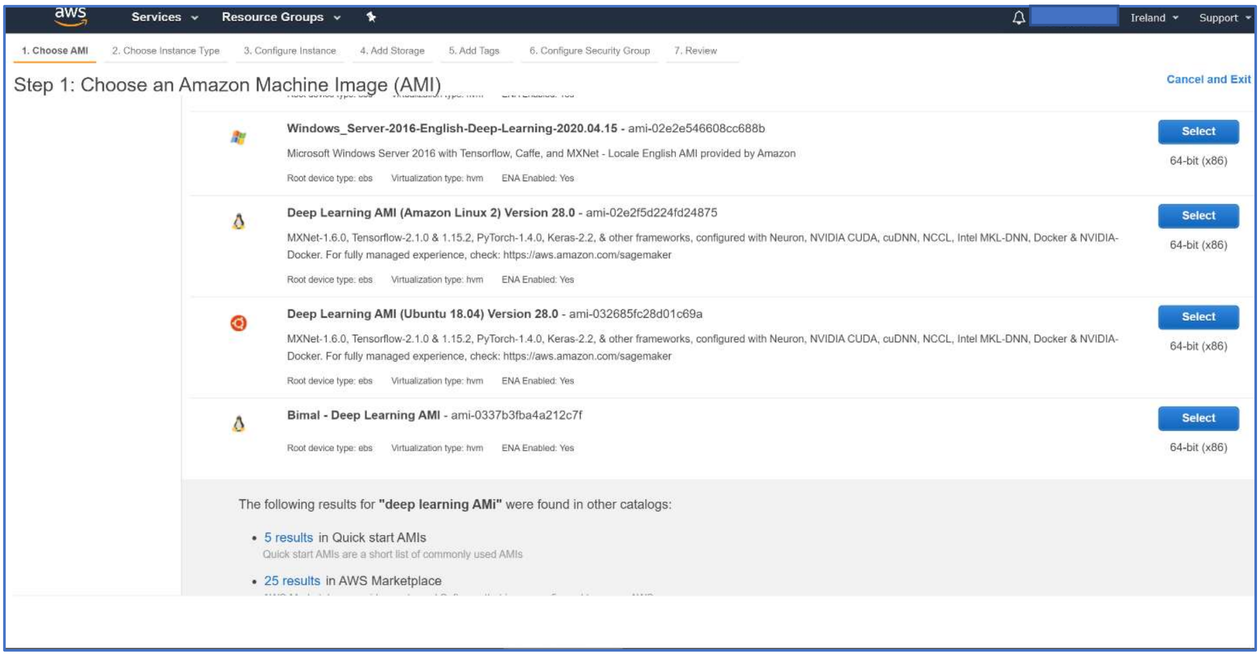
An AWS account was setup with billing information then filled in. Afterwards a virtual machine instance was setup through Amazon's Elastic Cloud Compute (EC2) device. The instance creation process begins with the selection of 'Launch Instance' as shown in Figure 1.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
aws-cloud9-...	i-0723036fe03517b10	t2.micro	eu-west-1a	● stopped		None	
	[redacted]	p2.xlarge	eu-west-1b	● running	✓ 2/2 checks passed	None	[redacted]

Figure 1: AWS creation of instance by launce instance

The first step of instance creation was to select an Amazon Machine Image (AMI). An AMI was chosen for the ec2 instance that had much of the required preinstalled software, the Deep-Learning Ubuntu AMI: pandas, numpy, python, jupyter notebook, java 1.8, Anaconda



Step 1: Choose an Amazon Machine Image (AMI)

Windows_Server-2016-English-Deep-Learning-2020.04.15 - ami-02e2e546608cc688b
Microsoft Windows Server 2016 with Tensorflow, Caffe, and MXNet - Locale English AMI provided by Amazon
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit (x86)

Deep Learning AMI (Amazon Linux 2) Version 28.0 - ami-02e2f5d224fd24875
MXNet-1.6.0, Tensorflow-2.1.0 & 1.15.2, PyTorch-1.4.0, Keras-2.2, & other frameworks, configured with Neuron, NVIDIA CUDA, cuDNN, NCCL, Intel MKL-DNN, Docker & NVIDIA-Docker. For fully managed experience, check: https://aws.amazon.com/sagemaker
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit (x86)

Deep Learning AMI (Ubuntu 18.04) Version 28.0 - ami-032685fc28d01c69a
MXNet-1.6.0, Tensorflow-2.1.0 & 1.15.2, PyTorch-1.4.0, Keras-2.2, & other frameworks, configured with Neuron, NVIDIA CUDA, cuDNN, NCCL, Intel MKL-DNN, Docker & NVIDIA-Docker. For fully managed experience, check: https://aws.amazon.com/sagemaker
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit (x86)

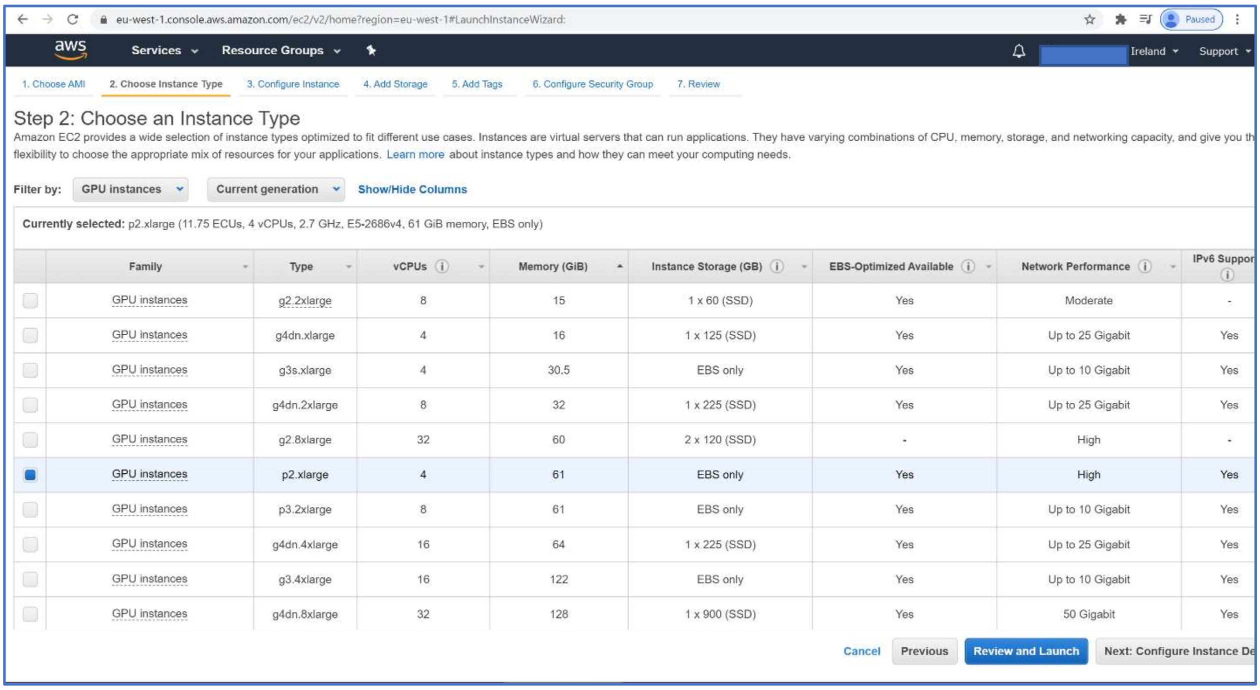
Bimal - Deep Learning AMI - ami-0337b3fba4a212c7f
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit (x86)

The following results for **"deep learning AMi"** were found in other catalogs:

- 5 results in Quick start AMIs
  Quick start AMIs are a short list of commonly used AMIs
- 25 results in AWS Marketplace

Figure 2: AWS section of AWS AMI

An EC2 instance of the gpu family was chosen as these instances are optimised for computationally heavy tasks in addition to their hardware specs. The p2x instance was selected due to being affordable relative to other instances in the GPU family. Specs for the instance can be seen in Figure 3.



Step 2: Choose an Instance Type

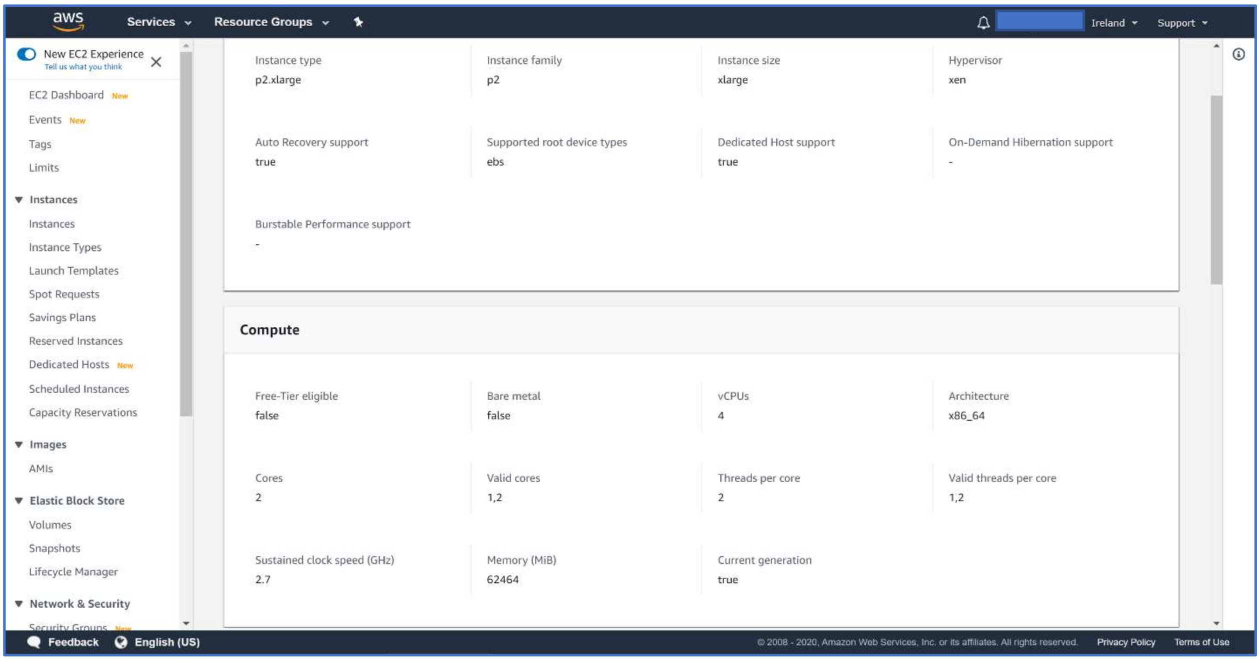
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you th flexibility to choose the appropriate mix of resources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

Filter by: GPU instances | Current generation | Show/Hide Columns

Currently selected: p2.xlarge (11.75 ECUs, 4 vCPUs, 2.7 GHz, E5-2686v4, 61 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Suppor
□	GPU instances	g2.2xlarge	8	15	1 x 60 (SSD)	Yes	Moderate	-
□	GPU instances	g4dn.xlarge	4	16	1 x 125 (SSD)	Yes	Up to 25 Gigabit	Yes
□	GPU instances	g3s.xlarge	4	30.5	EBS only	Yes	Up to 10 Gigabit	Yes
□	GPU instances	g4dn.2xlarge	8	32	1 x 225 (SSD)	Yes	Up to 25 Gigabit	Yes
□	GPU instances	g2.8xlarge	32	60	2 x 120 (SSD)	-	High	-
■	GPU instances	p2.xlarge	4	61	EBS only	Yes	High	Yes
□	GPU instances	p3.2xlarge	8	61	EBS only	Yes	Up to 10 Gigabit	Yes
□	GPU instances	g4dn.4xlarge	16	64	1 x 225 (SSD)	Yes	Up to 25 Gigabit	Yes
□	GPU instances	g3.4xlarge	16	122	EBS only	Yes	Up to 10 Gigabit	Yes
□	GPU instances	g4dn.8xlarge	32	128	1 x 900 (SSD)	Yes	50 Gigabit	Yes

Figure 3: AWS GPU Instance Hardware Type selection



[redacted]

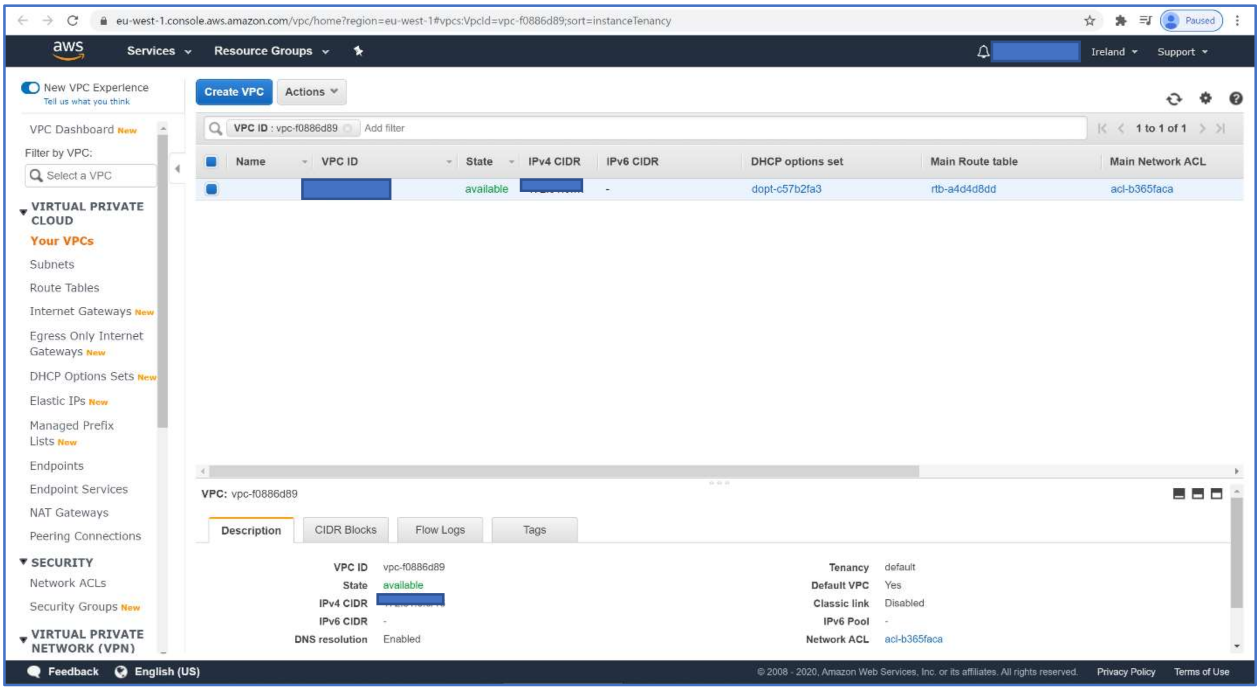
Instance type p2.xlarge	Instance family p2	Instance size xlarge	Hypervisor xen
Auto Recovery support true	Supported root device types ebs	Dedicated Host support true	On-Demand Hibernation support -
Burstable Performance support -			

Compute

Free-Tier eligible false	Bare metal false	vCPUs 4	Architecture x86_64
Cores 2	Valid cores 1,2	Threads per core 2	Valid threads per core 1,2
Sustained clock speed (GHz) 2.7	Memory (MiB) 62464	Current generation true	

Figure 4: AWS Brief overview of some of the instance hardware specs.

A standard VPC was created. This was this first step in allowing the instance to have an IP address so that it could be accessed. The IPV4 CIDR block size provides 64000 different ip addresses. (see figure 4).



eu-west-1.console.aws.amazon.com/vpc/home?region=eu-west-1#vpcs:VpcId=vpc-f0886d89;sort=instanceTenancy

[redacted]

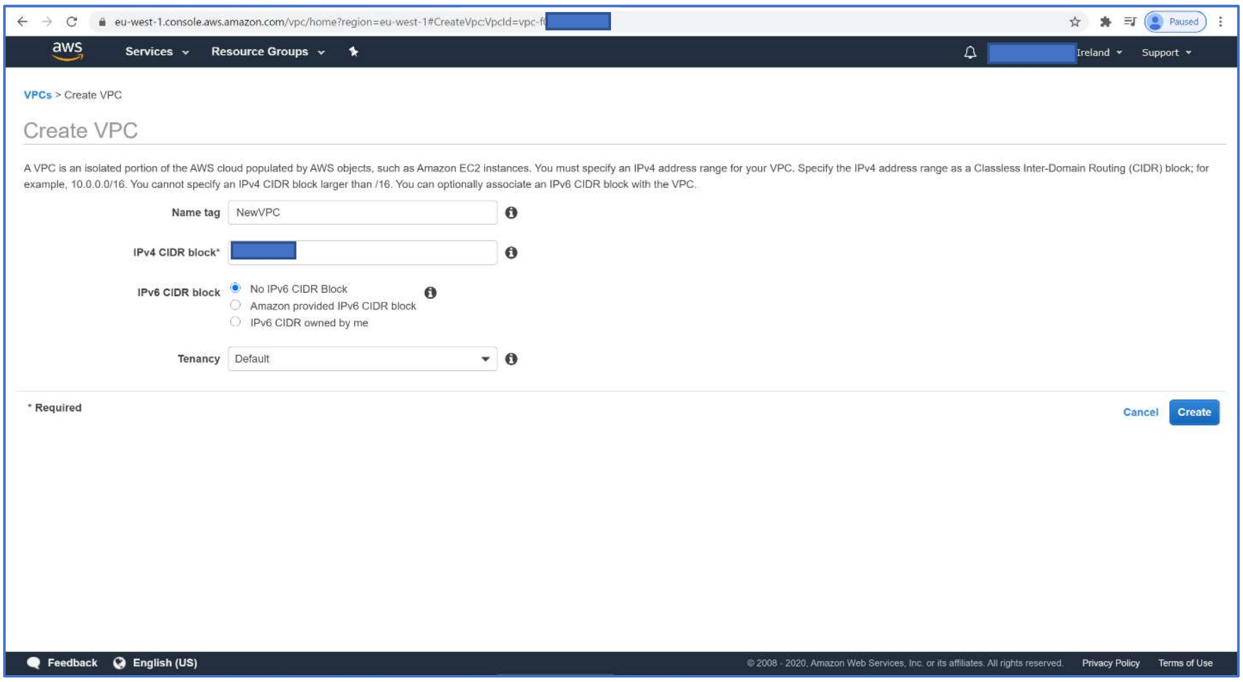
VPC ID : vpc-f0886d89

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Main Route table	Main Network ACL
[redacted]		available	[redacted]	-	dopt-c57b2fa3	rtb-a4d4d8dd	acl-b365faca

VPC: vpc-f0886d89

VPC ID	vpc-f0886d89	Tenancy	default
State	available	Default VPC	Yes
IPv4 CIDR	[redacted]	Classic link	Disabled
IPv6 CIDR	-	IPv6 Pool	-
DNS resolution	Enabled	Network ACL	acl-b365faca

Figure 5: AWS VPC creation part 1


eu-west-1.console.aws.amazon.com/vpc/home?region=eu-west-1#CreateVpc:VpcId=vpc-f[redacted]
Paused
aws
Services
Resource Groups
[redacted]Ireland
Support

VPCs > Create VPC

Create VPC

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances. You must specify an IPv4 address range for your VPC. Specify the IPv4 address range as a Classless Inter-Domain Routing (CIDR) block; for example, 10.0.0.0/16. You cannot specify an IPv4 CIDR block larger than /16. You can optionally associate an IPv6 CIDR block with the VPC.

Name tag: NewVPC

IPv4 CIDR block*: [redacted]

IPv6 CIDR block: ● No IPv6 CIDR Block ○ Amazon provided IPv6 CIDR block ○ IPv6 CIDR owned by me

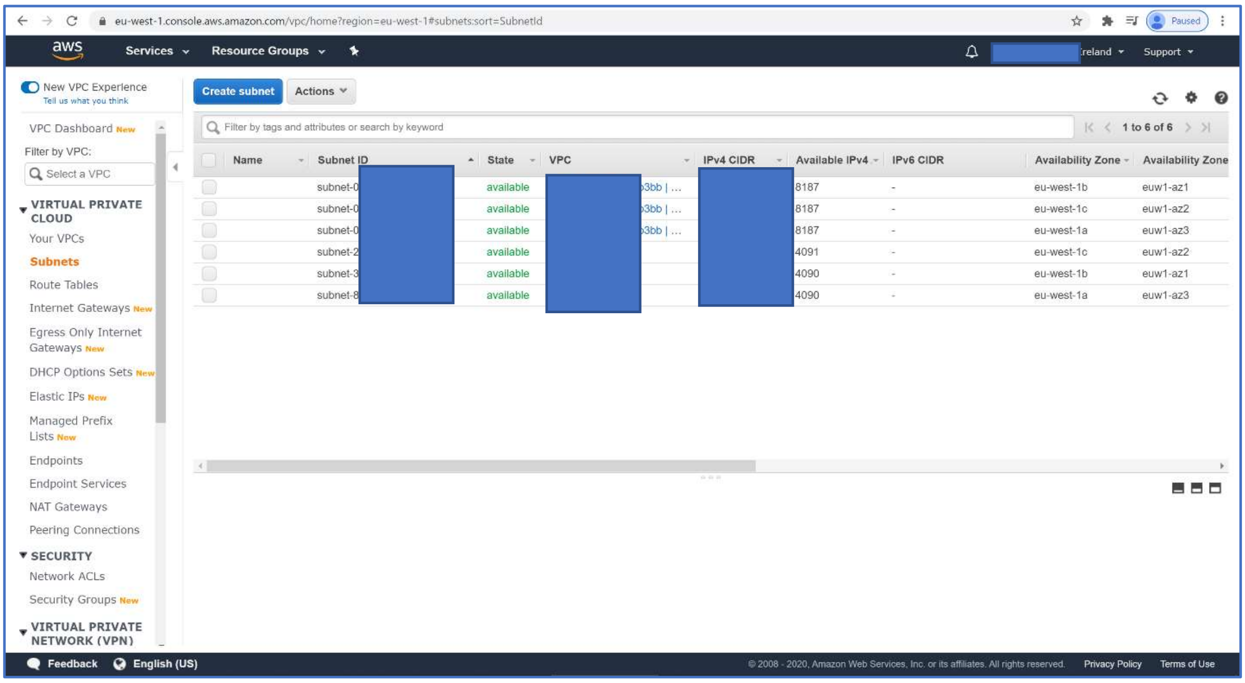
Tenancy: Default

* Required

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

Figure 6: AWS CIDR Block specification for VPC part 2

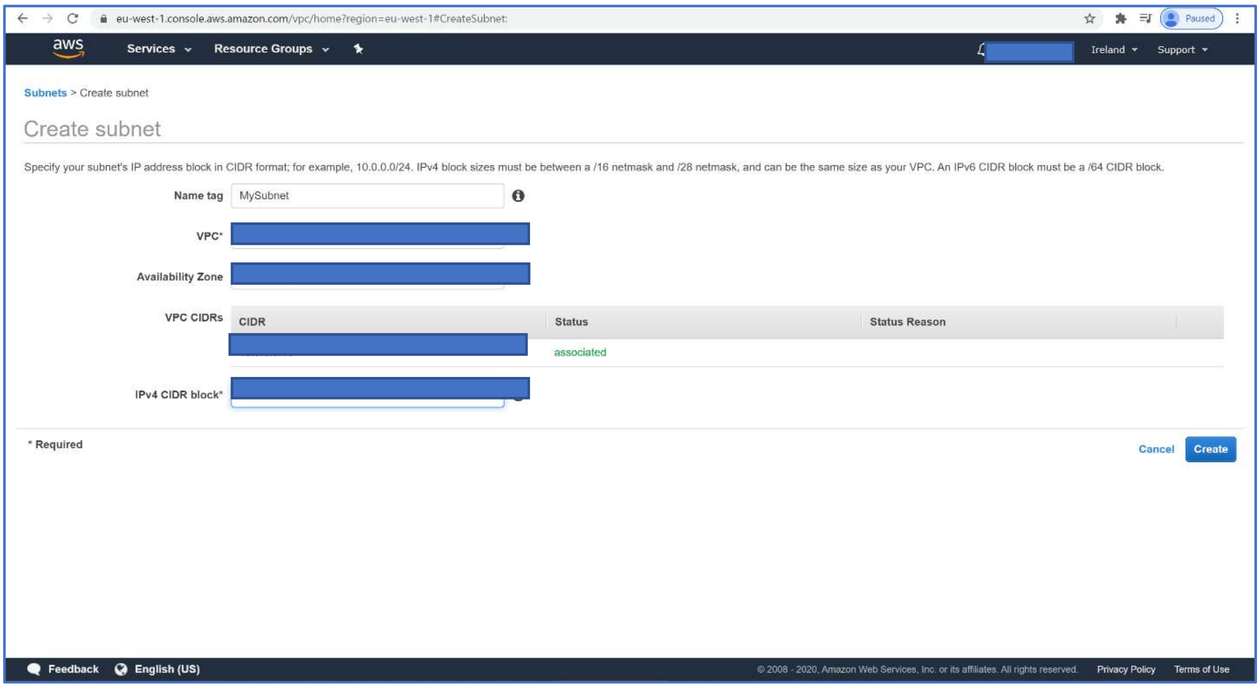
For the VPC to work, a subnet must be assigned. A subnet utilising 16 of the 64000 possible Ip addresses was created. Only one IP address will be needed, however no additional costs are accrued. The region of the instance is also set here. It is set to eu-west-1b. The region specified is not critical, as there are not thousands of clients accessing the instance, and thus any lag is negligible.



Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone	Availability Zone
	subnet-0[redacted]	available	[redacted]3bb | …	[redacted]	8187	-	eu-west-1b	euw1-az1
	subnet-0[redacted]	available	[redacted]3bb | …	[redacted]	8187	-	eu-west-1c	euw1-az2
	subnet-0[redacted]	available	[redacted]3bb | …	[redacted]	8187	-	eu-west-1a	euw1-az3
	subnet-2[redacted]	available	[redacted]	[redacted]	4091	-	eu-west-1c	euw1-az2
	subnet-3[redacted]	available	[redacted]	[redacted]	4090	-	eu-west-1b	euw1-az1
	subnet-8[redacted]	available	[redacted]	[redacted]	4090	-	eu-west-1a	euw1-az3

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

Figure 7: AWS Subnet creation for VPC



Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag: MySubnet

VPC*: [redacted]

Availability Zone: [redacted]

VPC CIDRs

CIDR	Status	Status Reason
[redacted]	associated	

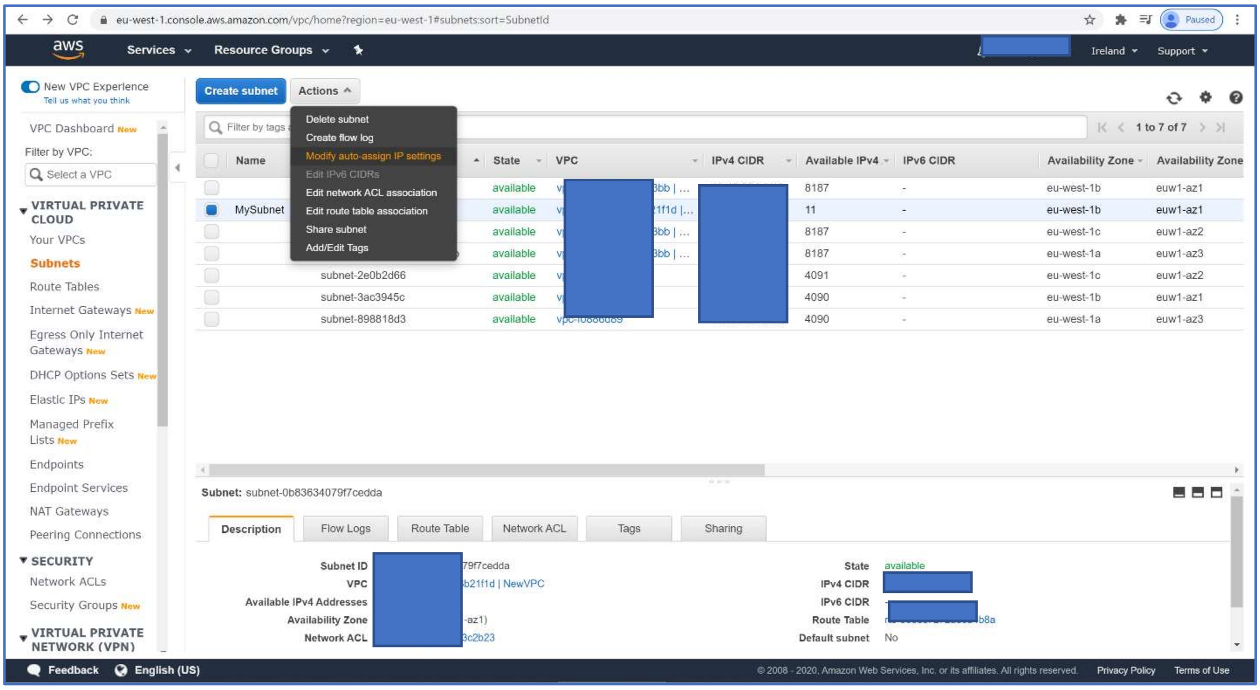
IPv4 CIDR block*: [redacted]

* Required

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

Figure 8: AWS Subnet Creation

This step was just done to initialise the instance with an ip address.



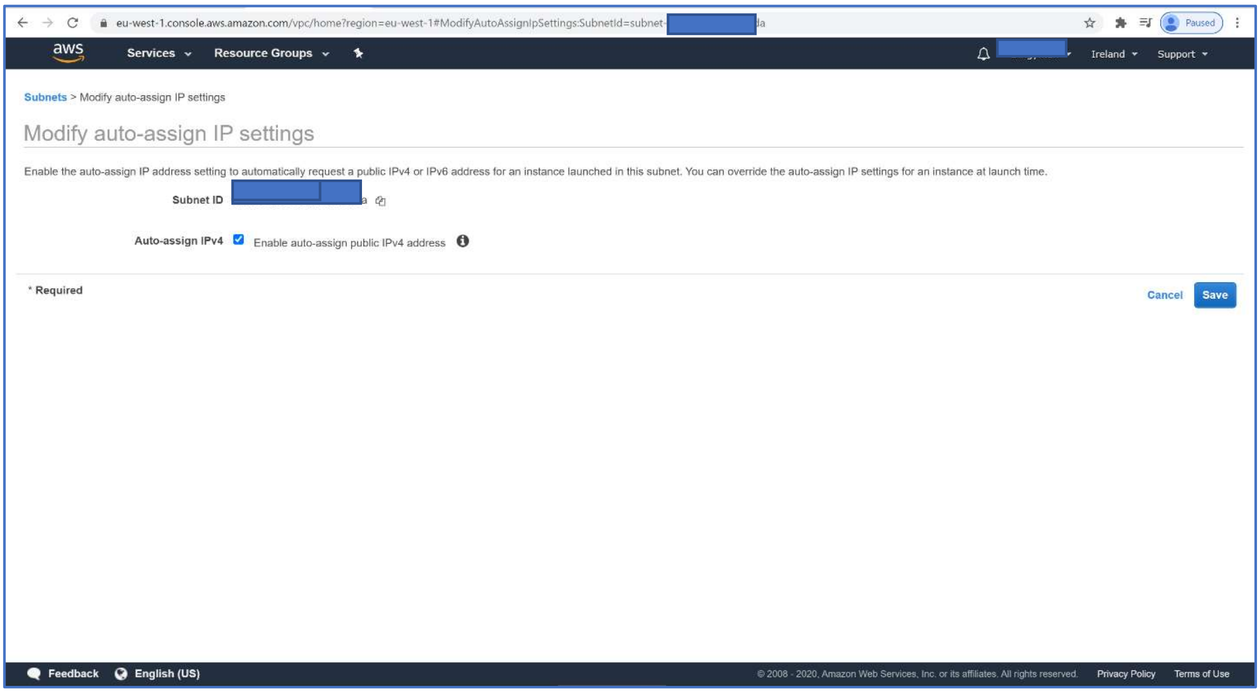
	Name		State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone	Availability Zone
□			available	v[redacted]3bb | …	[redacted]	8187	-	eu-west-1b	euw1-az1
■	MySubnet		available	v[redacted]1f1d | …	[redacted]	11	-	eu-west-1b	euw1-az1
□			available	v[redacted]3bb | …	[redacted]	8187	-	eu-west-1c	euw1-az2
□			available	v[redacted]3bb | …	[redacted]	8187	-	eu-west-1a	euw1-az3
□		subnet-2e0b2d66	available	v[redacted]	[redacted]	4091	-	eu-west-1c	euw1-az2
□		subnet-3ac3945c	available	v[redacted]	[redacted]	4090	-	eu-west-1b	euw1-az1
□		subnet-898818d3	available	[redacted]	[redacted]	4090	-	eu-west-1a	euw1-az3

Subnet: subnet-0b83634079f7cedda

Subnet ID	[redacted]79f7cedda	State	available
VPC	[redacted]b21f1d | NewVPC	IPv4 CIDR	[redacted]
Available IPv4 Addresses	[redacted]	IPv6 CIDR	[redacted]
Availability Zone	[redacted]-az1)	Route Table	[redacted]b8a
Network ACL	[redacted]3c2b23	Default subnet	No

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

Figure 9: AWS IP address allocation for instance creation



eu-west-1.console.aws.amazon.com/vpc/home?region=eu-west-1#ModifyAutoAssignIpSettings:SubnetId=subnet-[redacted]a
Paused
aws
Services
Resource Groups
[redacted]
Ireland
Support

Subnets > Modify auto-assign IP settings

Modify auto-assign IP settings

Enable the auto-assign IP address setting to automatically request a public IPv4 or IPv6 address for an instance launched in this subnet. You can override the auto-assign IP settings for an instance at launch time.

Subnet ID [redacted]a

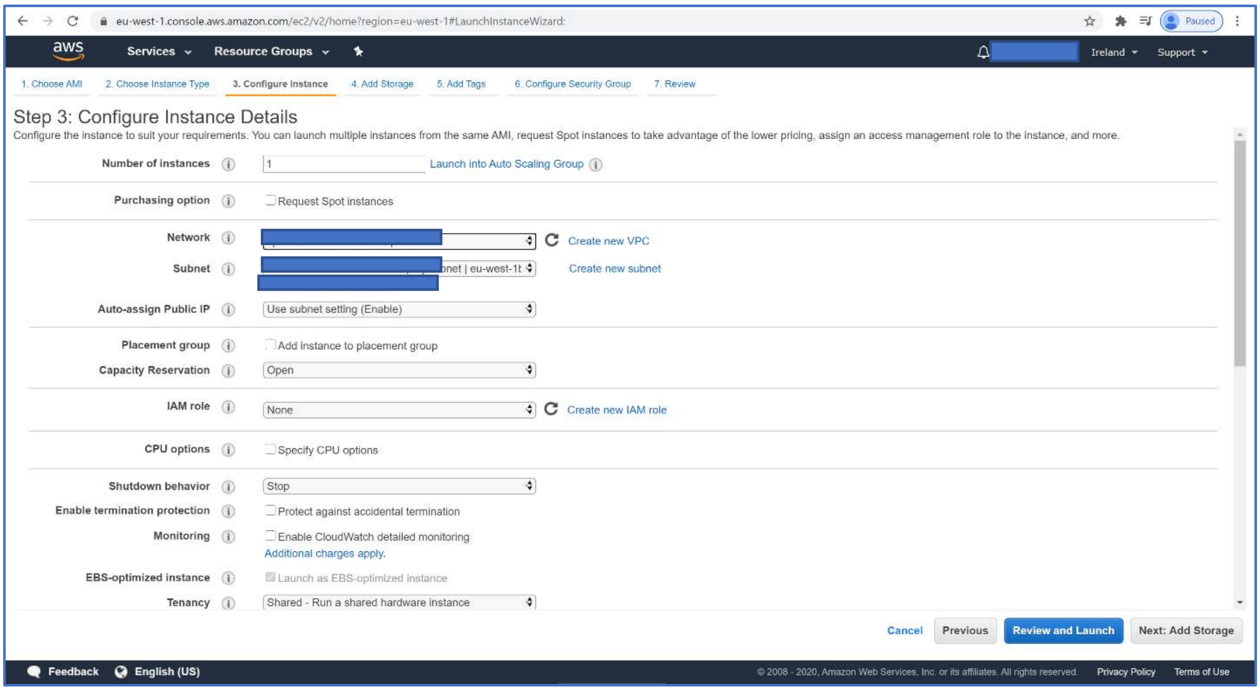
Auto-assign IPv4 ☑ Enable auto-assign public IPv4 address

* Required

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

Figure 10: AWS Auto-assign IP address

The Created VPC along with its subnet could now be allocated to the instance to be created.



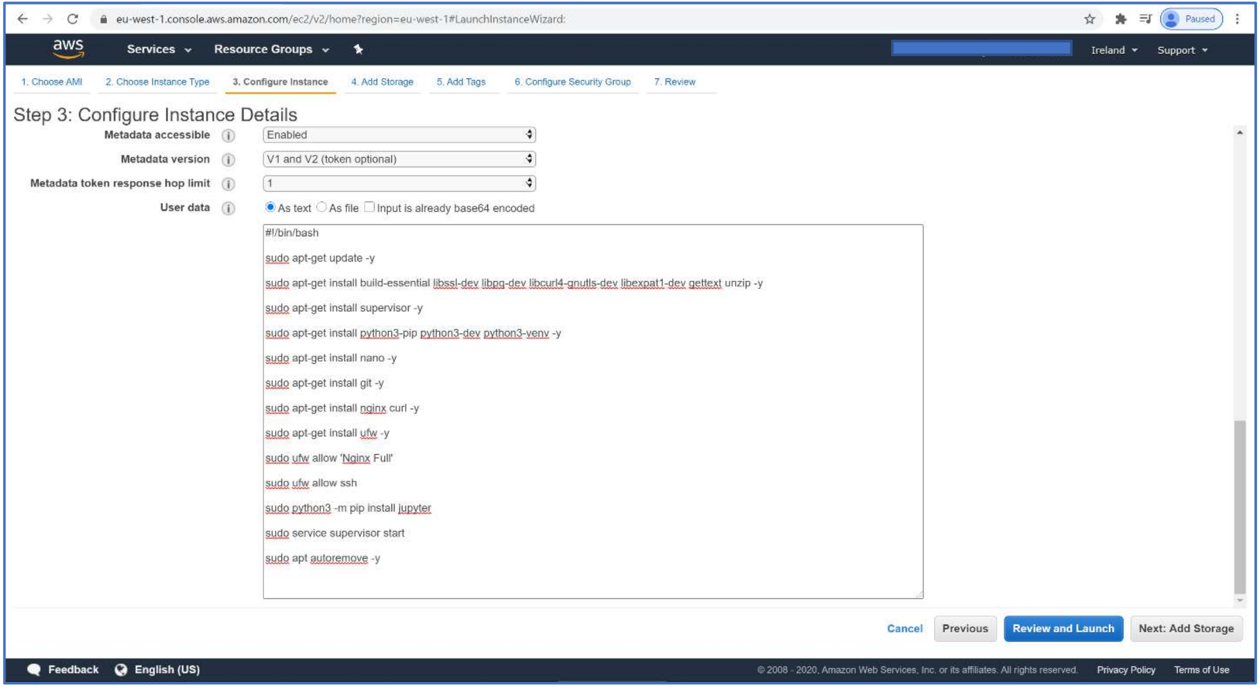
Step 3: Configure Instance Details

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	[redacted] Create new VPC
Subnet	[redacted]bnet | eu-west-1b [redacted] Create new subnet
Auto-assign Public IP	Use subnet setting (Enable)
Placement group	□ Add instance to placement group
Capacity Reservation	Open
IAM role	None Create new IAM role
CPU options	□ Specify CPU options
Shutdown behavior	Stop
Enable termination protection	□ Protect against accidental termination
Monitoring	□ Enable CloudWatch detailed monitoring Additional charges apply.
EBS-optimized instance	☑ Launch as EBS-optimized instance
Tenancy	Shared - Run a shared hardware instance

Figure 11: AWS Allocation of sub-net and vpc to instance

In step 3 of launching the instance, a list of installation commands were added for the environment. Installed software included nginx for the ability to run programs in the browser through the instance's associated Ip address, once the instance was running supervisor for starting up nginx automatically when starting the instance after it is stopped naano text editor.



Step 3: Configure Instance Details

Metadata accessible: Enabled

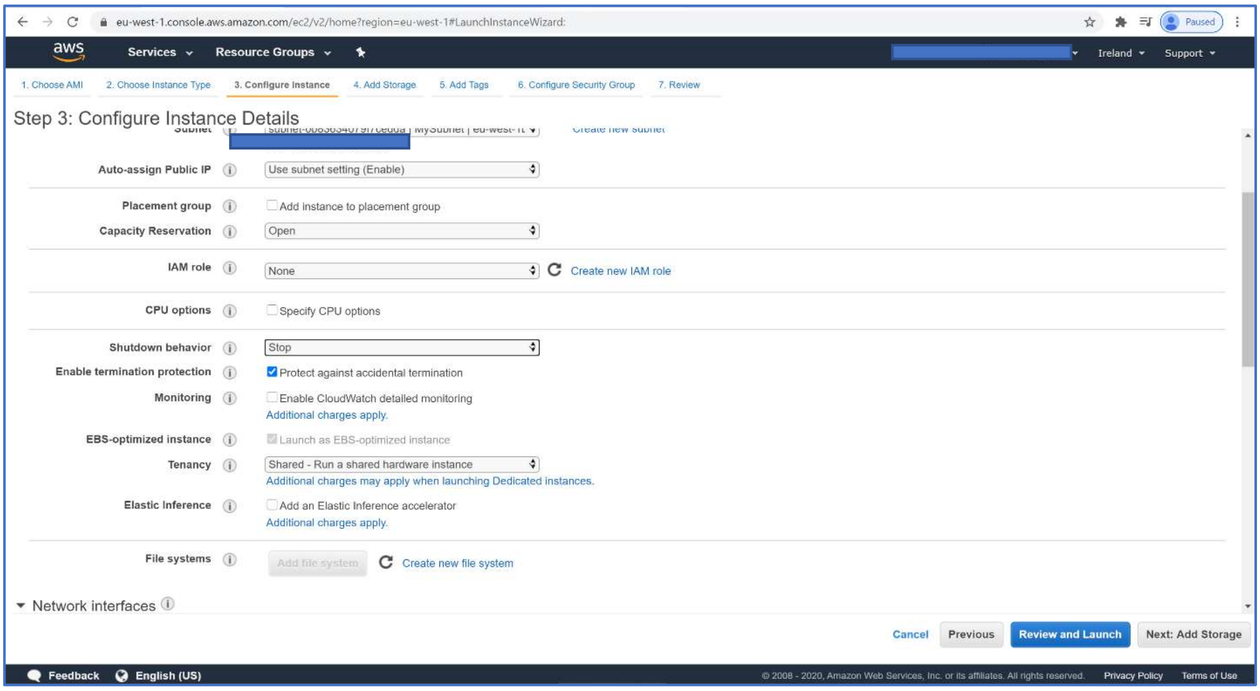
Metadata version: V1 and V2 (token optional)

Metadata token response hop limit: 1

User data: ● As text ○ As file □ Input is already base64 encoded

```
#!/bin/bash
sudo apt-get update -y
sudo apt-get install build-essential libssl-dev libpq-dev libcurl4-gnutls-dev libexpat1-dev gettext unzip -y
sudo apt-get install supervisor -y
sudo apt-get install python3-pip python3-dev python3-venv -y
sudo apt-get install nano -y
sudo apt-get install git -y
sudo apt-get install nginx curl -y
sudo apt-get install ufw -y
sudo ufw allow 'Nginx Full'
sudo ufw allow ssh
sudo python3 -m pip install jupyter
sudo service supervisor start
sudo apt autoremove -y
```

Figure 12: AWS Configuration Details



Step 3: Configure Instance Details

Subnet: [redacted]

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

Placement group: □ Add instance to placement group

Capacity Reservation: Open

IAM role: None

CPU options: □ Specify CPU options

Shutdown behavior: Stop

Enable termination protection: ☑ Protect against accidental termination

Monitoring: □ Enable CloudWatch detailed monitoring
Additional charges apply.

EBS-optimized instance: ☑ Launch as EBS-optimized instance

Tenancy: Shared - Run a shared hardware instance
Additional charges may apply when launching Dedicated instances.

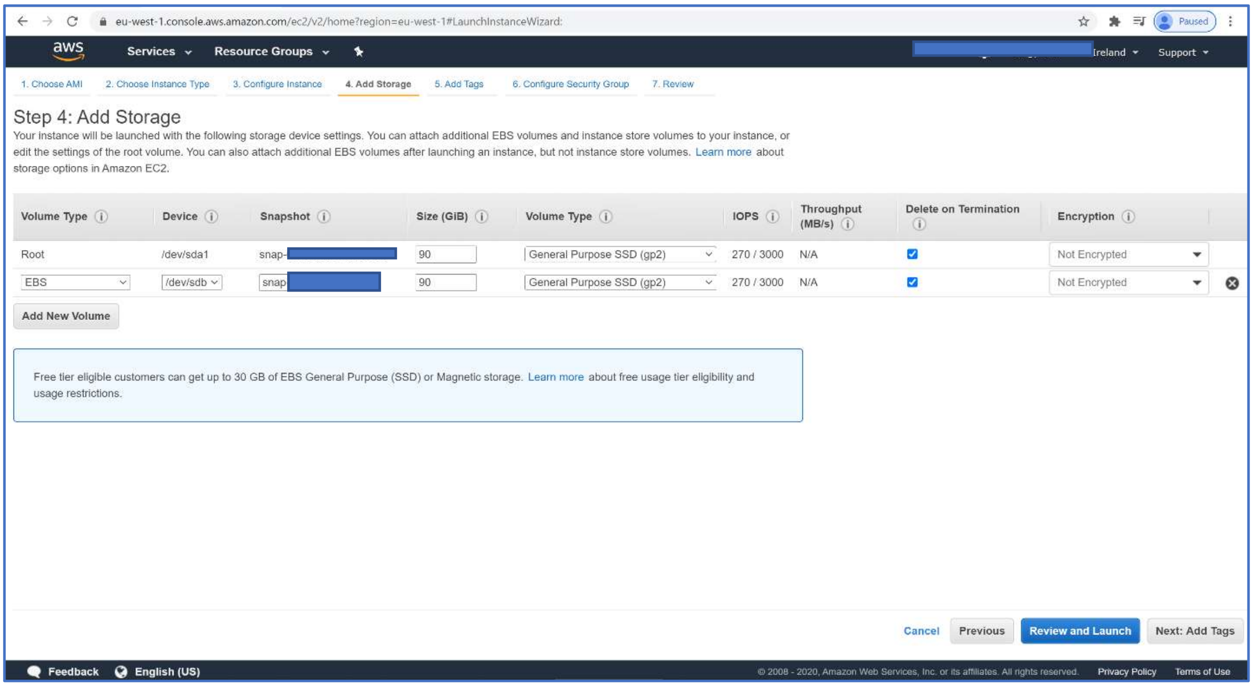
Elastic Inference: □ Add an Elastic Inference accelerator
Additional charges apply.

File systems

Network interfaces

Figure 13: AWS Configure Instance details 2

In stage 4, storage was allocated to the instance. An additional 90GB was allocated to the instance later on. All other settings were left default.



eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1#LaunchInstanceWizard:

Services · Resource Groups · [redacted] Ireland · Support

1. Choose AMI · 2. Choose Instance Type · 3. Configure Instance · 4. Add Storage · 5. Add Tags · 6. Configure Security Group · 7. Review

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-[redacted]	90	General Purpose SSD (gp2)	270 / 3000	N/A	☑	Not Encrypted
EBS	/dev/sdb	snap[redacted]	90	General Purpose SSD (gp2)	270 / 3000	N/A	☑	Not Encrypted

Add New Volume

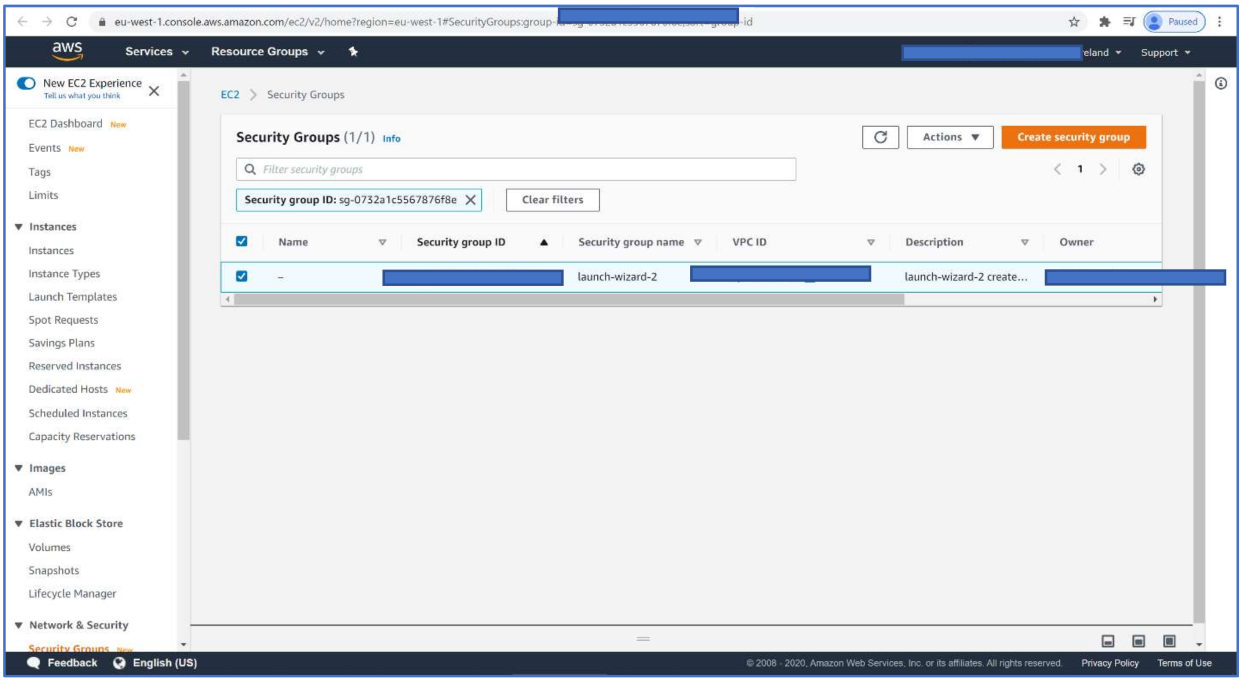
Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

Cancel · Previous · Review and Launch · Next: Add Tags

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

Figure 14: AWS Add storage

A security group had to be created to specify inbound rules to access Instance. Since Jupyter notebooks were to be accessed from the browser, https and http protocols were specified (the ubuntu shell by itself can't be used for interaction with jupyter notebooks, a browser is needed for this interactivity to be allowed). SSH was specified as well so that the ubuntu shell of the instance could be accessed – many programmes were to be run from the shell.



eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1#SecurityGroups:group-[redacted]-id

Services · Resource Groups · [redacted]eland · Support

New EC2 Experience
Tell us what you think

EC2 Dashboard New
Events New
Tags
Limits
▼ Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts New
Scheduled Instances
Capacity Reservations
▼ Images
AMIs
▼ Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
▼ Network & Security
Security Groups New

EC2 > Security Groups

Security Groups (1/1) Info · Actions · Create security group

Filter security groups

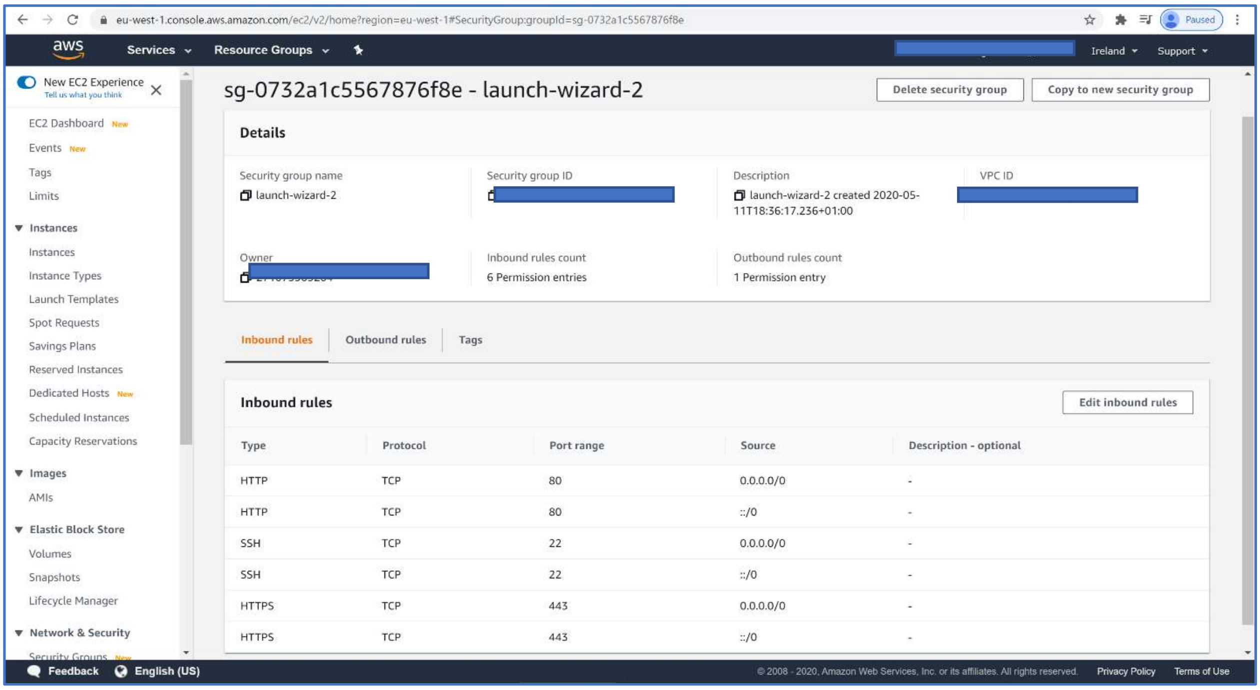
Security group ID: sg-0732a1c5567876f8e · Clear filters

☑	Name	Security group ID	Security group name	VPC ID	Description	Owner
☑	–	[redacted]	launch-wizard-2	[redacted]	launch-wizard-2 create...	[redacted]

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

Figure 15: AWS security group creation for protocol specifications

Rules I used for my own project, a port is allocated for ssh purposes, port used to ssh into the instance



sg-0732a1c5567876f8e - launch-wizard-2

Details

Security group name: launch-wizard-2

Security group ID: [redacted]

Description: launch-wizard-2 created 2020-05-11T18:36:17.236+01:00

VPC ID: [redacted]

Owner: [redacted]

Inbound rules count: 6 Permission entries

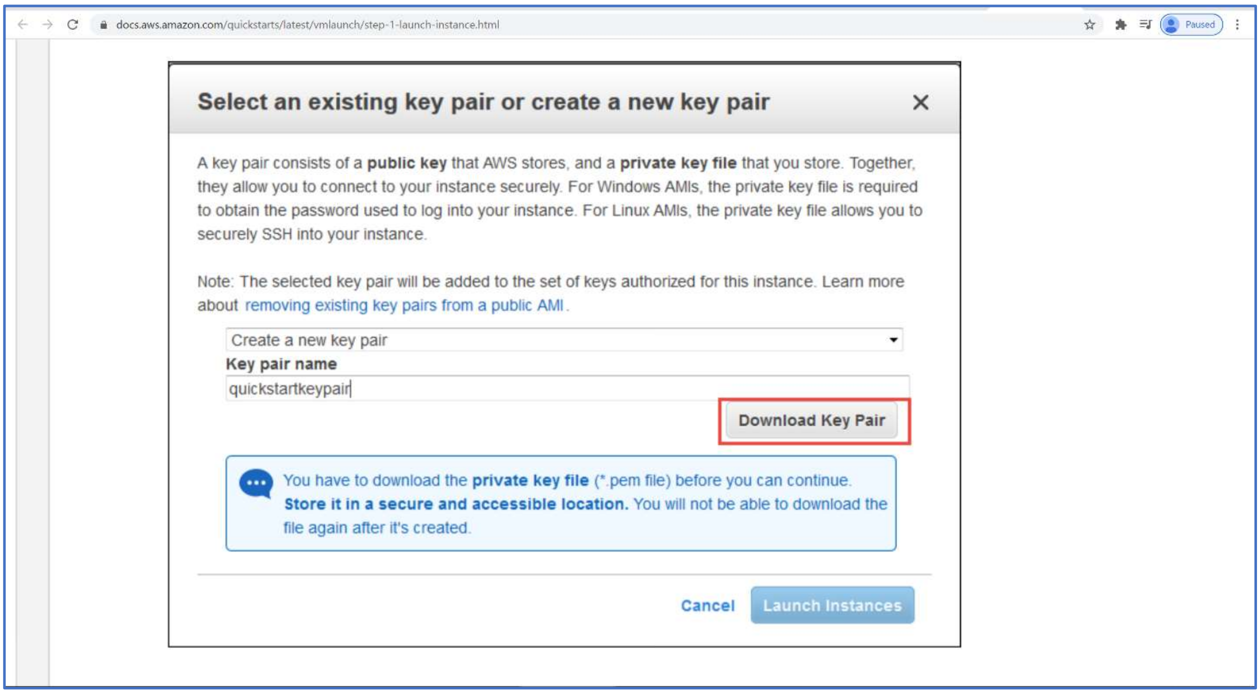
Outbound rules count: 1 Permission entry

Inbound rules

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	0.0.0.0/0	-
HTTP	TCP	80	::/0	-
SSH	TCP	22	0.0.0.0/0	-
SSH	TCP	22	::/0	-
HTTPS	TCP	443	0.0.0.0/0	-
HTTPS	TCP	443	::/0	-

Figure 16: AWS protocols for inbound rules regarding access of Instance

Upon reviewing and launching the instance, a private key was allocated to the instance and automatically downloaded. The instance was not accessed yet


docs.aws.amazon.com/quickstarts/latest/vmlaunch/step-1-launch-instance.html
Paused
Download Key Pair
Cancel
Launch Instances

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

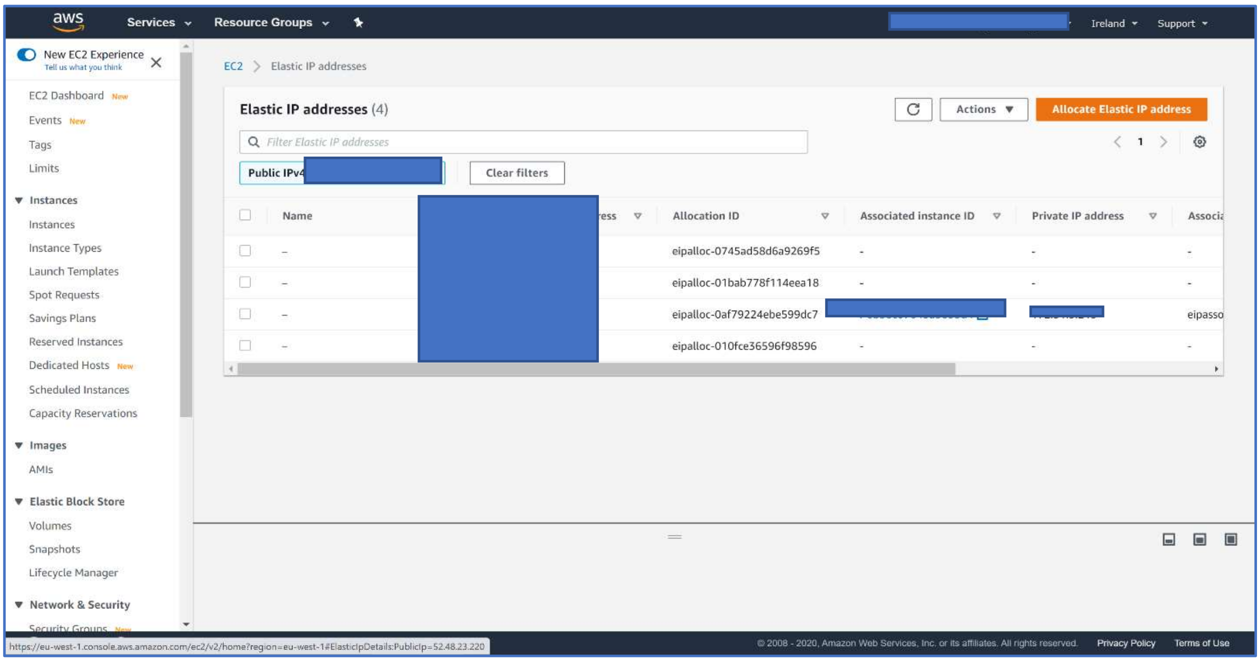
Key pair name

quickstartkeypair

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.

Figure 17: Creation of private hash key used to ssh into instance

Once stopped, an IP address from the subnet was then set to be an elastic IP address and was then allocated to the instance. Setting the ip address to static allowed for the instance to be accessed with the same ip address after the instance is stopped and started again. Nginx and supervisor programs (used for displaying the Jupyter notebook in the browser) would need to be reconfigured if IP address was not static.



Elastic IP addresses (4)

Public IPv4[redacted]

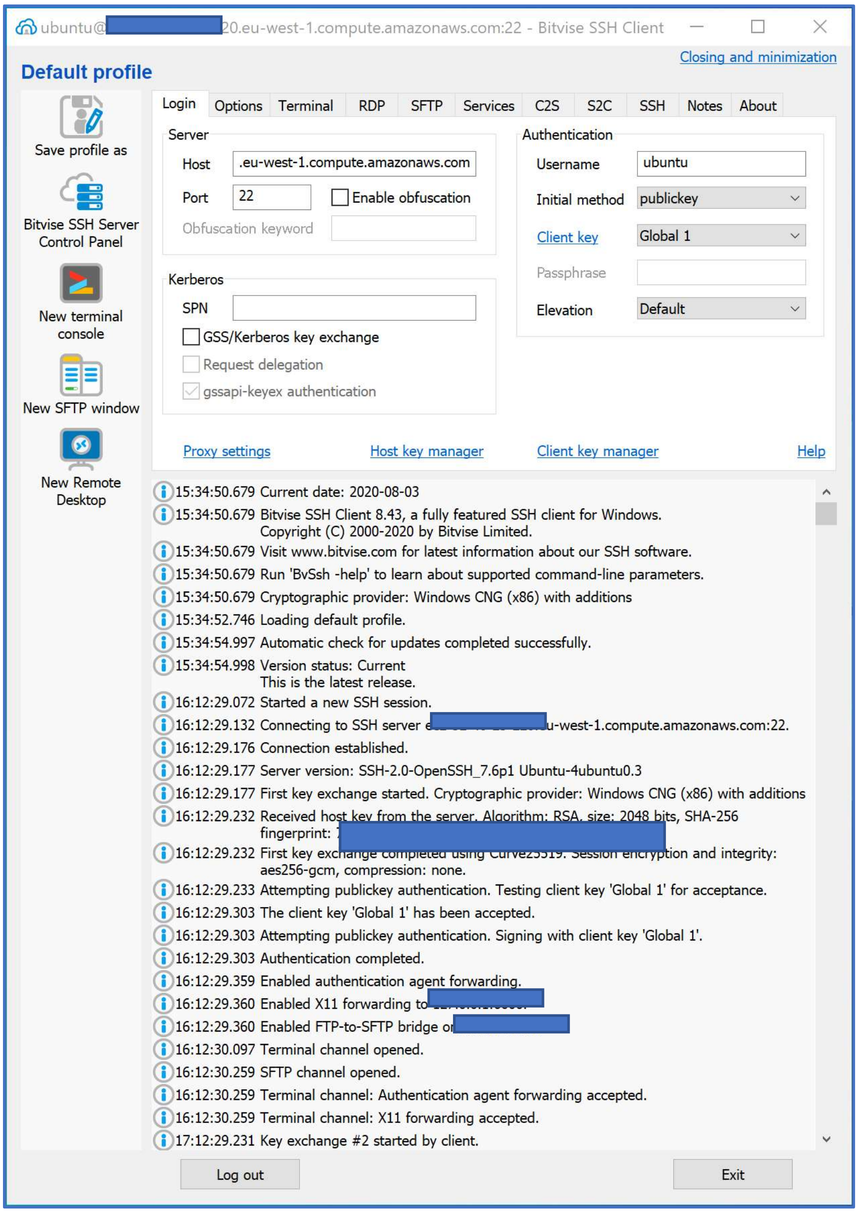
□	Name	[redacted]ess	Allocation ID	Associated instance ID	Private IP address	Associa
□	–	[redacted]	eipalloc-0745ad58d6a9269f5	-	-	-
□	–	[redacted]	eipalloc-01bab778f114eea18	-	-	-
□	–	[redacted]	eipalloc-0af79224ebe599dc7	[redacted]	[redacted]	eipasso
□	–	[redacted]	eipalloc-010fce36596f98596	-	-	-

https://eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1#ElasticIpDetails:PublicIp=52.48.23.220

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

Figure 18: Elastic IP address association

In order to conveniently access the EC2 instance via SSH, it was decided to use the client SSH software tool Bitvise as the tool also allowed for file transfer to the local computer. The host address and port was specified and the client key downloaded from Amazon was added for authentication purposes.



ubuntu@[redacted]20.eu-west-1.compute.amazonaws.com:22 - Bitvise SSH Client

Default profile

Login tab:

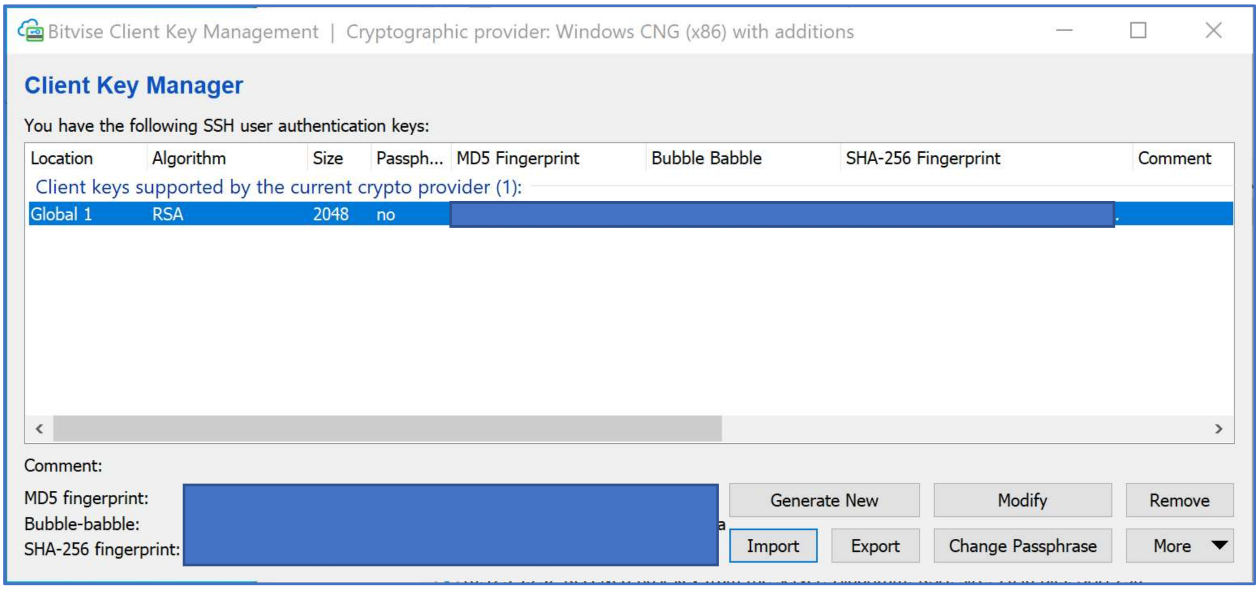
Server
- Host: .eu-west-1.compute.amazonaws.com
- Port: 22 □ Enable obfuscation
- Obfuscation keyword:

Kerberos
- SPN:
- □ GSS/Kerberos key exchange
- □ Request delegation
- ☑ gssapi-keyex authentication

Authentication
- Username: ubuntu
- Initial method: publickey
- Client key: Global 1
- Passphrase:
- Elevation: Default

```
15:34:50.679 Current date: 2020-08-03
15:34:50.679 Bitvise SSH Client 8.43, a fully featured SSH client for Windows.
             Copyright (C) 2000-2020 by Bitvise Limited.
15:34:50.679 Visit www.bitvise.com for latest information about our SSH software.
15:34:50.679 Run 'BvSsh -help' to learn about supported command-line parameters.
15:34:50.679 Cryptographic provider: Windows CNG (x86) with additions
15:34:52.746 Loading default profile.
15:34:54.997 Automatic check for updates completed successfully.
15:34:54.998 Version status: Current
             This is the latest release.
16:12:29.072 Started a new SSH session.
16:12:29.132 Connecting to SSH server [redacted]u-west-1.compute.amazonaws.com:22.
16:12:29.176 Connection established.
16:12:29.177 Server version: SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3
16:12:29.177 First key exchange started. Cryptographic provider: Windows CNG (x86) with additions
16:12:29.232 Received host key from the server. Algorithm: RSA, size: 2048 bits, SHA-256
             fingerprint: [redacted]
16:12:29.232 First key exch                                          ion and integrity:
             aes256-gcm, compression: none.
16:12:29.233 Attempting publickey authentication. Testing client key 'Global 1' for acceptance.
16:12:29.303 The client key 'Global 1' has been accepted.
16:12:29.303 Attempting publickey authentication. Signing with client key 'Global 1'.
16:12:29.303 Authentication completed.
16:12:29.359 Enabled authentication agent forwarding.
16:12:29.360 Enabled X11 forwarding to [redacted]
16:12:29.360 Enabled FTP-to-SFTP bridge o[redacted]
16:12:30.097 Terminal channel opened.
16:12:30.259 SFTP channel opened.
16:12:30.259 Terminal channel: Authentication agent forwarding accepted.
16:12:30.259 Terminal channel: X11 forwarding accepted.
17:12:29.231 Key exchange #2 started by client.
```

Figure 19: Using Bitvise to SSH into instance



Bitvise Client Key Management | Cryptographic provider: Windows CNG (x86) with additions

Client Key Manager

You have the following SSH user authentication keys:

Location	Algorithm	Size	Passph...	MD5 Fingerprint	Bubble Babble	SHA-256 Fingerprint	Comment
Client keys supported by the current crypto provider (1):							
Global 1	RSA	2048	no	[redacted]			

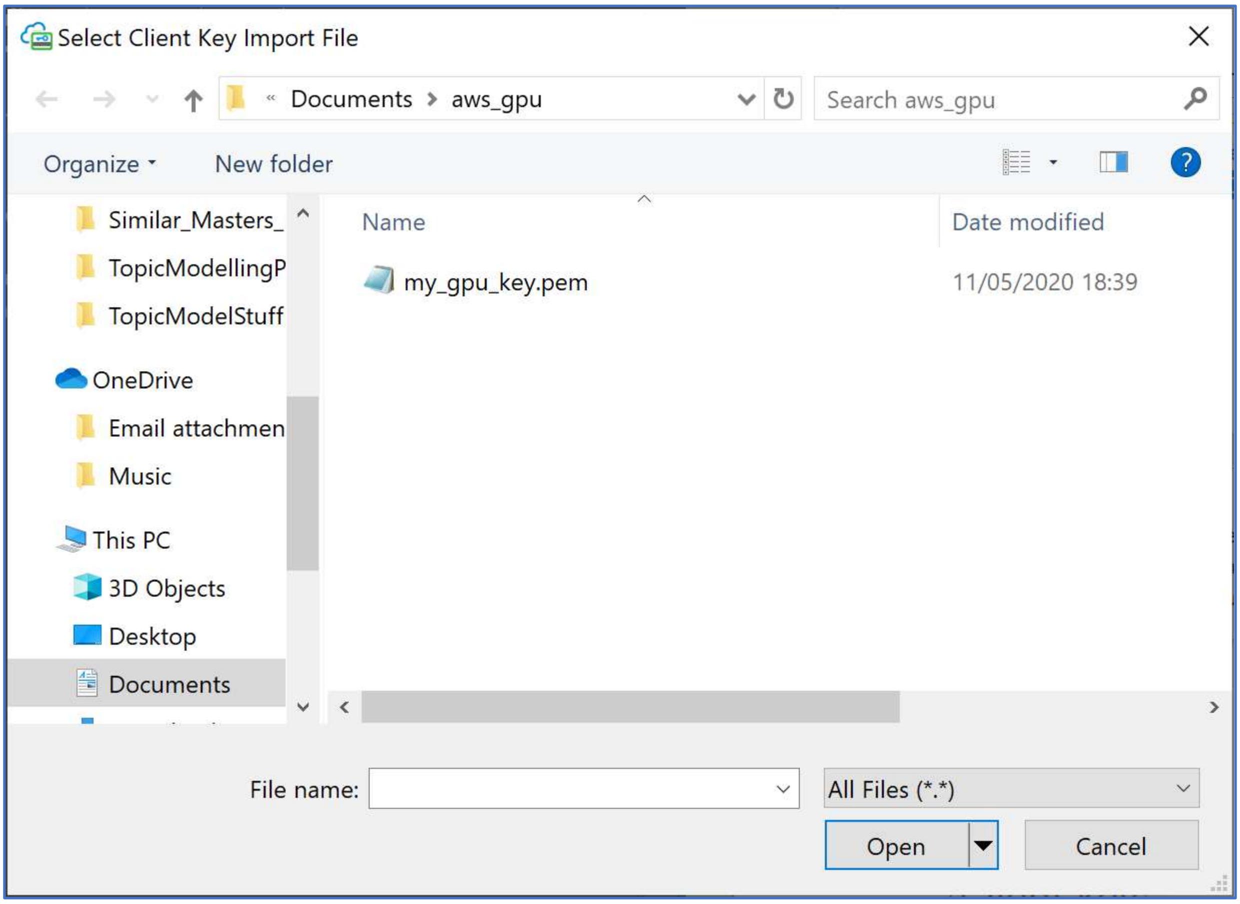
Comment:

MD5 fingerprint: [redacted]

Bubble-babble:

SHA-256 fingerprint:

Figure 20: Import PEM key into Bitvise



Select Client Key Import File

Documents › aws_gpu

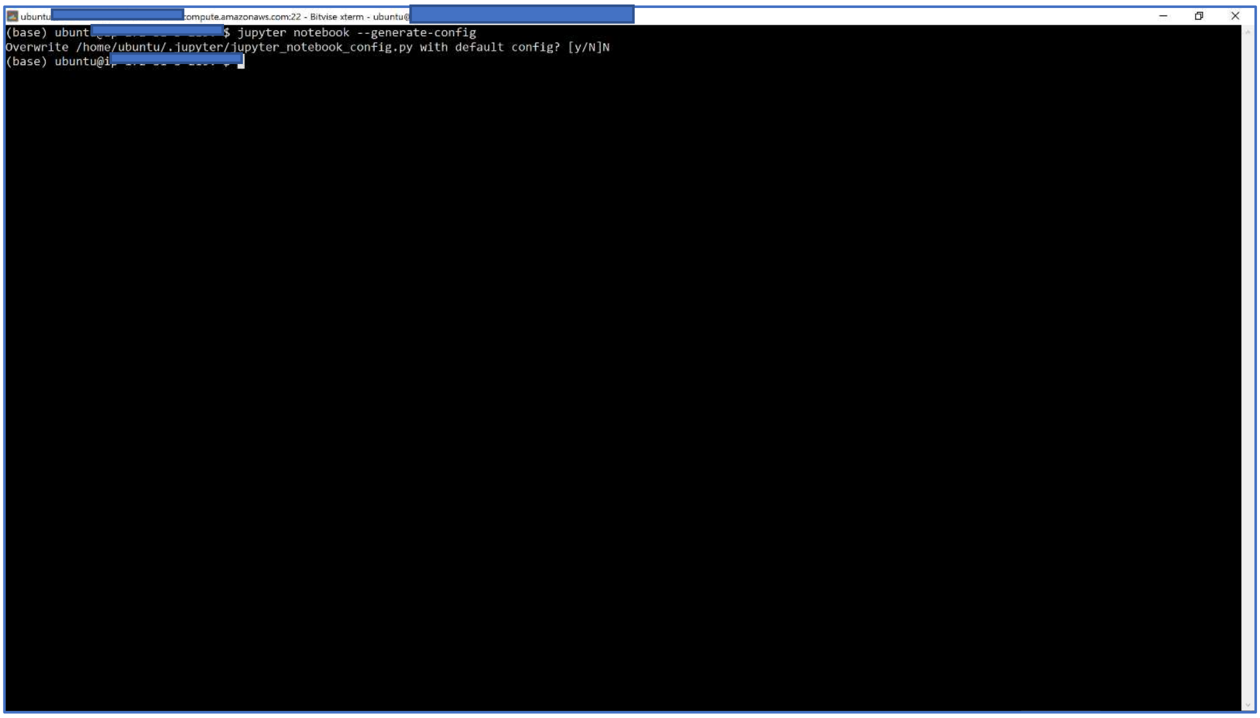
Search aws_gpu

Name	Date modified
my_gpu_key.pem	11/05/2020 18:39

File name:

All Files (*.*)

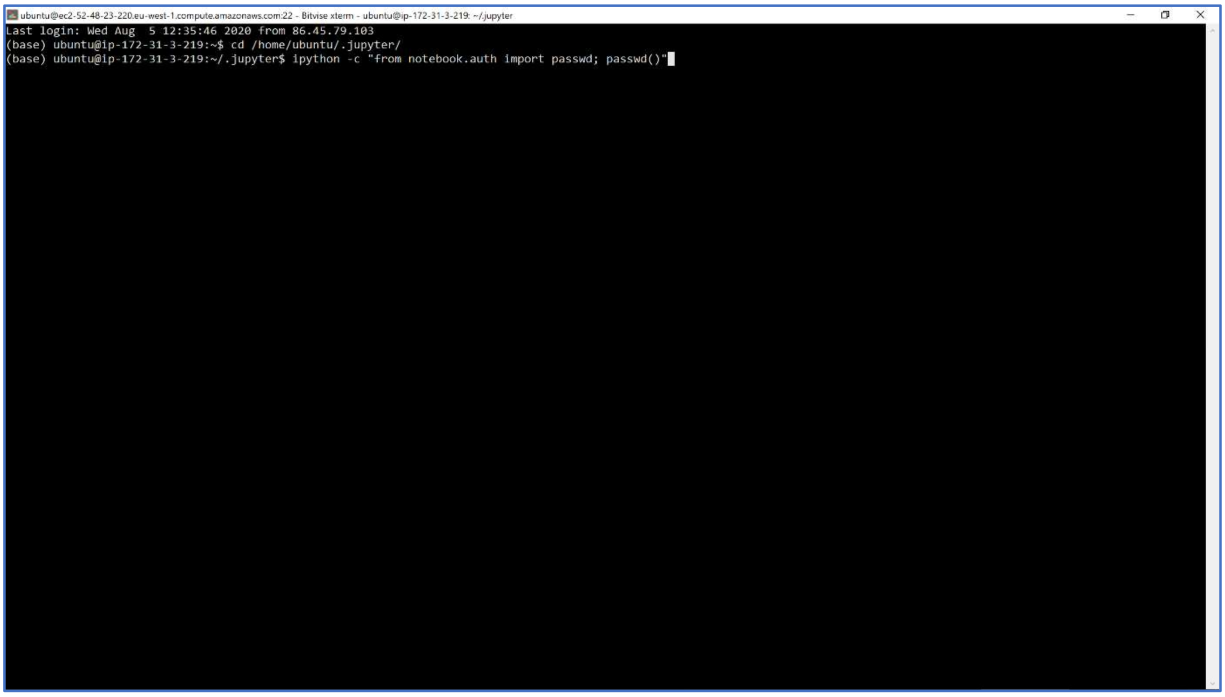
Figure 21: Key pair that was imported int Bitvise



```
ubuntu[redacted]compute.amazonaws.com:22 - Bitvise xterm - ubuntu@[redacted]
(base) ubunt[redacted]$ jupyter notebook --generate-config
Overwrite /home/ubuntu/.jupyter/jupyter_notebook_config.py with default config? [y/N]N
(base) ubuntu@i[redacted]
```

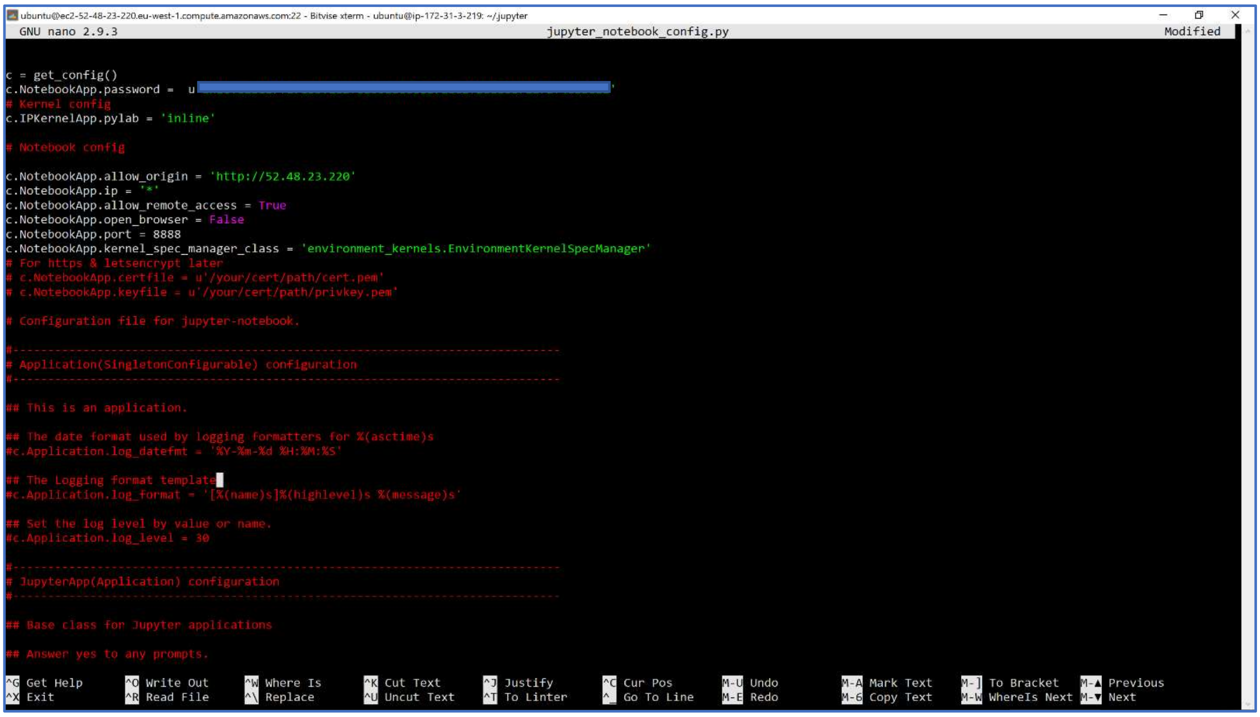
Figure 22: Jupyter notebook configuration initialisation

A hashed password that was generated is put into the config file the notebook. The network of the notebook is set to the ip address of the instance to prevent the unlikely sharing of resources with other external sites



```
ubuntu@ec2-52-48-23-220.eu-west-1.compute.amazonaws.com:22 - Bitvise xterm - ubuntu@ip-172-31-3-219: ~/.jupyter
Last login: Wed Aug  5 12:35:46 2020 from 86.45.79.103
(base) ubuntu@ip-172-31-3-219:~$ cd /home/ubuntu/.jupyter/
(base) ubuntu@ip-172-31-3-219:~/.jupyter$ ipython -c "from notebook.auth import passwd; passwd()"
```

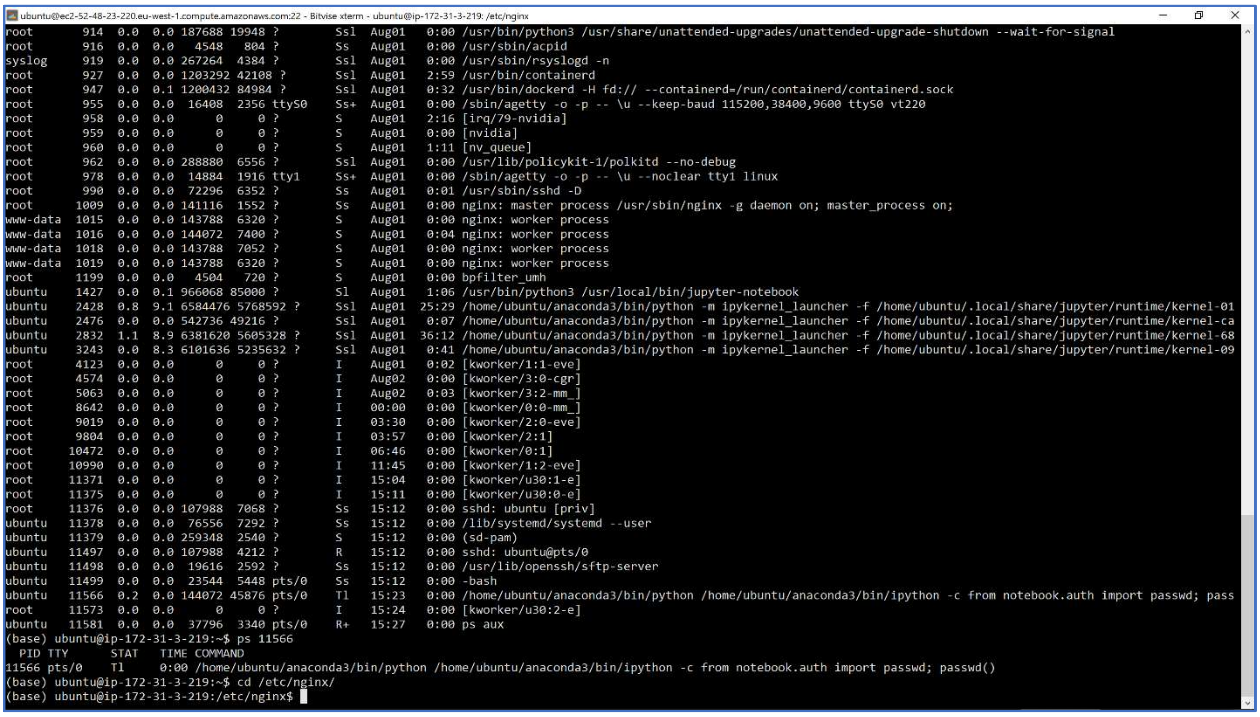
Figure 23: Generation of hashed password for notebook



```
ubuntu@ec2-52-48-23-220.eu-west-1.compute.amazonaws.com:22 - Bitvise xterm - ubuntu@ip-172-31-3-219: ~/jupyter
  GNU nano 2.9.3                                    jupyter_notebook_config.py                                    Modified


c = get_config()
c.NotebookApp.password =  u[redacted]'
# Kernel config
c.IPKernelApp.pylab = 'inline'

# Notebook config

c.NotebookApp.allow_origin = 'http://52.48.23.220'
c.NotebookApp.ip = '*'
c.NotebookApp.allow_remote_access = True
c.NotebookApp.open_browser = False
c.NotebookApp.port = 8888
c.NotebookApp.kernel_spec_manager_class = 'environment_kernels.EnvironmentKernelSpecManager'
# For https & letsencrypt later
# c.NotebookApp.certfile = u'/your/cert/path/cert.pem'
# c.NotebookApp.keyfile = u'/your/cert/path/privkey.pem'

# Configuration file for jupyter-notebook.

#------------------------------------------------------------------------------
# Application(SingletonConfigurable) configuration
#------------------------------------------------------------------------------

## This is an application.

## The date format used by logging formatters for %(asctime)s
#c.Application.log_datefmt = '%Y-%m-%d %H:%M:%S'

## The Logging format template
#c.Application.log_format = '[%(name)s]%(highlevel)s %(message)s'

## Set the log level by value or name.
#c.Application.log_level = 30

#------------------------------------------------------------------------------
# JupyterApp(Application) configuration
#------------------------------------------------------------------------------

## Base class for Jupyter applications

## Answer yes to any prompts.

^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos      M-U Undo        M-A Mark Text   M-] To Bracket  M-▲ Previous
^X Exit         ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Linter    ^_ Go To Line   M-E Redo        M-6 Copy Text   M-W WhereIs Next M-▼ Next
```

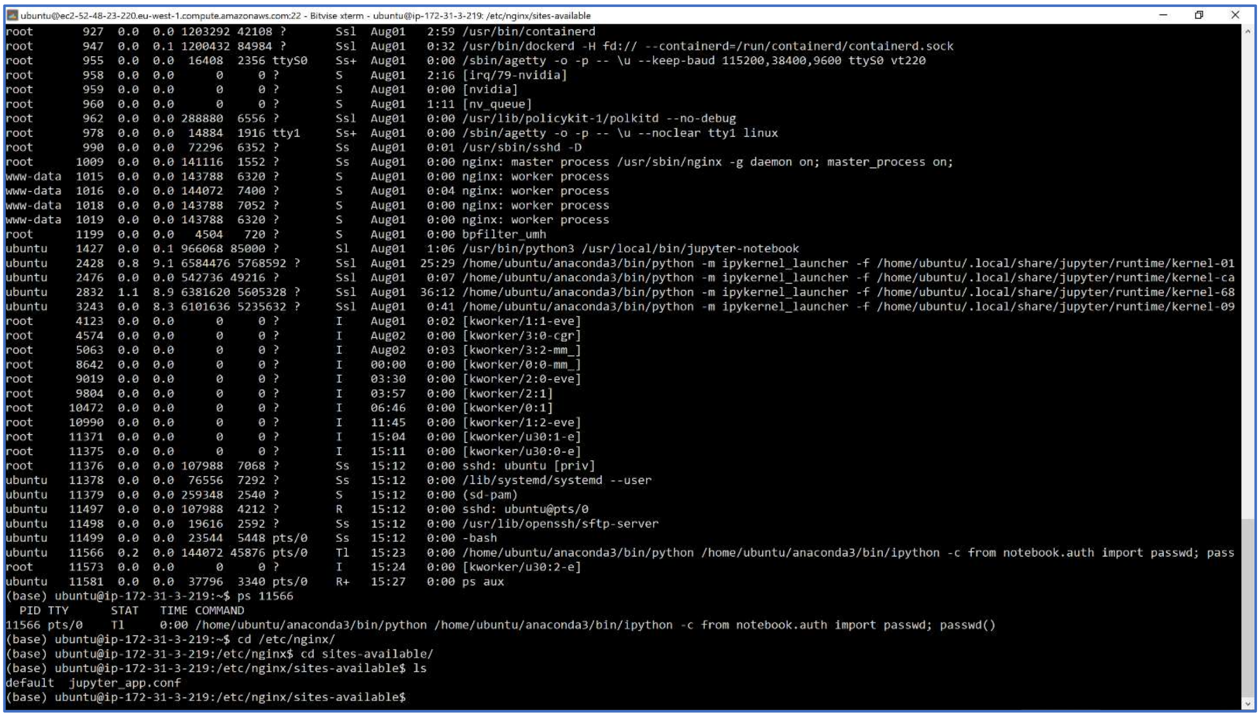
Figure 24: Hashed key notebook password and localhost Port to access Notebook

Nginx is now configured to display Jupyter notebook in the browser via the public ip address of the instance. The EC2 instance does not have visual rendering capabilities and thus   First the user must change directories to the sites-available under nginx directory.



```
ubuntu@ec2-52-48-23-220.eu-west-1.compute.amazonaws.com:22 - Bitvise xterm - ubuntu@ip-172-31-3-219: /etc/nginx
root       914  0.0  0.0 187688 19948 ?        Ssl  Aug01   0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
root       916  0.0  0.0   4548   804 ?        Ss   Aug01   0:00 /usr/sbin/acpid
syslog     919  0.0  0.0 267264  4384 ?        Ssl  Aug01   0:00 /usr/sbin/rsyslogd -n
root       927  0.0  0.0 1203292 42108 ?       Ssl  Aug01   2:59 /usr/bin/containerd
root       947  0.0  0.1 1200432 84984 ?       Ssl  Aug01   0:32 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
root       955  0.0  0.0  16408  2356 ttyS0    Ss+  Aug01   0:00 /sbin/agetty -o -p -- \u --keep-baud 115200,38400,9600 ttyS0 vt220
root       958  0.0  0.0      0     0 ?        S    Aug01   2:16 [irq/79-nvidia]
root       959  0.0  0.0      0     0 ?        S    Aug01   0:00 [nvidia]
root       960  0.0  0.0      0     0 ?        S    Aug01   1:11 [nv_queue]
root       962  0.0  0.0 288880  6556 ?        Ssl  Aug01   0:00 /usr/lib/policykit-1/polkitd --no-debug
root       978  0.0  0.0  14884  1916 tty1     Ss+  Aug01   0:00 /sbin/agetty -o -p -- \u --noclear tty1 linux
root       990  0.0  0.0  72296  6352 ?        Ss   Aug01   0:01 /usr/sbin/sshd -D
root      1009  0.0  0.0 141116  1552 ?        Ss   Aug01   0:00 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
www-data  1015  0.0  0.0 143788  6320 ?        S    Aug01   0:00 nginx: worker process
www-data  1016  0.0  0.0 144072  7400 ?        S    Aug01   0:04 nginx: worker process
www-data  1018  0.0  0.0 143788  7052 ?        S    Aug01   0:00 nginx: worker process
www-data  1019  0.0  0.0 143788  6320 ?        S    Aug01   0:00 nginx: worker process
root      1199  0.0  0.0   4504   720 ?        S    Aug01   0:00 bpfilter_umh
ubuntu    1427  0.0  0.1 966068 85000 ?        Sl   Aug01   1:06 /usr/bin/python3 /usr/local/bin/jupyter-notebook
ubuntu    2428  0.8  9.1 6584476 5768592 ?     Ssl  Aug01  25:29 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-01
ubuntu    2476  0.0  0.0 542736 49216 ?        Ssl  Aug01   0:07 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-ca
ubuntu    2832  1.1  8.9 6381620 5605328 ?     Ssl  Aug01  36:12 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-68
ubuntu    3243  0.0  8.3 6101636 5235632 ?     Ssl  Aug01   0:41 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-09
root      4123  0.0  0.0      0     0 ?        I    Aug01   0:02 [kworker/1:1-eve]
root      4574  0.0  0.0      0     0 ?        I    Aug02   0:00 [kworker/3:0-cgr]
root      5063  0.0  0.0      0     0 ?        I    Aug02   0:03 [kworker/3:2-mm_]
root      8642  0.0  0.0      0     0 ?        I    00:00   0:00 [kworker/0:0-mm_]
root      9019  0.0  0.0      0     0 ?        I    03:30   0:00 [kworker/2:0-eve]
root      9804  0.0  0.0      0     0 ?        I    03:57   0:00 [kworker/2:1]
root     10472  0.0  0.0      0     0 ?        I    06:46   0:00 [kworker/0:1]
root     10990  0.0  0.0      0     0 ?        I    11:45   0:00 [kworker/1:2-eve]
root     11371  0.0  0.0      0     0 ?        I    15:04   0:00 [kworker/u30:1-e]
root     11375  0.0  0.0      0     0 ?        I    15:11   0:00 [kworker/u30:0-e]
root     11376  0.0  0.0 107988  7068 ?        Ss   15:12   0:00 sshd: ubuntu [priv]
ubuntu   11378  0.0  0.0  76556  7292 ?        Ss   15:12   0:00 /lib/systemd/systemd --user
ubuntu   11379  0.0  0.0 259348  2540 ?        S    15:12   0:00 (sd-pam)
ubuntu   11497  0.0  0.0 107988  4212 ?        R    15:12   0:00 sshd: ubuntu@pts/0
ubuntu   11498  0.0  0.0  19616  2592 ?        Ss   15:12   0:00 /usr/lib/openssh/sftp-server
ubuntu   11499  0.0  0.0  23544  5448 pts/0    Ss   15:12   0:00 -bash
ubuntu   11566  0.2  0.0 144072 45876 pts/0    Tl   15:23   0:00 /home/ubuntu/anaconda3/bin/python /home/ubuntu/anaconda3/bin/ipython -c from notebook.auth import passwd; pass
root     11573  0.0  0.0      0     0 ?        I    15:24   0:00 [kworker/u30:2-e]
ubuntu   11581  0.0  0.0  37796  3340 pts/0    R+   15:27   0:00 ps aux
(base) ubuntu@ip-172-31-3-219:~$ ps 11566
  PID TTY      STAT   TIME COMMAND
11566 pts/0    Tl     0:00 /home/ubuntu/anaconda3/bin/python /home/ubuntu/anaconda3/bin/ipython -c from notebook.auth import passwd; passwd()
(base) ubuntu@ip-172-31-3-219:~$ cd /etc/nginx/
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ 
```

Figure 25: Nginx directory

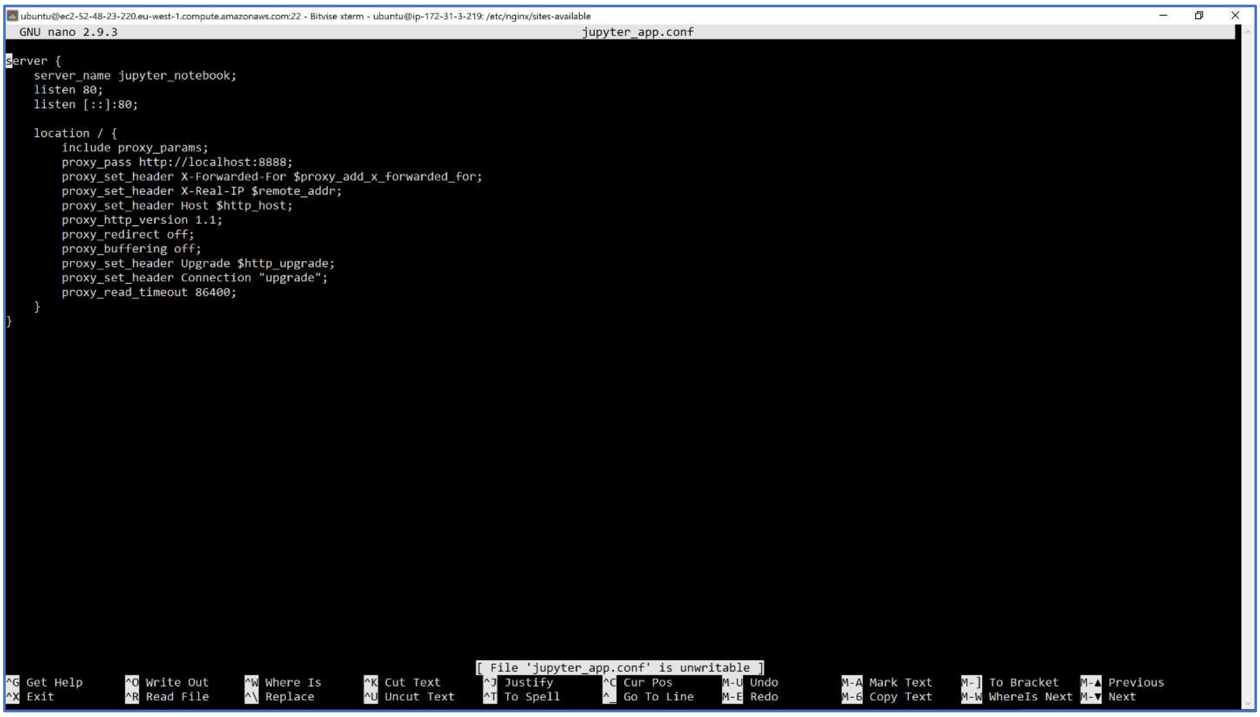


ubuntu@ec2-52-48-23-220.eu-west-1.compute.amazonaws.com:22 - Bitvise xterm - ubuntu@ip-172-31-3-219: /etc/nginx/sites-available

```
root       927  0.0  0.0 1203292 42108 ?       Ssl  Aug01   2:59 /usr/bin/containerd
root       947  0.0  0.1 1200432 84984 ?       Ssl  Aug01   0:32 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
root       955  0.0  0.0  16408  2356 ttyS0    Ss+  Aug01   0:00 /sbin/agetty -o -p -- \u --keep-baud 115200,38400,9600 ttyS0 vt220
root       958  0.0  0.0      0     0 ?        S    Aug01   2:16 [irq/79-nvidia]
root       959  0.0  0.0      0     0 ?        S    Aug01   0:00 [nvidia]
root       960  0.0  0.0      0     0 ?        S    Aug01   1:11 [nv_queue]
root       962  0.0  0.0 288880  6556 ?        Ssl  Aug01   0:00 /usr/lib/policykit-1/polkitd --no-debug
root       978  0.0  0.0  14884  1916 tty1     Ss+  Aug01   0:00 /sbin/agetty -o -p -- \u --noclear tty1 linux
root       990  0.0  0.0  72296  6352 ?        Ss   Aug01   0:01 /usr/sbin/sshd -D
root      1009  0.0  0.0 141116  1552 ?        Ss   Aug01   0:00 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
www-data  1015  0.0  0.0 143788  6320 ?        S    Aug01   0:00 nginx: worker process
www-data  1016  0.0  0.0 144072  7400 ?        S    Aug01   0:04 nginx: worker process
www-data  1018  0.0  0.0 143788  7052 ?        S    Aug01   0:00 nginx: worker process
www-data  1019  0.0  0.0 143788  6320 ?        S    Aug01   0:00 nginx: worker process
root      1199  0.0  0.0   4504   720 ?        S    Aug01   0:00 bpfilter_umh
ubuntu    1427  0.0  0.1 966068 85000 ?        Sl   Aug01   1:06 /usr/bin/python3 /usr/local/bin/jupyter-notebook
ubuntu    2428  0.8  9.1 6584476 5768592 ?     Ssl  Aug01  25:29 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-01
ubuntu    2476  0.0  0.0 542736 49216 ?        Ssl  Aug01   0:07 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-ca
ubuntu    2832  1.1  8.9 6381620 5605328 ?     Ssl  Aug01  36:12 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-68
ubuntu    3243  0.0  8.3 6101636 5235632 ?     Ssl  Aug01   0:41 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-09
root      4123  0.0  0.0      0     0 ?        I    Aug01   0:02 [kworker/1:1-eve]
root      4574  0.0  0.0      0     0 ?        I    Aug02   0:00 [kworker/3:0-cgr]
root      5063  0.0  0.0      0     0 ?        I    Aug02   0:03 [kworker/3:2-mm_]
root      8642  0.0  0.0      0     0 ?        I    00:00   0:00 [kworker/0:0-mm_]
root      9019  0.0  0.0      0     0 ?        I    03:30   0:00 [kworker/2:0-eve]
root      9804  0.0  0.0      0     0 ?        I    03:57   0:00 [kworker/2:1]
root     10472  0.0  0.0      0     0 ?        I    06:46   0:00 [kworker/0:1]
root     10990  0.0  0.0      0     0 ?        I    11:45   0:00 [kworker/1:2-eve]
root     11371  0.0  0.0      0     0 ?        I    15:04   0:00 [kworker/u30:1-e]
root     11375  0.0  0.0      0     0 ?        I    15:11   0:00 [kworker/u30:0-e]
root     11376  0.0  0.0 107988  7068 ?        Ss   15:12   0:00 sshd: ubuntu [priv]
ubuntu   11378  0.0  0.0  76556  7292 ?        Ss   15:12   0:00 /lib/systemd/systemd --user
ubuntu   11379  0.0  0.0 259348  2540 ?        S    15:12   0:00 (sd-pam)
ubuntu   11497  0.0  0.0 107988  4212 ?        R    15:12   0:00 sshd: ubuntu@pts/0
ubuntu   11498  0.0  0.0  19616  2592 ?        Ss   15:12   0:00 /usr/lib/openssh/sftp-server
ubuntu   11499  0.0  0.0  23544  5448 pts/0    Ss   15:12   0:00 -bash
ubuntu   11566  0.2  0.0 144072 45876 pts/0    Tl   15:23   0:00 /home/ubuntu/anaconda3/bin/python /home/ubuntu/anaconda3/bin/ipython -c from notebook.auth import passwd; pass
root     11573  0.0  0.0      0     0 ?        I    15:24   0:00 [kworker/u30:2-e]
ubuntu   11581  0.0  0.0  37796  3340 pts/0    R+   15:27   0:00 ps aux
(base) ubuntu@ip-172-31-3-219:~$ ps 11566
  PID TTY      STAT   TIME COMMAND
11566 pts/0    Tl     0:00 /home/ubuntu/anaconda3/bin/python /home/ubuntu/anaconda3/bin/ipython -c from notebook.auth import passwd; passwd()
(base) ubuntu@ip-172-31-3-219:~$ cd /etc/nginx/
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ cd sites-available/
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ ls
default  jupyter_app.conf
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$
```

Figure 26: Configuration file created for nginx

Specify the listening ports of the ec2 instance to allow nginx to display on the localhost port 8888 what is being received on ec2 port 80. The Jupyter notebook (which was configured to render visually on port 8888) can be rendered visually through the proxy server nginx on localhost port 8888

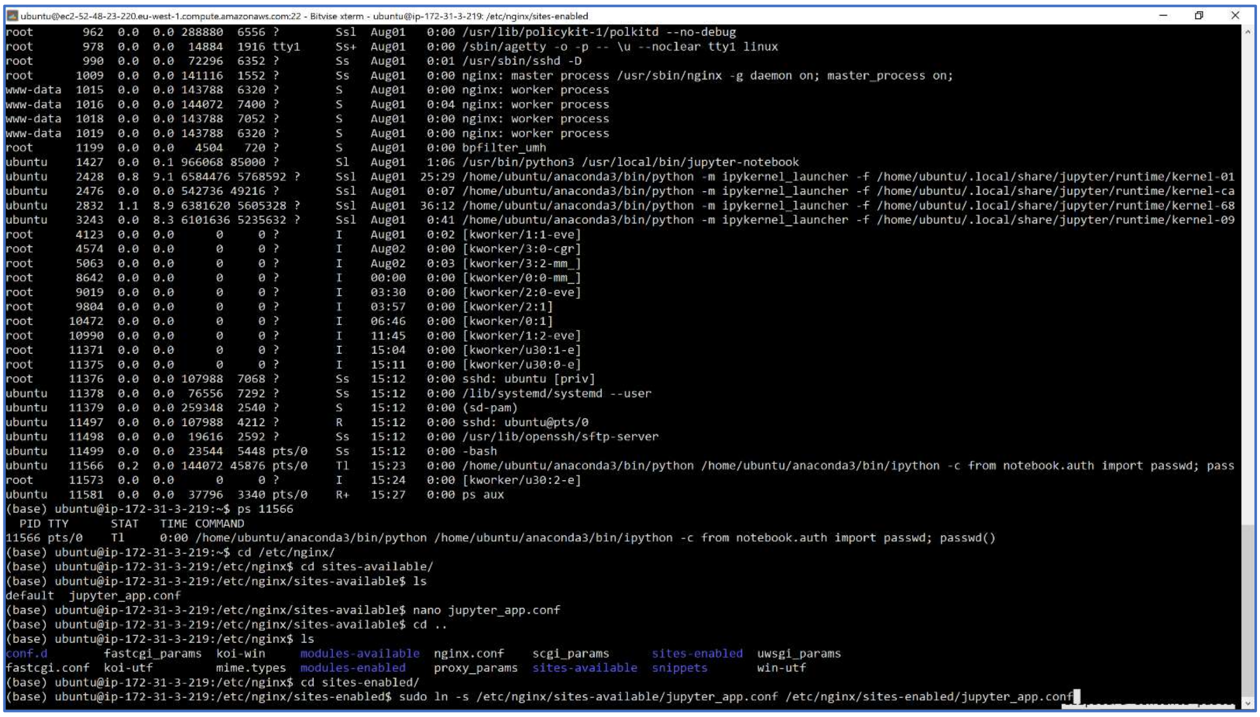


ubuntu@ec2-52-48-23-220.eu-west-1.compute.amazonaws.com:22 - Bitvise xterm - ubuntu@ip-172-31-3-219: /etc/nginx/sites-available

```
  GNU nano 2.9.3                                                    jupyter_app.conf

server {
    server_name jupyter_notebook;
    listen 80;
    listen [::]:80;

    location / {
        include proxy_params;
        proxy_pass http://localhost:8888;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header Host $http_host;
        proxy_http_version 1.1;
        proxy_redirect off;
        proxy_buffering off;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_read_timeout 86400;
    }
}

                                  [ File 'jupyter_app.conf' is unwritable ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text   M-] To Bracket   M-▲ Previous
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^_ Go To Line  M-E Redo       M-6 Copy Text   M-W WhereIs Next M-▼ Next
```

Figure 27: Port specification for nginx Jupyter notebook rendering

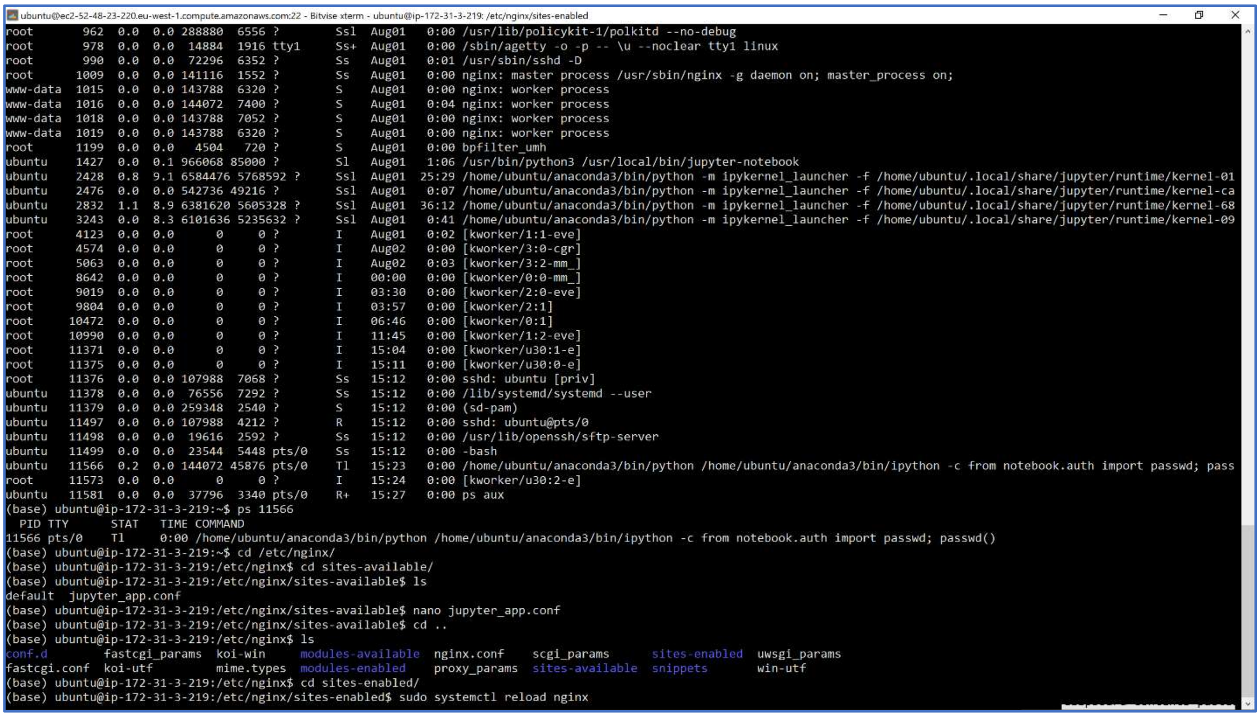
Once the sites-available file has been saved, the file can be copies to the sites-enabled directory and a symbolic link can be used to create a link between jupyter_app.conf in both sites-available and sites-enabled directories.


ubuntu@ec2-52-48-23-220.eu-west-1.compute.amazonaws.com:22 - Bitvise xterm - ubuntu@ip-172-31-3-219: /etc/nginx/sites-enabled

```
root       962  0.0  0.0 288880  6556 ?        Ssl  Aug01   0:00 /usr/lib/policykit-1/polkitd --no-debug
root       978  0.0  0.0  14884  1916 tty1     Ss+  Aug01   0:00 /sbin/agetty -o -p -- \u --noclear tty1 linux
root       990  0.0  0.0  72296  6352 ?        Ss   Aug01   0:01 /usr/sbin/sshd -D
root      1009  0.0  0.0 141116  1552 ?        Ss   Aug01   0:00 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
www-data  1015  0.0  0.0 143788  6320 ?        S    Aug01   0:00 nginx: worker process
www-data  1016  0.0  0.0 144072  7400 ?        S    Aug01   0:04 nginx: worker process
www-data  1018  0.0  0.0 143788  7052 ?        S    Aug01   0:00 nginx: worker process
www-data  1019  0.0  0.0 143788  6320 ?        S    Aug01   0:00 nginx: worker process
root      1199  0.0  0.0   4504   720 ?        S    Aug01   0:00 bpfilter_umh
ubuntu    1427  0.0  0.1 966068 85000 ?        Sl   Aug01   1:06 /usr/bin/python3 /usr/local/bin/jupyter-notebook
ubuntu    2428  0.8  9.1 6584476 5768592 ?     Ssl  Aug01  25:29 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-01
ubuntu    2476  0.0  0.0 542736 49216 ?        Ssl  Aug01   0:07 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-ca
ubuntu    2832  1.1  8.9 6381620 5605328 ?     Ssl  Aug01  36:12 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-68
ubuntu    3243  0.0  8.3 6101636 5235632 ?     Ssl  Aug01   0:41 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-09
root      4123  0.0  0.0      0     0 ?        I    Aug01   0:02 [kworker/1:1-eve]
root      4574  0.0  0.0      0     0 ?        I    Aug02   0:00 [kworker/3:0-cgr]
root      5063  0.0  0.0      0     0 ?        I    Aug02   0:03 [kworker/3:2-mm_]
root      8642  0.0  0.0      0     0 ?        I    00:00   0:00 [kworker/0:0-mm_]
root      9019  0.0  0.0      0     0 ?        I    03:30   0:00 [kworker/2:0-eve]
root      9804  0.0  0.0      0     0 ?        I    03:57   0:00 [kworker/2:1]
root     10472  0.0  0.0      0     0 ?        I    06:46   0:00 [kworker/0:1]
root     10990  0.0  0.0      0     0 ?        I    11:45   0:00 [kworker/1:2-eve]
root     11371  0.0  0.0      0     0 ?        I    15:04   0:00 [kworker/u30:1-e]
root     11375  0.0  0.0      0     0 ?        I    15:11   0:00 [kworker/u30:0-e]
root     11376  0.0  0.0 107988  7068 ?        Ss   15:12   0:00 sshd: ubuntu [priv]
ubuntu   11378  0.0  0.0  76556  7292 ?        Ss   15:12   0:00 /lib/systemd/systemd --user
ubuntu   11379  0.0  0.0 259348  2540 ?        S    15:12   0:00 (sd-pam)
ubuntu   11497  0.0  0.0 107988  4212 ?        R    15:12   0:00 sshd: ubuntu@pts/0
ubuntu   11498  0.0  0.0  19616  2592 ?        Ss   15:12   0:00 /usr/lib/openssh/sftp-server
ubuntu   11499  0.0  0.0  23544  5448 pts/0    Ss   15:12   0:00 -bash
ubuntu   11566  0.2  0.0 144072 45876 pts/0    Tl   15:23   0:00 /home/ubuntu/anaconda3/bin/python /home/ubuntu/anaconda3/bin/ipython -c from notebook.auth import passwd; pass
root     11573  0.0  0.0      0     0 ?        I    15:24   0:00 [kworker/u30:2-e]
ubuntu   11581  0.0  0.0  37796  3340 pts/0    R+   15:27   0:00 ps aux
(base) ubuntu@ip-172-31-3-219:~$ ps 11566
  PID TTY      STAT   TIME COMMAND
11566 pts/0    Tl     0:00 /home/ubuntu/anaconda3/bin/python /home/ubuntu/anaconda3/bin/ipython -c from notebook.auth import passwd; passwd()
(base) ubuntu@ip-172-31-3-219:~$ cd /etc/nginx/
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ cd sites-available/
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ ls
default  jupyter_app.conf
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ nano jupyter_app.conf
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ cd ..
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ ls
conf.d        fastcgi_params  koi-win     modules-available  nginx.conf    scgi_params      sites-enabled  uwsgi_params
fastcgi.conf  koi-utf         mime.types  modules-enabled    proxy_params  sites-available  snippets       win-utf
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ cd sites-enabled/
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-enabled$ sudo ln -s /etc/nginx/sites-available/jupyter_app.conf /etc/nginx/sites-enabled/jupyter_app.conf
```

Figure 28: Symbolic link creation between sites available and sites-enabled

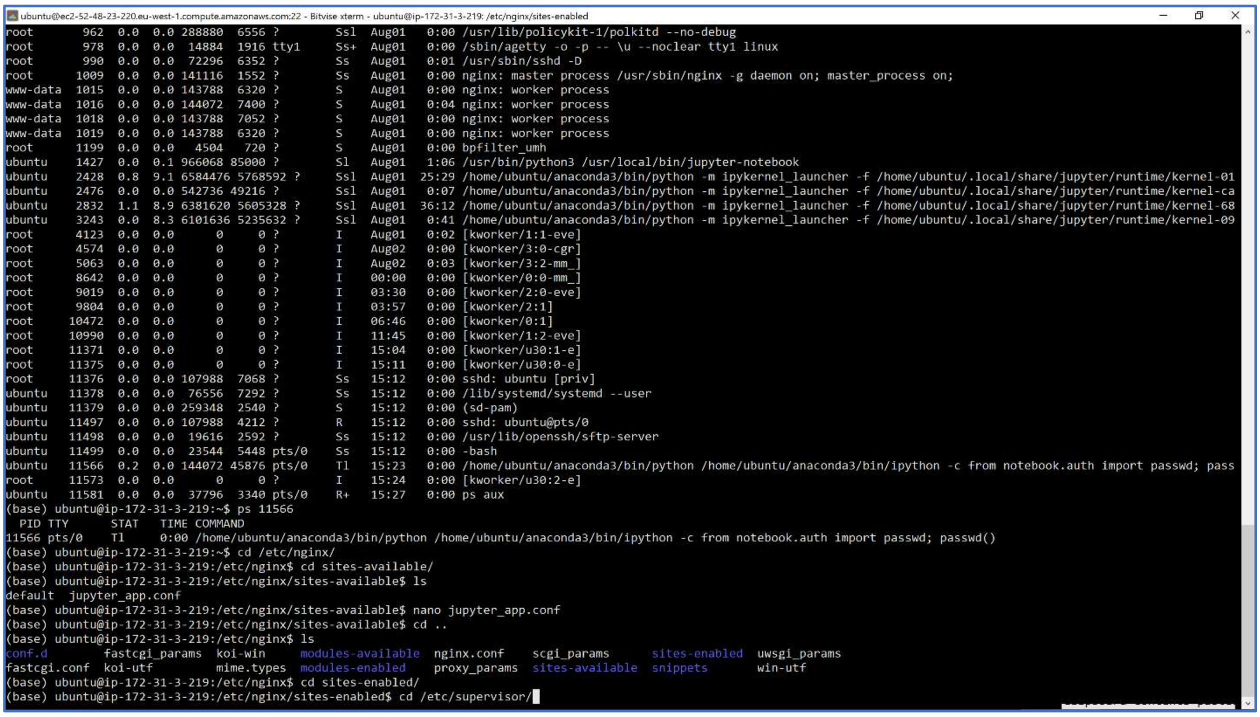
Command used to reload and update nginx configuration settings to recognise the new site configuration.


ubuntu@ec2-52-48-23-220.eu-west-1.compute.amazonaws.com:22 - Bitvise xterm - ubuntu@ip-172-31-3-219: /etc/nginx/sites-enabled

```
root       962  0.0  0.0 288880  6556 ?        Ssl  Aug01   0:00 /usr/lib/policykit-1/polkitd --no-debug
root       978  0.0  0.0  14884  1916 tty1     Ss+  Aug01   0:00 /sbin/agetty -o -p -- \u --noclear tty1 linux
root       990  0.0  0.0  72296  6352 ?        Ss   Aug01   0:01 /usr/sbin/sshd -D
root      1009  0.0  0.0 141116  1552 ?        Ss   Aug01   0:00 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
www-data  1015  0.0  0.0 143788  6320 ?        S    Aug01   0:00 nginx: worker process
www-data  1016  0.0  0.0 144072  7400 ?        S    Aug01   0:04 nginx: worker process
www-data  1018  0.0  0.0 143788  7052 ?        S    Aug01   0:00 nginx: worker process
www-data  1019  0.0  0.0 143788  6320 ?        S    Aug01   0:00 nginx: worker process
root      1199  0.0  0.0   4504   720 ?        S    Aug01   0:00 bpfilter_umh
ubuntu    1427  0.0  0.1 966068 85000 ?        Sl   Aug01   1:06 /usr/bin/python3 /usr/local/bin/jupyter-notebook
ubuntu    2428  0.8  9.1 6584476 5768592 ?     Ssl  Aug01  25:29 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-01
ubuntu    2476  0.0  0.0 542736 49216 ?        Ssl  Aug01   0:07 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-ca
ubuntu    2832  1.1  8.9 6381620 5605328 ?     Ssl  Aug01  36:12 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-68
ubuntu    3243  0.0  8.3 6101636 5235632 ?     Ssl  Aug01   0:41 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-09
root      4123  0.0  0.0      0     0 ?        I    Aug01   0:02 [kworker/1:1-eve]
root      4574  0.0  0.0      0     0 ?        I    Aug02   0:00 [kworker/3:0-cgr]
root      5063  0.0  0.0      0     0 ?        I    Aug02   0:03 [kworker/3:2-mm_]
root      8642  0.0  0.0      0     0 ?        I    00:00   0:00 [kworker/0:0-mm_]
root      9019  0.0  0.0      0     0 ?        I    03:30   0:00 [kworker/2:0-eve]
root      9804  0.0  0.0      0     0 ?        I    03:57   0:00 [kworker/2:1]
root     10472  0.0  0.0      0     0 ?        I    06:46   0:00 [kworker/0:1]
root     10990  0.0  0.0      0     0 ?        I    11:45   0:00 [kworker/1:2-eve]
root     11371  0.0  0.0      0     0 ?        I    15:04   0:00 [kworker/u30:1-e]
root     11375  0.0  0.0      0     0 ?        I    15:11   0:00 [kworker/u30:0-e]
root     11376  0.0  0.0 107988  7068 ?        Ss   15:12   0:00 sshd: ubuntu [priv]
ubuntu   11378  0.0  0.0  76556  7292 ?        Ss   15:12   0:00 /lib/systemd/systemd --user
ubuntu   11379  0.0  0.0 259348  2540 ?        S    15:12   0:00 (sd-pam)
ubuntu   11497  0.0  0.0 107988  4212 ?        R    15:12   0:00 sshd: ubuntu@pts/0
ubuntu   11498  0.0  0.0  19616  2592 ?        Ss   15:12   0:00 /usr/lib/openssh/sftp-server
ubuntu   11499  0.0  0.0  23544  5448 pts/0    Ss   15:12   0:00 -bash
ubuntu   11566  0.2  0.0 144072 45876 pts/0    Tl   15:23   0:00 /home/ubuntu/anaconda3/bin/python /home/ubuntu/anaconda3/bin/ipython -c from notebook.auth import passwd; pass
root     11573  0.0  0.0      0     0 ?        I    15:24   0:00 [kworker/u30:2-e]
ubuntu   11581  0.0  0.0  37796  3340 pts/0    R+   15:27   0:00 ps aux
(base) ubuntu@ip-172-31-3-219:~$ ps 11566
  PID TTY      STAT   TIME COMMAND
11566 pts/0    Tl     0:00 /home/ubuntu/anaconda3/bin/python /home/ubuntu/anaconda3/bin/ipython -c from notebook.auth import passwd; passwd()
(base) ubuntu@ip-172-31-3-219:~$ cd /etc/nginx/
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ cd sites-available/
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ ls
default  jupyter_app.conf
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ nano jupyter_app.conf
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ cd ..
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ ls
conf.d         fastcgi_params  koi-win     modules-available  nginx.conf    scgi_params      sites-enabled  uwsgi_params
fastcgi.conf   koi-utf         mime.types  modules-enabled    proxy_params  sites-available  snippets       win-utf
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ cd sites-enabled/
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-enabled$ sudo systemctl reload nginx
```

Figure 29: Updating of nginx sites available / sites enabled

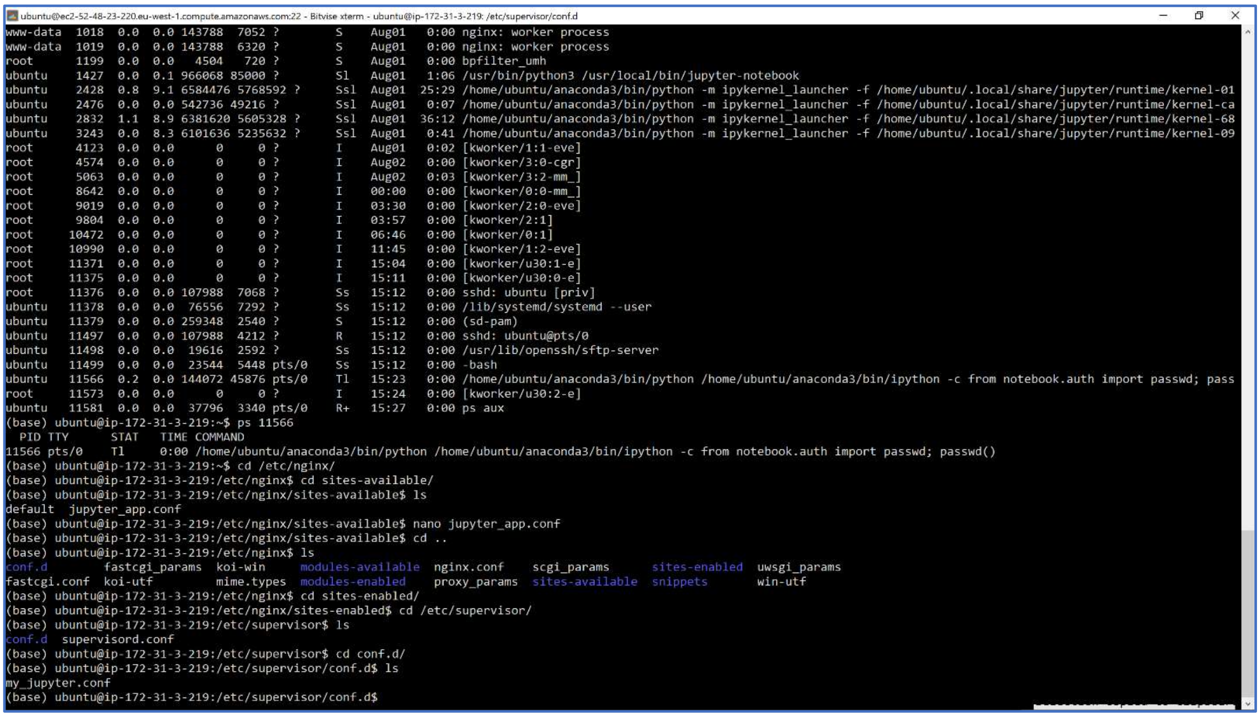
Jupyter notebook can now be run on local host. However, we still need to ssh into the instance's ubuntu terminal and start jupyter notebook before accessing the notebook via localhost with the browser. It is more convenient to configure the notebook to run in the background of the instance. For this, supervisor will be needed.


ubuntu@ec2-52-48-23-220.eu-west-1.compute.amazonaws.com:22 - Bitvise xterm - ubuntu@ip-172-31-3-219: /etc/nginx/sites-enabled

```
root       962  0.0  0.0 288880  6556 ?        Ssl  Aug01   0:00 /usr/lib/policykit-1/polkitd --no-debug
root       978  0.0  0.0  14884  1916 tty1     Ss+  Aug01   0:00 /sbin/agetty -o -p -- \u --noclear tty1 linux
root       990  0.0  0.0  72296  6352 ?        Ss   Aug01   0:01 /usr/sbin/sshd -D
root      1009  0.0  0.0 141116  1552 ?        Ss   Aug01   0:00 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
www-data  1015  0.0  0.0 143788  6320 ?        S    Aug01   0:00 nginx: worker process
www-data  1016  0.0  0.0 144072  7400 ?        S    Aug01   0:04 nginx: worker process
www-data  1018  0.0  0.0 143788  7052 ?        S    Aug01   0:00 nginx: worker process
www-data  1019  0.0  0.0 143788  6320 ?        S    Aug01   0:00 nginx: worker process
root      1199  0.0  0.0   4504   720 ?        S    Aug01   0:00 bpfilter_umh
ubuntu    1427  0.0  0.1 966068 85000 ?        Sl   Aug01   1:06 /usr/bin/python3 /usr/local/bin/jupyter-notebook
ubuntu    2428  0.8  9.1 6584476 5768592 ?     Ssl  Aug01  25:29 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-01
ubuntu    2476  0.0  0.0 542736 49216 ?        Ssl  Aug01   0:07 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-ca
ubuntu    2832  1.1  8.9 6381620 5605328 ?     Ssl  Aug01  36:12 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-68
ubuntu    3243  0.0  8.3 6101636 5235632 ?     Ssl  Aug01   0:41 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-09
root      4123  0.0  0.0      0     0 ?        I    Aug01   0:02 [kworker/1:1-eve]
root      4574  0.0  0.0      0     0 ?        I    Aug02   0:00 [kworker/3:0-cgr]
root      5063  0.0  0.0      0     0 ?        I    Aug02   0:03 [kworker/3:2-mm_]
root      8642  0.0  0.0      0     0 ?        I    00:00   0:00 [kworker/0:0-mm_]
root      9019  0.0  0.0      0     0 ?        I    03:30   0:00 [kworker/2:0-eve]
root      9804  0.0  0.0      0     0 ?        I    03:57   0:00 [kworker/2:1]
root     10472  0.0  0.0      0     0 ?        I    06:46   0:00 [kworker/0:1]
root     10990  0.0  0.0      0     0 ?        I    11:45   0:00 [kworker/1:2-eve]
root     11371  0.0  0.0      0     0 ?        I    15:04   0:00 [kworker/u30:1-e]
root     11375  0.0  0.0      0     0 ?        I    15:11   0:00 [kworker/u30:0-e]
root     11376  0.0  0.0 107988  7068 ?        Ss   15:12   0:00 sshd: ubuntu [priv]
ubuntu   11378  0.0  0.0  76556  7292 ?        Ss   15:12   0:00 /lib/systemd/systemd --user
ubuntu   11379  0.0  0.0 259348  2540 ?        S    15:12   0:00 (sd-pam)
ubuntu   11497  0.0  0.0 107988  4212 ?        R    15:12   0:00 sshd: ubuntu@pts/0
ubuntu   11498  0.0  0.0  19616  2592 ?        Ss   15:12   0:00 /usr/lib/openssh/sftp-server
ubuntu   11499  0.0  0.0  23544  5448 pts/0    Ss   15:12   0:00 -bash
ubuntu   11566  0.2  0.0 144072 45876 pts/0    Tl   15:23   0:00 /home/ubuntu/anaconda3/bin/python /home/ubuntu/anaconda3/bin/ipython -c from notebook.auth import passwd; pass
root     11573  0.0  0.0      0     0 ?        I    15:24   0:00 [kworker/u30:2-e]
ubuntu   11581  0.0  0.0  37796  3340 pts/0    R+   15:27   0:00 ps aux
(base) ubuntu@ip-172-31-3-219:~$ ps 11566
  PID TTY      STAT   TIME COMMAND
11566 pts/0    Tl     0:00 /home/ubuntu/anaconda3/bin/python /home/ubuntu/anaconda3/bin/ipython -c from notebook.auth import passwd; passwd()
(base) ubuntu@ip-172-31-3-219:~$ cd /etc/nginx/
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ cd sites-available/
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ ls
default  jupyter_app.conf
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ nano jupyter_app.conf
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ cd ..
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ ls
conf.d         fastcgi_params  koi-win     modules-available  nginx.conf    scgi_params      sites-enabled  uwsgi_params
fastcgi.conf   koi-utf         mime.types  modules-enabled    proxy_params  sites-available  snippets       win-utf
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ cd sites-enabled/
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-enabled$ cd /etc/supervisor/
```

Figure 30: Configuration moving to supervisor directory

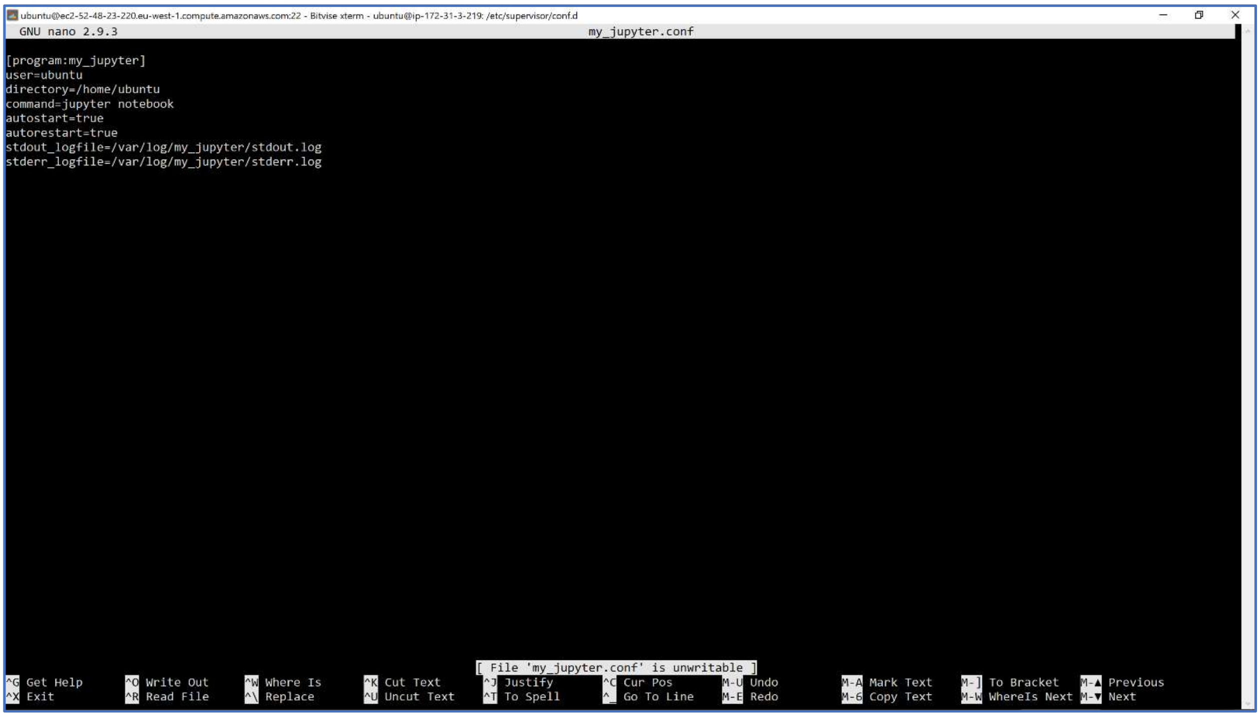
Use sudo nano to create my_jupyter.conf


ubuntu@ec2-52-48-23-220.eu-west-1.compute.amazonaws.com:22 - Bitvise xterm - ubuntu@ip-172-31-3-219: /etc/supervisor/conf.d

```
www-data  1018  0.0  0.0 143788  7052 ?        S    Aug01   0:00 nginx: worker process
www-data  1019  0.0  0.0 143788  6320 ?        S    Aug01   0:00 nginx: worker process
root      1199  0.0  0.0   4504   720 ?        S    Aug01   0:00 bpfilter_umh
ubuntu    1427  0.0  0.1 966068 85000 ?        Sl   Aug01   1:06 /usr/bin/python3 /usr/local/bin/jupyter-notebook
ubuntu    2428  0.8  9.1 6584476 5768592 ?     Ssl  Aug01  25:29 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-01
ubuntu    2476  0.0  0.0 542736 49216 ?        Ssl  Aug01   0:07 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-ca
ubuntu    2832  1.1  8.9 6381620 5605328 ?     Ssl  Aug01  36:12 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-68
ubuntu    3243  0.0  8.3 6101636 5235632 ?     Ssl  Aug01   0:41 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-09
root      4123  0.0  0.0      0     0 ?        I    Aug01   0:02 [kworker/1:1-eve]
root      4574  0.0  0.0      0     0 ?        I    Aug02   0:00 [kworker/3:0-cgr]
root      5063  0.0  0.0      0     0 ?        I    Aug02   0:03 [kworker/3:2-mm_]
root      8642  0.0  0.0      0     0 ?        I    00:00   0:00 [kworker/0:0-mm_]
root      9019  0.0  0.0      0     0 ?        I    03:30   0:00 [kworker/2:0-eve]
root      9804  0.0  0.0      0     0 ?        I    03:57   0:00 [kworker/2:1]
root     10472  0.0  0.0      0     0 ?        I    06:46   0:00 [kworker/0:1]
root     10990  0.0  0.0      0     0 ?        I    11:45   0:00 [kworker/1:2-eve]
root     11371  0.0  0.0      0     0 ?        I    15:04   0:00 [kworker/u30:1-e]
root     11375  0.0  0.0      0     0 ?        I    15:11   0:00 [kworker/u30:0-e]
root     11376  0.0  0.0 107988  7068 ?        Ss   15:12   0:00 sshd: ubuntu [priv]
ubuntu   11378  0.0  0.0  76556  7292 ?        Ss   15:12   0:00 /lib/systemd/systemd --user
ubuntu   11379  0.0  0.0 259348  2540 ?        S    15:12   0:00 (sd-pam)
ubuntu   11497  0.0  0.0 107988  4212 ?        R    15:12   0:00 sshd: ubuntu@pts/0
ubuntu   11498  0.0  0.0  19616  2592 ?        Ss   15:12   0:00 /usr/lib/openssh/sftp-server
ubuntu   11499  0.0  0.0  23544  5448 pts/0    Ss   15:12   0:00 -bash
ubuntu   11566  0.2  0.0 144072 45876 pts/0    Tl   15:23   0:00 /home/ubuntu/anaconda3/bin/python /home/ubuntu/anaconda3/bin/ipython -c from notebook.auth import passwd; pass
root     11573  0.0  0.0      0     0 ?        I    15:24   0:00 [kworker/u30:2-e]
ubuntu   11581  0.0  0.0  37796  3340 pts/0    R+   15:27   0:00 ps aux
(base) ubuntu@ip-172-31-3-219:~$ ps 11566
  PID TTY      STAT   TIME COMMAND
11566 pts/0    Tl     0:00 /home/ubuntu/anaconda3/bin/python /home/ubuntu/anaconda3/bin/ipython -c from notebook.auth import passwd; passwd()
(base) ubuntu@ip-172-31-3-219:~$ cd /etc/nginx/
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ cd sites-available/
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ ls
default  jupyter_app.conf
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ nano jupyter_app.conf
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ cd ..
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ ls
conf.d         fastcgi_params  koi-win     modules-available  nginx.conf    scgi_params      sites-enabled  uwsgi_params
fastcgi.conf  koi-utf         mime.types  modules-enabled    proxy_params  sites-available  snippets       win-utf
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ cd sites-enabled/
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-enabled$ cd /etc/supervisor/
(base) ubuntu@ip-172-31-3-219:/etc/supervisor$ ls
conf.d  supervisord.conf
(base) ubuntu@ip-172-31-3-219:/etc/supervisor$ cd conf.d/
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$ ls
my_jupyter.conf
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$
```

Figure 31: Use sudo nano to create my_jupyter.conf

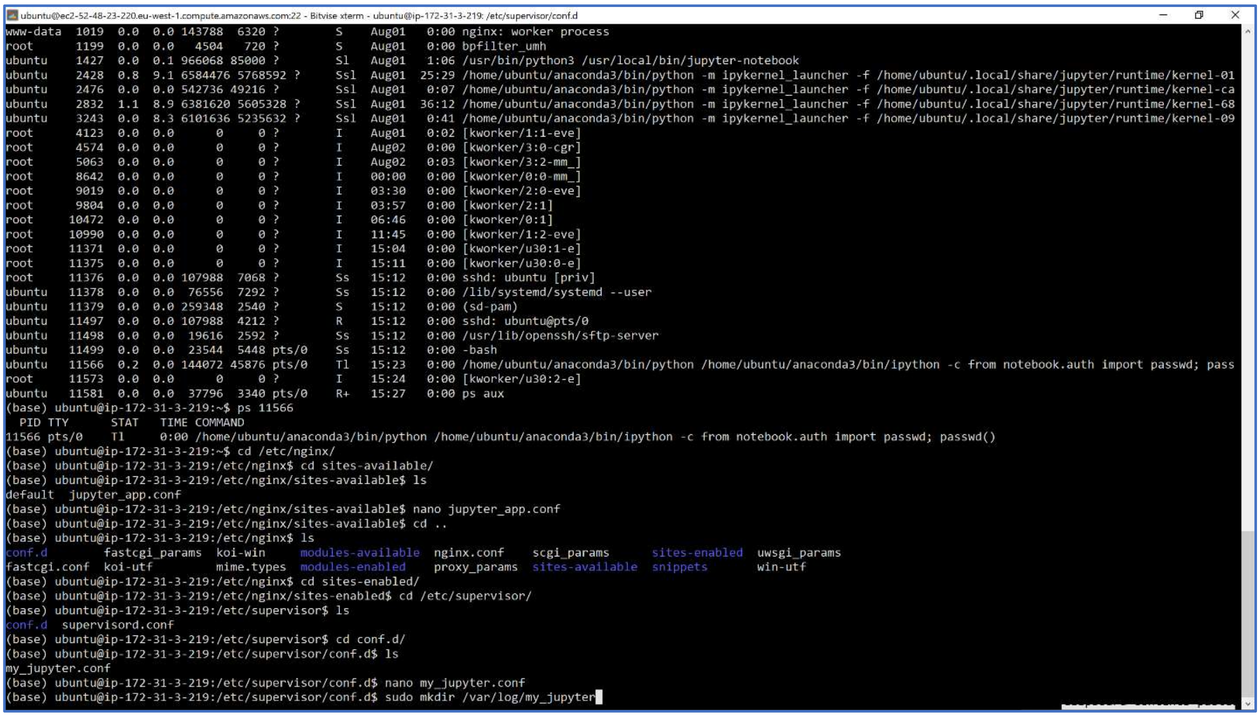
Specify supervisor to run the jupyter notebook command upon starting up



```
[program:my_jupyter]
user=ubuntu
directory=/home/ubuntu
command=jupyter notebook
autostart=true
autorestart=true
stdout_logfile=/var/log/my_jupyter/stdout.log
stderr_logfile=/var/log/my_jupyter/stderr.log
```

```
[ File 'my_jupyter.conf' is unwritable ]
```

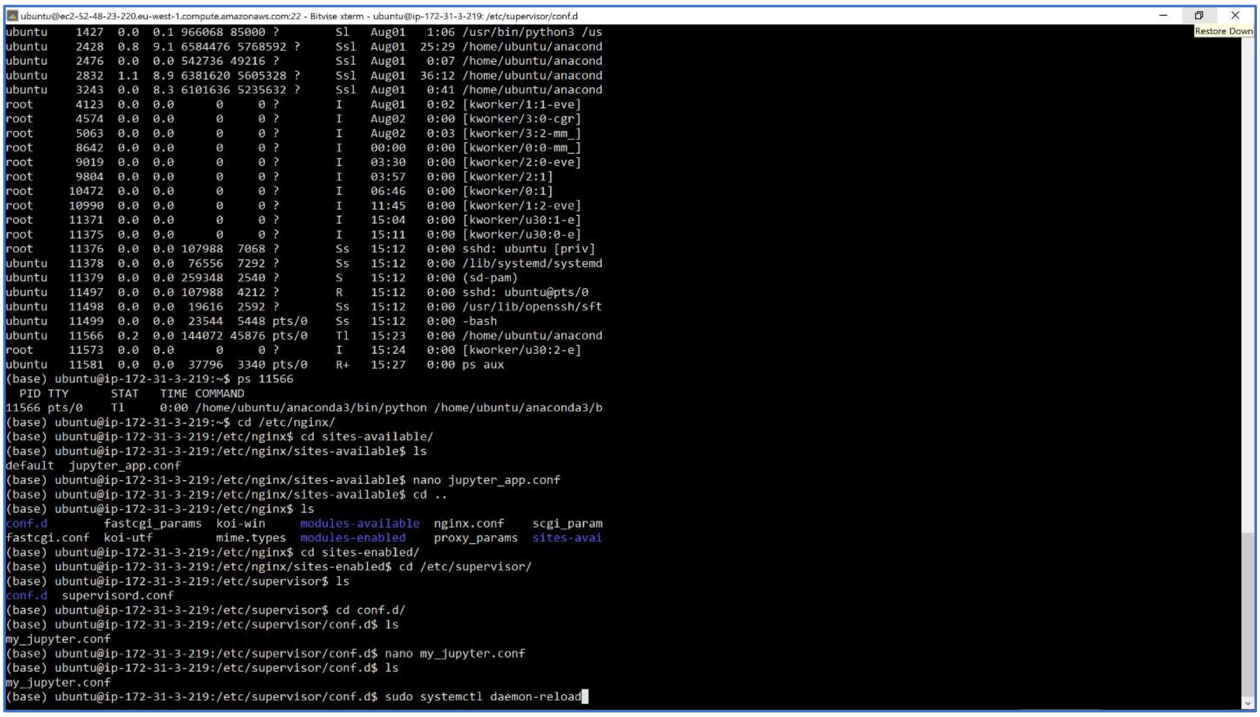
Figure 32: Configure Jupiter notebook to run when instance is started



```
ubuntu@ec2-52-48-23-220.eu-west-1.compute.amazonaws.com:22 - Bitvise xterm - ubuntu@ip-172-31-3-219: /etc/supervisor/conf.d
www-data  1019  0.0  0.0 143788  6320 ?        S    Aug01   0:00 nginx: worker process
root      1199  0.0  0.0   4504   720 ?        S    Aug01   0:00 bpfilter_umh
ubuntu    1427  0.0  0.1 966068 85000 ?        Sl   Aug01   1:06 /usr/bin/python3 /usr/local/bin/jupyter-notebook
ubuntu    2428  0.8  9.1 6584476 5768592 ?     Ssl  Aug01  25:29 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-01
ubuntu    2476  0.0  0.0 542736 49216 ?        Ssl  Aug01   0:07 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-ca
ubuntu    2832  1.1  8.9 6381620 5605328 ?     Ssl  Aug01  36:12 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-68
ubuntu    3243  0.0  8.3 6101636 5235632 ?     Ssl  Aug01   0:41 /home/ubuntu/anaconda3/bin/python -m ipykernel_launcher -f /home/ubuntu/.local/share/jupyter/runtime/kernel-09
root      4123  0.0  0.0      0     0 ?        I    Aug01   0:02 [kworker/1:1-eve]
root      4574  0.0  0.0      0     0 ?        I    Aug02   0:00 [kworker/3:0-cgr]
root      5063  0.0  0.0      0     0 ?        I    Aug02   0:03 [kworker/3:2-mm_]
root      8642  0.0  0.0      0     0 ?        I    00:00   0:00 [kworker/0:0-mm_]
root      9019  0.0  0.0      0     0 ?        I    03:30   0:00 [kworker/2:0-eve]
root      9804  0.0  0.0      0     0 ?        I    03:57   0:00 [kworker/2:1]
root     10472  0.0  0.0      0     0 ?        I    06:46   0:00 [kworker/0:1]
root     10990  0.0  0.0      0     0 ?        I    11:45   0:00 [kworker/1:2-eve]
root     11371  0.0  0.0      0     0 ?        I    15:04   0:00 [kworker/u30:1-e]
root     11375  0.0  0.0      0     0 ?        I    15:11   0:00 [kworker/u30:0-e]
root     11376  0.0  0.0 107988  7068 ?        Ss   15:12   0:00 sshd: ubuntu [priv]
ubuntu   11378  0.0  0.0  76556  7292 ?        Ss   15:12   0:00 /lib/systemd/systemd --user
ubuntu   11379  0.0  0.0 259348  2540 ?        S    15:12   0:00 (sd-pam)
ubuntu   11497  0.0  0.0 107988  4212 ?        R    15:12   0:00 sshd: ubuntu@pts/0
ubuntu   11498  0.0  0.0  19616  2592 ?        Ss   15:12   0:00 /usr/lib/openssh/sftp-server
ubuntu   11499  0.0  0.0  23544  5448 pts/0    Ss   15:12   0:00 -bash
ubuntu   11566  0.2  0.0 144072 45876 pts/0    Tl   15:23   0:00 /home/ubuntu/anaconda3/bin/python /home/ubuntu/anaconda3/bin/ipython -c from notebook.auth import passwd; pass
root     11573  0.0  0.0      0     0 ?        I    15:24   0:00 [kworker/u30:2-e]
ubuntu   11581  0.0  0.0  37796  3340 pts/0    R+   15:27   0:00 ps aux
(base) ubuntu@ip-172-31-3-219:~$ ps 11566
  PID TTY      STAT   TIME COMMAND
11566 pts/0    Tl     0:00 /home/ubuntu/anaconda3/bin/python /home/ubuntu/anaconda3/bin/ipython -c from notebook.auth import passwd; passwd()
(base) ubuntu@ip-172-31-3-219:~$ cd /etc/nginx/
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ cd sites-available/
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ ls
default  jupyter_app.conf
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ nano jupyter_app.conf
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ cd ..
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ ls
conf.d        fastcgi_params  koi-win     modules-available  nginx.conf    scgi_params      sites-enabled  uwsgi_params
fastcgi.conf  koi-utf         mime.types  modules-enabled    proxy_params  sites-available  snippets       win-utf
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ cd sites-enabled/
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-enabled$ cd /etc/supervisor/
(base) ubuntu@ip-172-31-3-219:/etc/supervisor$ ls
conf.d  supervisord.conf
(base) ubuntu@ip-172-31-3-219:/etc/supervisor$ cd conf.d/
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$ ls
my_jupyter.conf
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$ nano my_jupyter.conf
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$ sudo mkdir /var/log/my_jupyter
```

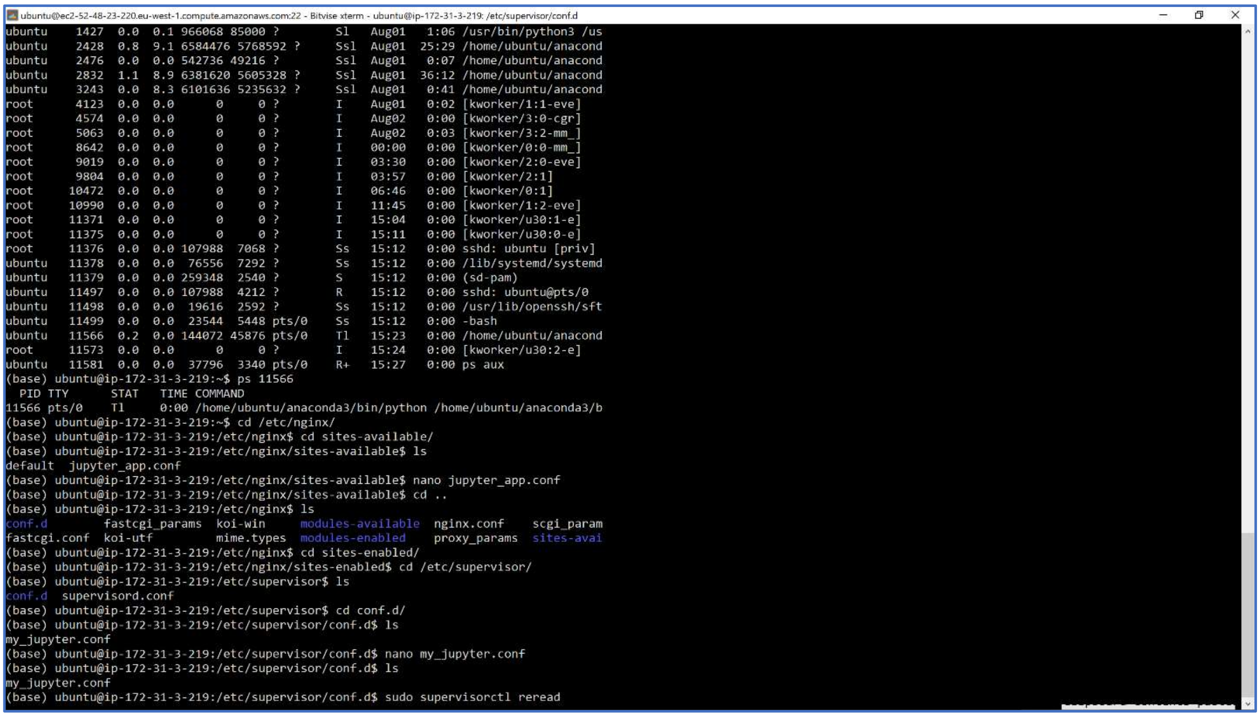
Figure 33: Create log file directory

Create log folder, for output as specified in my_jupyter.conf



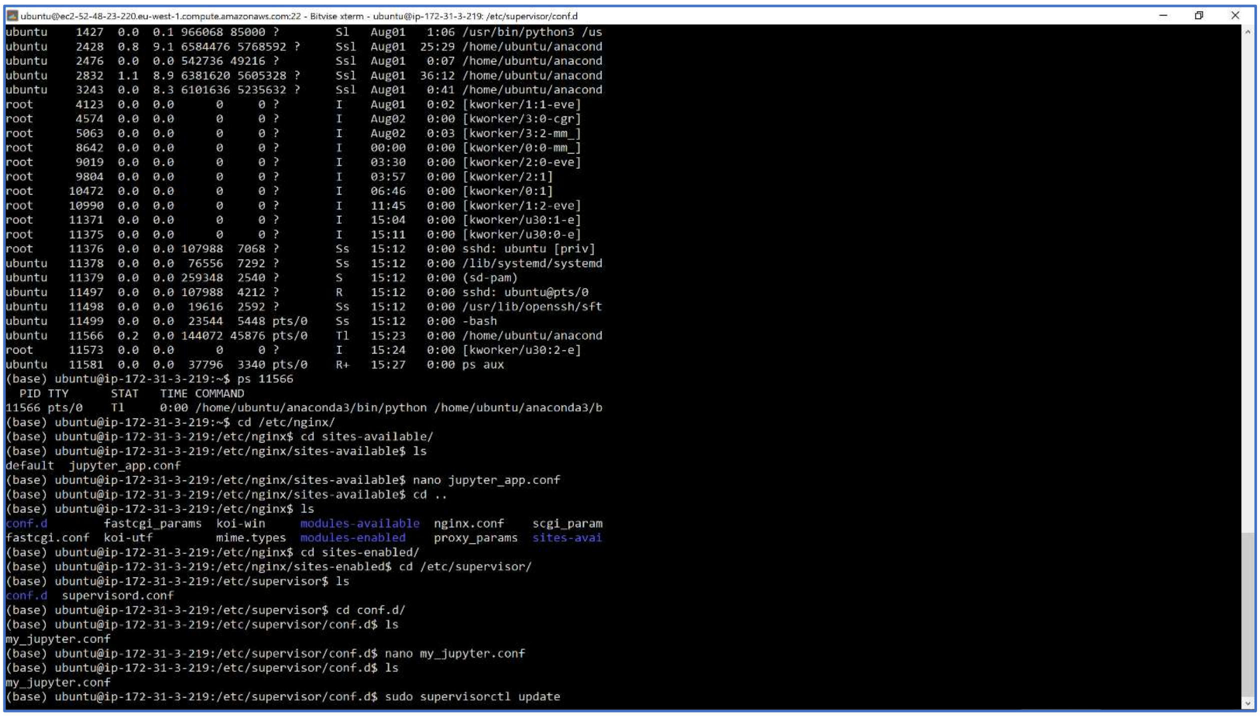
```
ubuntu@ec2-52-48-23-220.eu-west-1.compute.amazonaws.com:22 - Bitvise xterm - ubuntu@ip-172-31-3-219: /etc/supervisor/conf.d
ubuntu    1427  0.0  0.1 966068 85000 ?        Sl   Aug01   1:06 /usr/bin/python3 /us
ubuntu    2428  0.8  9.1 6584476 5768592 ?     Ssl  Aug01  25:29 /home/ubuntu/anacond
ubuntu    2476  0.0  0.0 542736 49216 ?        Ssl  Aug01   0:07 /home/ubuntu/anacond
ubuntu    2832  1.1  8.9 6381620 5605328 ?     Ssl  Aug01  36:12 /home/ubuntu/anacond
ubuntu    3243  0.0  8.3 6101636 5235632 ?     Ssl  Aug01   0:41 /home/ubuntu/anacond
root      4123  0.0  0.0      0     0 ?        I    Aug01   0:02 [kworker/1:1-eve]
root      4574  0.0  0.0      0     0 ?        I    Aug02   0:00 [kworker/3:0-cgr]
root      5063  0.0  0.0      0     0 ?        I    Aug02   0:03 [kworker/3:2-mm_]
root      8642  0.0  0.0      0     0 ?        I    00:00   0:00 [kworker/0:0-mm_]
root      9019  0.0  0.0      0     0 ?        I    03:30   0:00 [kworker/2:0-eve]
root      9804  0.0  0.0      0     0 ?        I    03:57   0:00 [kworker/2:1]
root     10472  0.0  0.0      0     0 ?        I    06:46   0:00 [kworker/0:1]
root     10990  0.0  0.0      0     0 ?        I    11:45   0:00 [kworker/1:2-eve]
root     11371  0.0  0.0      0     0 ?        I    15:04   0:00 [kworker/u30:1-e]
root     11375  0.0  0.0      0     0 ?        I    15:11   0:00 [kworker/u30:0-e]
root     11376  0.0  0.0 107988  7068 ?        Ss   15:12   0:00 sshd: ubuntu [priv]
ubuntu   11378  0.0  0.0  76556  7292 ?        Ss   15:12   0:00 /lib/systemd/systemd
ubuntu   11379  0.0  0.0 259348  2540 ?        S    15:12   0:00 (sd-pam)
ubuntu   11497  0.0  0.0 107988  4212 ?        R    15:12   0:00 sshd: ubuntu@pts/0
ubuntu   11498  0.0  0.0  19616  2592 ?        Ss   15:12   0:00 /usr/lib/openssh/sft
ubuntu   11499  0.0  0.0  23544  5448 pts/0    Ss   15:12   0:00 -bash
ubuntu   11566  0.2  0.0 144072 45876 pts/0    Tl   15:23   0:00 /home/ubuntu/anacond
root     11573  0.0  0.0      0     0 ?        I    15:24   0:00 [kworker/u30:2-e]
ubuntu   11581  0.0  0.0  37796  3340 pts/0    R+   15:27   0:00 ps aux
(base) ubuntu@ip-172-31-3-219:~$ ps 11566
  PID TTY      STAT   TIME COMMAND
11566 pts/0    Tl     0:00 /home/ubuntu/anaconda3/bin/python /home/ubuntu/anaconda3/b
(base) ubuntu@ip-172-31-3-219:~$ cd /etc/nginx/
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ cd sites-available/
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ ls
default  jupyter_app.conf
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ nano jupyter_app.conf
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ cd ..
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ ls
conf.d        fastcgi_params  koi-win     modules-available  nginx.conf    scgi_param
fastcgi.conf  koi-utf         mime.types  modules-enabled    proxy_params  sites-avai
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ cd sites-enabled/
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-enabled$ cd /etc/supervisor/
(base) ubuntu@ip-172-31-3-219:/etc/supervisor$ ls
conf.d  supervisord.conf
(base) ubuntu@ip-172-31-3-219:/etc/supervisor$ cd conf.d/
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$ ls
my_jupyter.conf
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$ nano my_jupyter.conf
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$ ls
my_jupyter.conf
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$ sudo systemctl daemon-reload
```

Figure 34: Refresh systemctl daemon that uses supervisor daemon process to re-initialise supervisor


ubuntu@ec2-52-48-23-220.eu-west-1.compute.amazonaws.com:22 - Bitvise xterm - ubuntu@ip-172-31-3-219: /etc/supervisor/conf.d

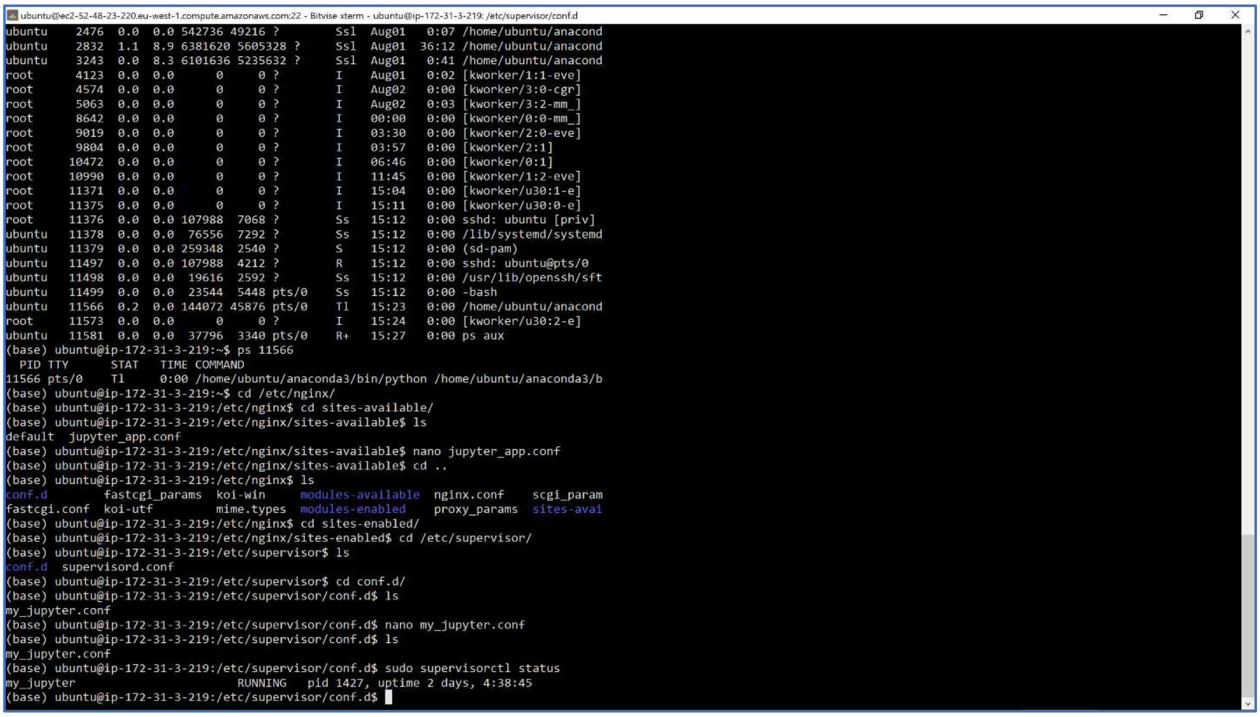
```
ubuntu    1427  0.0  0.1 966068 85000 ?        Sl   Aug01   1:06 /usr/bin/python3 /us
ubuntu    2428  0.8  9.1 6584476 5768592 ?     Ssl  Aug01  25:29 /home/ubuntu/anacond
ubuntu    2476  0.0  0.0 542736 49216 ?        Ssl  Aug01   0:07 /home/ubuntu/anacond
ubuntu    2832  1.1  8.9 6381620 5605328 ?     Ssl  Aug01  36:12 /home/ubuntu/anacond
ubuntu    3243  0.0  8.3 6101636 5235632 ?     Ssl  Aug01   0:41 /home/ubuntu/anacond
root      4123  0.0  0.0      0     0 ?        I    Aug01   0:02 [kworker/1:1-eve]
root      4574  0.0  0.0      0     0 ?        I    Aug02   0:00 [kworker/3:0-cgr]
root      5063  0.0  0.0      0     0 ?        I    Aug02   0:03 [kworker/3:2-mm_]
root      8642  0.0  0.0      0     0 ?        I    00:00   0:00 [kworker/0:0-mm_]
root      9019  0.0  0.0      0     0 ?        I    03:30   0:00 [kworker/2:0-eve]
root      9804  0.0  0.0      0     0 ?        I    03:57   0:00 [kworker/2:1]
root     10472  0.0  0.0      0     0 ?        I    06:46   0:00 [kworker/0:1]
root     10990  0.0  0.0      0     0 ?        I    11:45   0:00 [kworker/1:2-eve]
root     11371  0.0  0.0      0     0 ?        I    15:04   0:00 [kworker/u30:1-e]
root     11375  0.0  0.0      0     0 ?        I    15:11   0:00 [kworker/u30:0-e]
root     11376  0.0  0.0 107988  7068 ?        Ss   15:12   0:00 sshd: ubuntu [priv]
ubuntu   11378  0.0  0.0  76556  7292 ?        Ss   15:12   0:00 /lib/systemd/systemd
ubuntu   11379  0.0  0.0 259348  2540 ?        S    15:12   0:00 (sd-pam)
ubuntu   11497  0.0  0.0 107988  4212 ?        R    15:12   0:00 sshd: ubuntu@pts/0
ubuntu   11498  0.0  0.0  19616  2592 ?        Ss   15:12   0:00 /usr/lib/openssh/sft
ubuntu   11499  0.0  0.0  23544  5448 pts/0    Ss   15:12   0:00 -bash
ubuntu   11566  0.2  0.0 144072 45876 pts/0    Tl   15:23   0:00 /home/ubuntu/anacond
root     11573  0.0  0.0      0     0 ?        I    15:24   0:00 [kworker/u30:2-e]
ubuntu   11581  0.0  0.0  37796  3340 pts/0    R+   15:27   0:00 ps aux
(base) ubuntu@ip-172-31-3-219:~$ ps 11566
  PID TTY      STAT   TIME COMMAND
11566 pts/0    Tl     0:00 /home/ubuntu/anaconda3/bin/python /home/ubuntu/anaconda3/b
(base) ubuntu@ip-172-31-3-219:~$ cd /etc/nginx/
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ cd sites-available/
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ ls
default  jupyter_app.conf
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ nano jupyter_app.conf
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ cd ..
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ ls
conf.d        fastcgi_params  koi-win     modules-available  nginx.conf    scgi_param
fastcgi.conf  koi-utf         mime.types  modules-enabled    proxy_params  sites-avai
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ cd sites-enabled/
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-enabled$ cd /etc/supervisor/
(base) ubuntu@ip-172-31-3-219:/etc/supervisor$ ls
conf.d  supervisord.conf
(base) ubuntu@ip-172-31-3-219:/etc/supervisor$ cd conf.d/
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$ ls
my_jupyter.conf
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$ nano my_jupyter.conf
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$ ls
my_jupyter.conf
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$ sudo supervisorctl reread
```

Figure 35: Re-read configuration file for supervisor daemon


ubuntu@ec2-52-48-23-220.eu-west-1.compute.amazonaws.com:22 - Bitvise xterm - ubuntu@ip-172-31-3-219: /etc/supervisor/conf.d

```
ubuntu    1427  0.0  0.1 966068 85000 ?        Sl   Aug01   1:06 /usr/bin/python3 /us
ubuntu    2428  0.8  9.1 6584476 5768592 ?     Ssl  Aug01  25:29 /home/ubuntu/anacond
ubuntu    2476  0.0  0.0 542736 49216 ?        Ssl  Aug01   0:07 /home/ubuntu/anacond
ubuntu    2832  1.1  8.9 6381620 5605328 ?     Ssl  Aug01  36:12 /home/ubuntu/anacond
ubuntu    3243  0.0  8.3 6101636 5235632 ?     Ssl  Aug01   0:41 /home/ubuntu/anacond
root      4123  0.0  0.0      0     0 ?        I    Aug01   0:02 [kworker/1:1-eve]
root      4574  0.0  0.0      0     0 ?        I    Aug02   0:00 [kworker/3:0-cgr]
root      5063  0.0  0.0      0     0 ?        I    Aug02   0:03 [kworker/3:2-mm_]
root      8642  0.0  0.0      0     0 ?        I    00:00   0:00 [kworker/0:0-mm_]
root      9019  0.0  0.0      0     0 ?        I    03:30   0:00 [kworker/2:0-eve]
root      9804  0.0  0.0      0     0 ?        I    03:57   0:00 [kworker/2:1]
root     10472  0.0  0.0      0     0 ?        I    06:46   0:00 [kworker/0:1]
root     10990  0.0  0.0      0     0 ?        I    11:45   0:00 [kworker/1:2-eve]
root     11371  0.0  0.0      0     0 ?        I    15:04   0:00 [kworker/u30:1-e]
root     11375  0.0  0.0      0     0 ?        I    15:11   0:00 [kworker/u30:0-e]
root     11376  0.0  0.0 107988  7068 ?        Ss   15:12   0:00 sshd: ubuntu [priv]
ubuntu   11378  0.0  0.0  76556  7292 ?        Ss   15:12   0:00 /lib/systemd/systemd
ubuntu   11379  0.0  0.0 259348  2540 ?        S    15:12   0:00 (sd-pam)
ubuntu   11497  0.0  0.0 107988  4212 ?        R    15:12   0:00 sshd: ubuntu@pts/0
ubuntu   11498  0.0  0.0  19616  2592 ?        Ss   15:12   0:00 /usr/lib/openssh/sft
ubuntu   11499  0.0  0.0  23544  5448 pts/0    Ss   15:12   0:00 -bash
ubuntu   11566  0.2  0.0 144072 45876 pts/0    Tl   15:23   0:00 /home/ubuntu/anacond
root     11573  0.0  0.0      0     0 ?        I    15:24   0:00 [kworker/u30:2-e]
ubuntu   11581  0.0  0.0  37796  3340 pts/0    R+   15:27   0:00 ps aux
(base) ubuntu@ip-172-31-3-219:~$ ps 11566
  PID TTY      STAT   TIME COMMAND
11566 pts/0    Tl     0:00 /home/ubuntu/anaconda3/bin/python /home/ubuntu/anaconda3/b
(base) ubuntu@ip-172-31-3-219:~$ cd /etc/nginx/
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ cd sites-available/
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ ls
default  jupyter_app.conf
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ nano jupyter_app.conf
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ cd ..
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ ls
conf.d        fastcgi_params  koi-win     modules-available  nginx.conf    scgi_param
fastcgi.conf  koi-utf         mime.types  modules-enabled    proxy_params  sites-avai
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ cd sites-enabled/
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-enabled$ cd /etc/supervisor/
(base) ubuntu@ip-172-31-3-219:/etc/supervisor$ ls
conf.d  supervisord.conf
(base) ubuntu@ip-172-31-3-219:/etc/supervisor$ cd conf.d/
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$ ls
my_jupyter.conf
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$ nano my_jupyter.conf
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$ ls
my_jupyter.conf
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$ sudo supervisorctl update
```

Figure 36: Refresh supervisor daemon

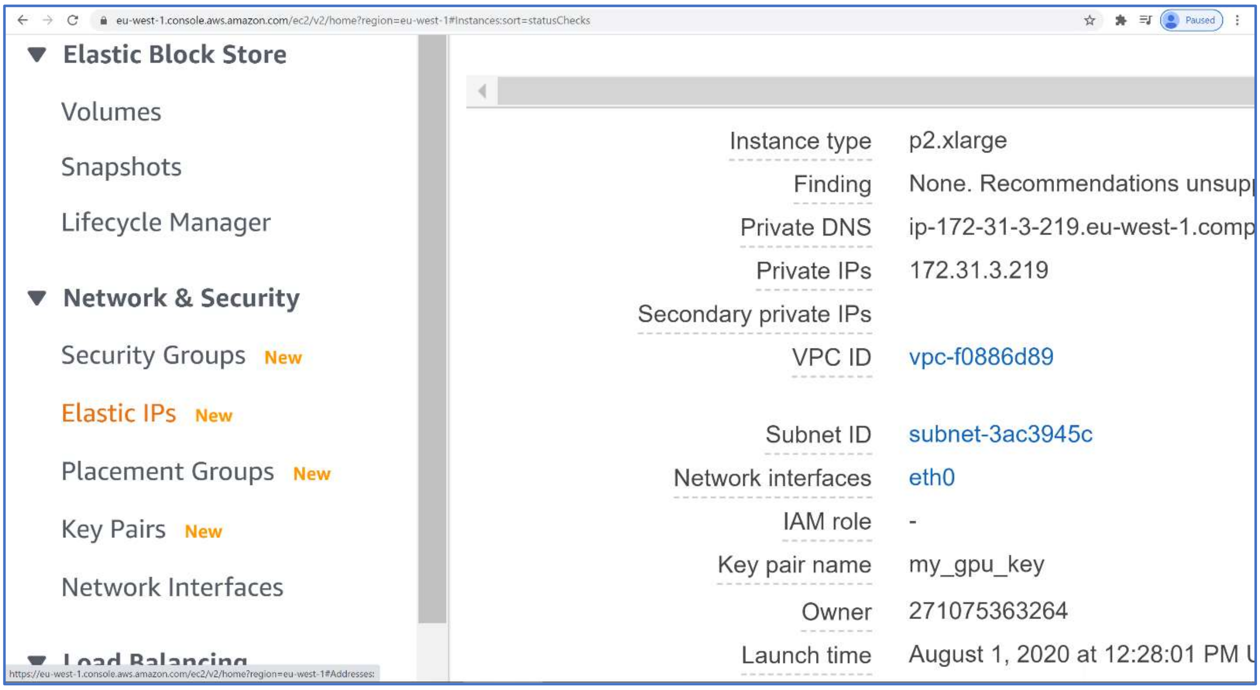


ubuntu@ec2-52-48-23-220.eu-west-1.compute.amazonaws.com:22 - Bitvise xterm - ubuntu@ip-172-31-3-219: /etc/supervisor/conf.d

```
ubuntu    2476  0.0  0.0 542736 49216 ?       Ssl  Aug01   0:07 /home/ubuntu/anacond
ubuntu    2832  1.1  8.9 6381620 5605328 ?    Ssl  Aug01  36:12 /home/ubuntu/anacond
ubuntu    3243  0.0  8.3 6101636 5235632 ?    Ssl  Aug01   0:41 /home/ubuntu/anacond
root      4123  0.0  0.0      0     0 ?       I    Aug01   0:02 [kworker/1:1-eve]
root      4574  0.0  0.0      0     0 ?       I    Aug02   0:00 [kworker/3:0-cgr]
root      5063  0.0  0.0      0     0 ?       I    Aug02   0:03 [kworker/3:2-mm_]
root      8642  0.0  0.0      0     0 ?       I    00:00   0:00 [kworker/0:0-mm_]
root      9019  0.0  0.0      0     0 ?       I    03:30   0:00 [kworker/2:0-eve]
root      9804  0.0  0.0      0     0 ?       I    03:57   0:00 [kworker/2:1]
root     10472  0.0  0.0      0     0 ?       I    06:46   0:00 [kworker/0:1]
root     10990  0.0  0.0      0     0 ?       I    11:45   0:00 [kworker/1:2-eve]
root     11371  0.0  0.0      0     0 ?       I    15:04   0:00 [kworker/u30:1-e]
root     11375  0.0  0.0      0     0 ?       I    15:11   0:00 [kworker/u30:0-e]
root     11376  0.0  0.0 107988  7068 ?       Ss   15:12   0:00 sshd: ubuntu [priv]
ubuntu   11378  0.0  0.0  76556  7292 ?       Ss   15:12   0:00 /lib/systemd/systemd
ubuntu   11379  0.0  0.0 259348  2540 ?       S    15:12   0:00 (sd-pam)
ubuntu   11497  0.0  0.0 107988  4212 ?       R    15:12   0:00 sshd: ubuntu@pts/0
ubuntu   11498  0.0  0.0  19616  2592 ?       Ss   15:12   0:00 /usr/lib/openssh/sft
ubuntu   11499  0.0  0.0  23544  5448 pts/0   Ss   15:12   0:00 -bash
ubuntu   11566  0.2  0.0 144072 45876 pts/0   Tl   15:23   0:00 /home/ubuntu/anacond
root     11573  0.0  0.0      0     0 ?       I    15:24   0:00 [kworker/u30:2-e]
ubuntu   11581  0.0  0.0  37796  3340 pts/0   R+   15:27   0:00 ps aux
(base) ubuntu@ip-172-31-3-219:~$ ps 11566
  PID TTY      STAT   TIME COMMAND
11566 pts/0    Tl     0:00 /home/ubuntu/anaconda3/bin/python /home/ubuntu/anaconda3/b
(base) ubuntu@ip-172-31-3-219:~$ cd /etc/nginx/
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ cd sites-available/
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ ls
default  jupyter_app.conf
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ nano jupyter_app.conf
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-available$ cd ..
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ ls
conf.d         fastcgi_params  koi-win     modules-available  nginx.conf    scgi_param
fastcgi.conf  koi-utf         mime.types  modules-enabled    proxy_params  sites-avai
(base) ubuntu@ip-172-31-3-219:/etc/nginx$ cd sites-enabled/
(base) ubuntu@ip-172-31-3-219:/etc/nginx/sites-enabled$ cd /etc/supervisor/
(base) ubuntu@ip-172-31-3-219:/etc/supervisor$ ls
conf.d  supervisord.conf
(base) ubuntu@ip-172-31-3-219:/etc/supervisor$ cd conf.d/
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$ ls
my_jupyter.conf
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$ nano my_jupyter.conf
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$ ls
my_jupyter.conf
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$ sudo supervisorctl status
my_jupyter                       RUNNING   pid 1427, uptime 2 days, 4:38:45
(base) ubuntu@ip-172-31-3-219:/etc/supervisor/conf.d$
```

Figure 37: Running supervisor daemon

The notebook will now run upon starting the instance



Instance type p2.xlarge

Finding None. Recommendations unsup

Private DNS ip-172-31-3-219.eu-west-1.comp

Private IPs 172.31.3.219

Secondary private IPs

VPC ID vpc-f0886d89

Subnet ID subnet-3ac3945c

Network interfaces eth0

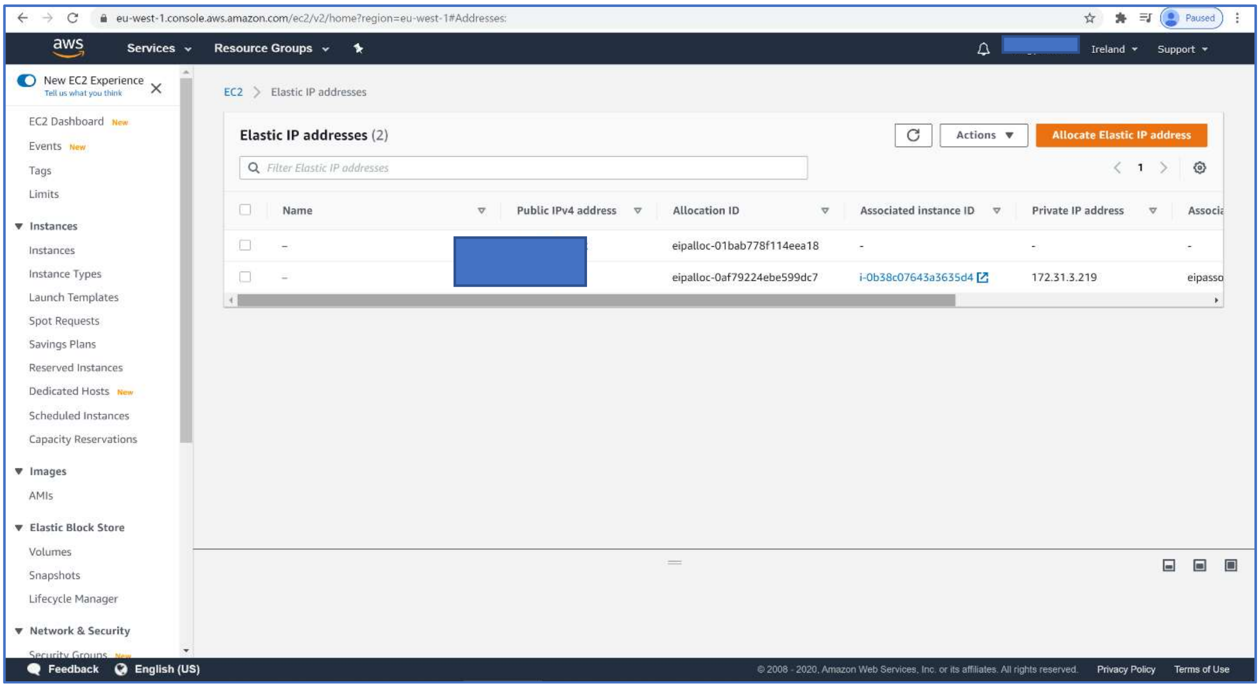
IAM role -

Key pair name my_gpu_key

Owner 271075363264

Launch time August 1, 2020 at 12:28:01 PM U

Figure 38: Selection of Elastic IP addresses



eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1#Addresses:

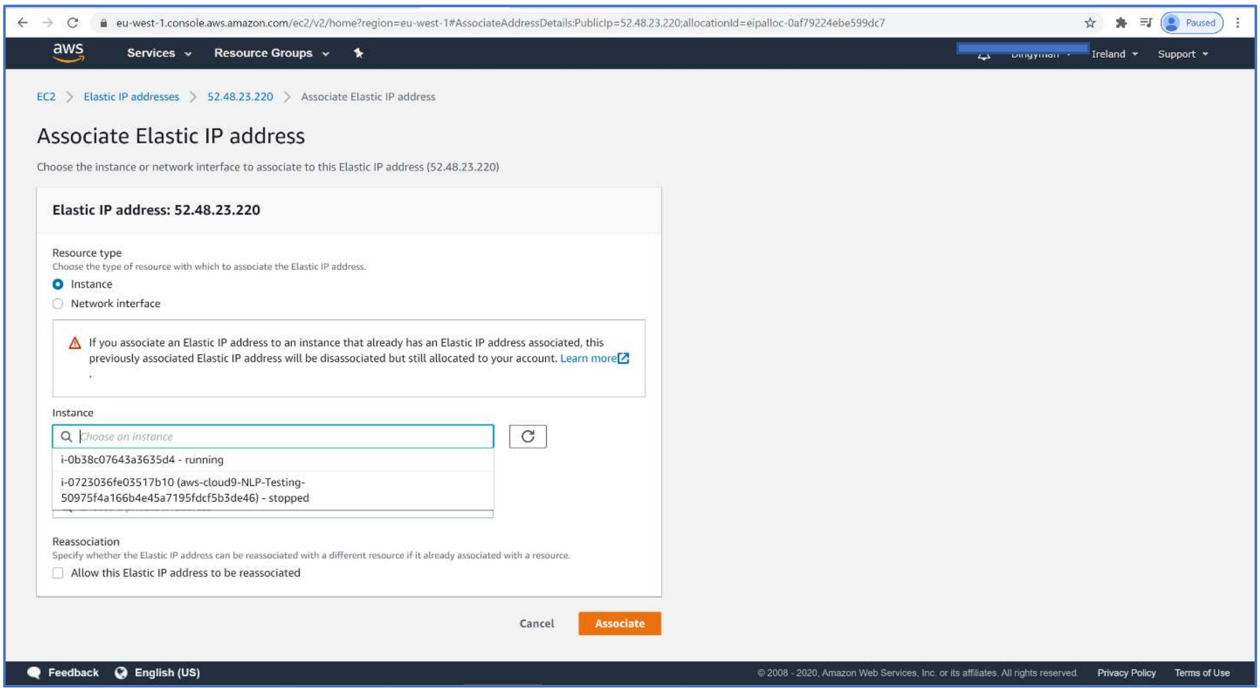
EC2 > Elastic IP addresses

Elastic IP addresses (2)

☐	Name	Public IPv4 address	Allocation ID	Associated instance ID	Private IP address	Associa
☐	–	[redacted]	eipalloc-01bab778f114eea18	-	-	-
☐	–	[redacted]	eipalloc-0af79224ebe599dc7	i-0b38c07643a3635d4	172.31.3.219	eipasso

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

Figure 39: List of generated Elastic IP addresses



eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1#AssociateAddressDetails:PublicIp=52.48.23.220;allocationId=eipalloc-0af79224ebe599dc7

EC2 > Elastic IP addresses > 52.48.23.220 > Associate Elastic IP address

Associate Elastic IP address

Choose the instance or network interface to associate to this Elastic IP address (52.48.23.220)

Elastic IP address: 52.48.23.220

Resource type
Choose the type of resource with which to associate the Elastic IP address.
● Instance
○ Network interface

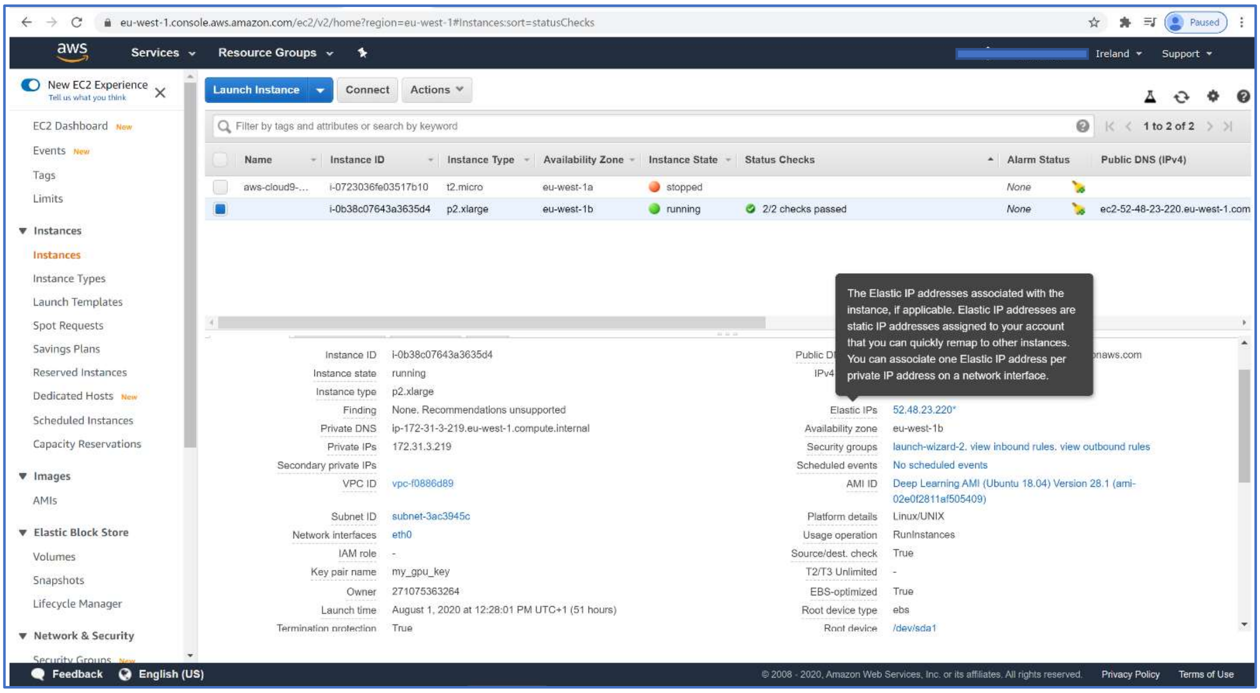
⚠ If you associate an Elastic IP address to an instance that already has an Elastic IP address associated, this previously associated Elastic IP address will be disassociated but still allocated to your account. Learn more

Instance
Choose an instance
i-0b38c07643a3635d4 - running
i-0723036fe03517b10 (aws-cloud9-NLP-Testing-50975f4a166b4e45a7195fdcf5b3de46) - stopped

Reassociation
Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.
☐ Allow this Elastic IP address to be reassociated

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

Figure 40: IP address association to current instance

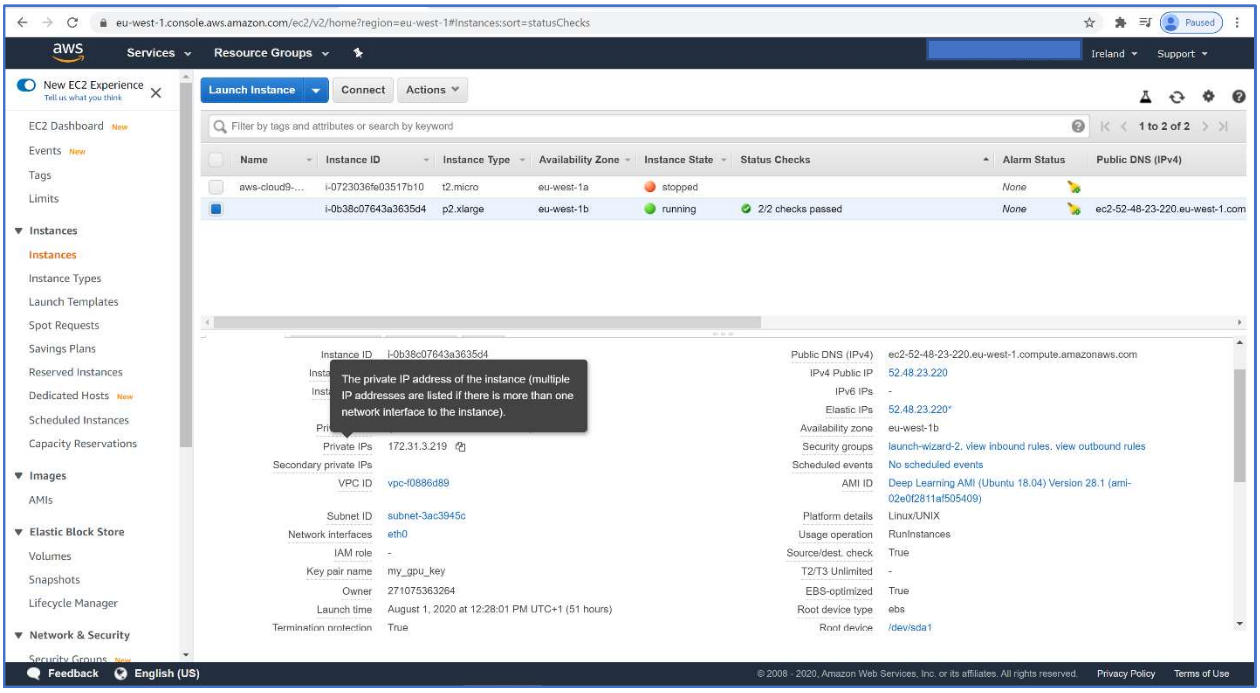


Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
aws-cloud9-...	i-0723036fe03517b10	t2.micro	eu-west-1a	stopped		None	
	i-0b38c07643a3635d4	p2.xlarge	eu-west-1b	running	2/2 checks passed	None	ec2-52-48-23-220.eu-west-1.com

The Elastic IP addresses associated with the instance, if applicable. Elastic IP addresses are static IP addresses assigned to your account that you can quickly remap to other instances. You can associate one Elastic IP address per private IP address on a network interface.

Instance ID	i-0b38c07643a3635d4	Public D	naws.com
Instance state	running	IPv4	
Instance type	p2.xlarge		
Finding	None. Recommendations unsupported	Elastic IPs	52.48.23.220*
Private DNS	ip-172-31-3-219.eu-west-1.compute.internal	Availability zone	eu-west-1b
Private IPs	172.31.3.219	Security groups	launch-wizard-2. view inbound rules. view outbound rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-f0886d89	AMI ID	Deep Learning AMI (Ubuntu 18.04) Version 28.1 (ami-02e0f2811af505409)
Subnet ID	subnet-3ac3945c	Platform details	Linux/UNIX
Network interfaces	eth0	Usage operation	RunInstances
IAM role	-	Source/dest. check	True
Key pair name	my_gpu_key	T2/T3 Unlimited	-
Owner	271075363264	EBS-optimized	True
Launch time	August 1, 2020 at 12:28:01 PM UTC+1 (51 hours)	Root device type	ebs
Termination protection	True	Root device	/dev/sda1

Figure 41: AWS Launch Instance



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
aws-cloud9-...	i-0723036fe03517b10	t2.micro	eu-west-1a	stopped		None	
	i-0b38c07643a3635d4	p2.xlarge	eu-west-1b	running	2/2 checks passed	None	ec2-52-48-23-220.eu-west-1.com

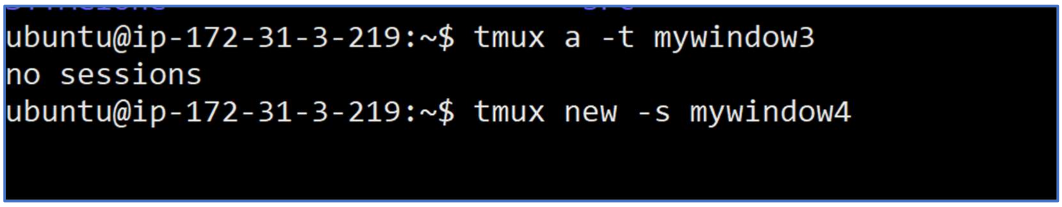
The private IP address of the instance (multiple IP addresses are listed if there is more than one network interface to the instance).

Instance ID	i-0b38c07643a3635d4	Public DNS (IPv4)	ec2-52-48-23-220.eu-west-1.compute.amazonaws.com
Insta		IPv4 Public IP	52.48.23.220
Inst		IPv6 IPs	-
		Elastic IPs	52.48.23.220*
Pri		Availability zone	eu-west-1b
Private IPs	172.31.3.219	Security groups	launch-wizard-2. view inbound rules. view outbound rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-f0886d89	AMI ID	Deep Learning AMI (Ubuntu 18.04) Version 28.1 (ami-02e0f2811af505409)
Subnet ID	subnet-3ac3945c	Platform details	Linux/UNIX
Network interfaces	eth0	Usage operation	RunInstances
IAM role	-	Source/dest. check	True
Key pair name	my_gpu_key	T2/T3 Unlimited	-
Owner	271075363264	EBS-optimized	True
Launch time	August 1, 2020 at 12:28:01 PM UTC+1 (51 hours)	Root device type	ebs
Termination protection	True	Root device	/dev/sda1

Figure 42: Private IP address

2 Data Collection and Processing

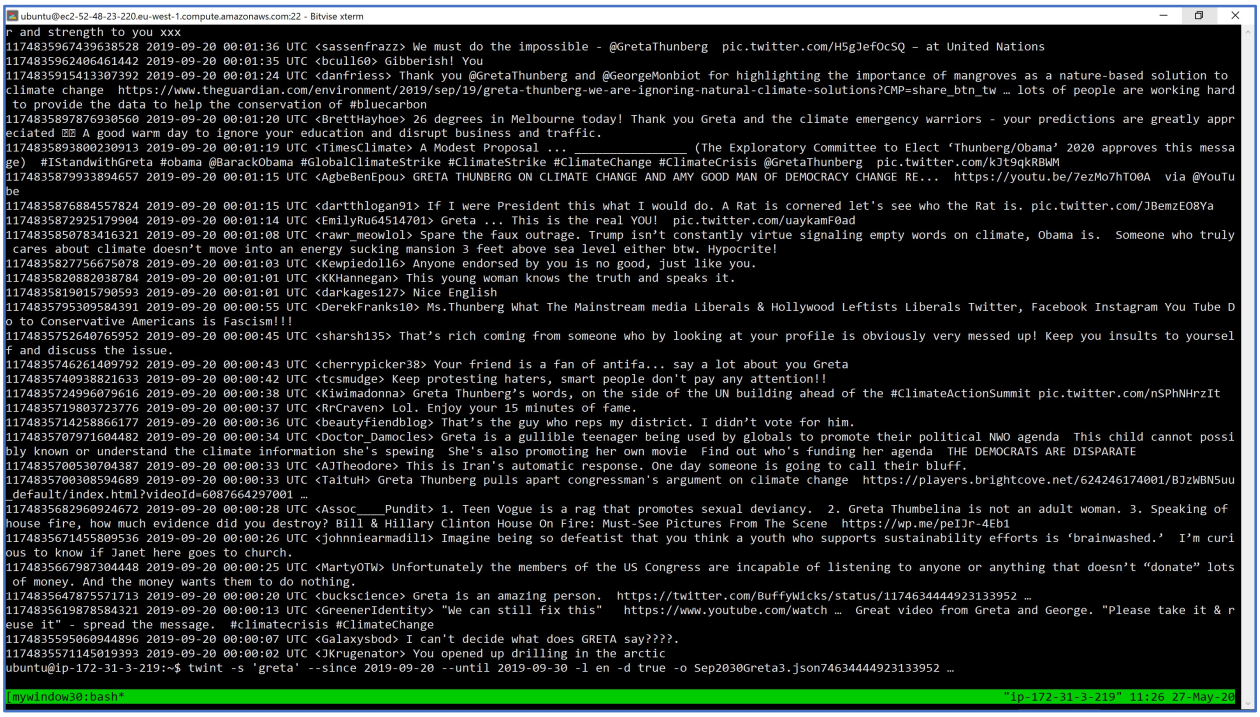
twint -s 'greta' --since 2019-09-20 --until 2019-09-30 -l en -d true -o Sep2030Greta3.json



```
ubuntu@ip-172-31-3-219:~$ tmux a -t mywindow3
no sessions
ubuntu@ip-172-31-3-219:~$ tmux new -s mywindow4
```

Figure 43: Tmux for creating processes that can continue running uninterrupted and be returned to later on

Twint was collected with the following command

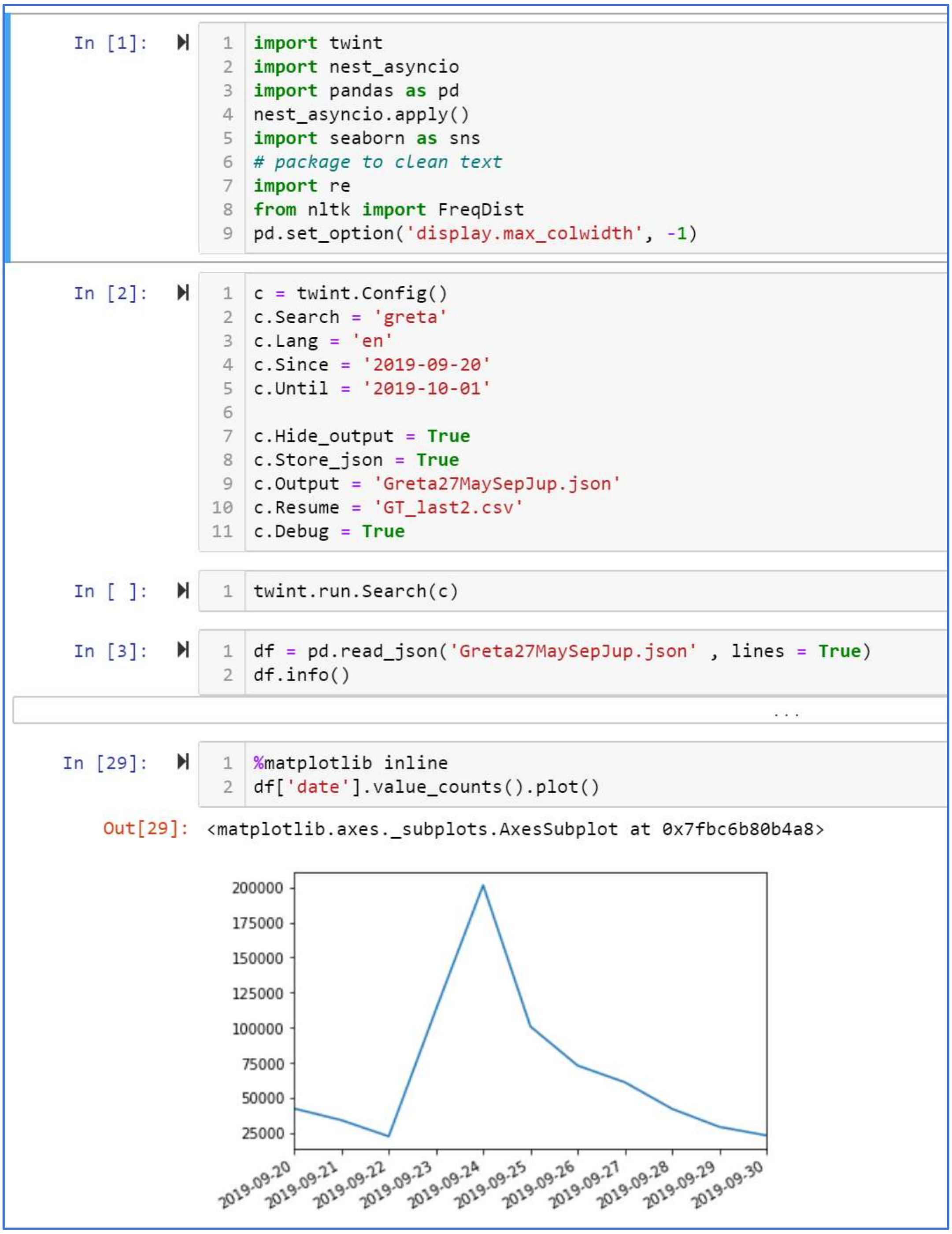


ubuntu@ec2-52-48-23-220.eu-west-1.compute.amazonaws.com:22 - Bitvise xterm

```
r and strength to you xxx
1174835967439638528 2019-09-20 00:01:36 UTC <sassenfrazz> We must do the impossible - @GretaThunberg  pic.twitter.com/H5gJefOcSQ – at United Nations
1174835962406461442 2019-09-20 00:01:35 UTC <bcull60> Gibberish! You
1174835915413307392 2019-09-20 00:01:24 UTC <danfriess> Thank you @GretaThunberg and @GeorgeMonbiot for highlighting the importance of mangroves as a nature-based solution to
climate change  https://www.theguardian.com/environment/2019/sep/19/greta-thunberg-we-are-ignoring-natural-climate-solutions?CMP=share_btn_tw … lots of people are working hard
 to provide the data to help the conservation of #bluecarbon
1174835897876930560 2019-09-20 00:01:20 UTC <BrettHayhoe> 26 degrees in Melbourne today! Thank you Greta and the climate emergency warriors - your predictions are greatly appr
eciated □□ A good warm day to ignore your education and disrupt business and traffic.
1174835893800230913 2019-09-20 00:01:19 UTC <TimesClimate> A Modest Proposal ... ______________ (The Exploratory Committee to Elect ‘Thunberg/Obama’ 2020 approves this messa
ge)  #IStandwithGreta #obama @BarackObama #GlobalClimateStrike #ClimateStrike #ClimateChange #ClimateCrisis @GretaThunberg  pic.twitter.com/kJt9qkRBWM
1174835879933894657 2019-09-20 00:01:15 UTC <AgbeBenEpou> GRETA THUNBERG ON CLIMATE CHANGE AND AMY GOOD MAN OF DEMOCRACY CHANGE RE...  https://youtu.be/7ezMo7hTO0A  via @YouTu
be
1174835876884557824 2019-09-20 00:01:15 UTC <dartthlogan91> If I were President this what I would do. A Rat is cornered let's see who the Rat is. pic.twitter.com/JBemzEO8Ya
1174835872925179904 2019-09-20 00:01:14 UTC <EmilyRu64514701> Greta ... This is the real YOU!  pic.twitter.com/uaykamF0ad
1174835850783416321 2019-09-20 00:01:08 UTC <rawr_meowlol> Spare the faux outrage. Trump isn’t constantly virtue signaling empty words on climate, Obama is.  Someone who truly
 cares about climate doesn’t move into an energy sucking mansion 3 feet above sea level either btw. Hypocrite!
1174835827756675078 2019-09-20 00:01:03 UTC <Kewpiedoll6> Anyone endorsed by you is no good, just like you.
1174835820882038784 2019-09-20 00:01:01 UTC <KKHannegan> This young woman knows the truth and speaks it.
1174835819015790593 2019-09-20 00:01:01 UTC <darkages127> Nice English
1174835795309584391 2019-09-20 00:00:55 UTC <DerekFranks10> Ms.Thunberg What The Mainstream media Liberals & Hollywood Leftists Liberals Twitter, Facebook Instagram You Tube D
o to Conservative Americans is Fascism!!!
1174835752640765952 2019-09-20 00:00:45 UTC <sharsh135> That’s rich coming from someone who by looking at your profile is obviously very messed up! Keep you insults to yoursel
f and discuss the issue.
1174835746261409792 2019-09-20 00:00:43 UTC <cherrypicker38> Your friend is a fan of antifa... say a lot about you Greta
1174835740938821633 2019-09-20 00:00:42 UTC <tcsmudge> Keep protesting haters, smart people don't pay any attention!!
1174835724996079616 2019-09-20 00:00:38 UTC <Kiwimadonna> Greta Thunberg’s words, on the side of the UN building ahead of the #ClimateActionSummit pic.twitter.com/nSPhNHrzIt
1174835719803723776 2019-09-20 00:00:37 UTC <RrCraven> Lol. Enjoy your 15 minutes of fame.
1174835714258866177 2019-09-20 00:00:36 UTC <beautyfiendblog> That’s the guy who reps my district. I didn’t vote for him.
1174835707971604482 2019-09-20 00:00:34 UTC <Doctor_Damocles> Greta is a gullible teenager being used by globals to promote their political NWO agenda  This child cannot possi
bly known or understand the climate information she's spewing  She's also promoting her own movie  Find out who's funding her agenda  THE DEMOCRATS ARE DISPARATE
1174835700530704387 2019-09-20 00:00:33 UTC <AJTheodore> This is Iran's automatic response. One day someone is going to call their bluff.
1174835700308594689 2019-09-20 00:00:33 UTC <TaituH> Greta Thunberg pulls apart congressman's argument on climate change  https://players.brightcove.net/624246174001/BJzWBN5uu
_default/index.html?videoId=6087664297001 …
1174835682960924672 2019-09-20 00:00:28 UTC <Assoc____Pundit> 1. Teen Vogue is a rag that promotes sexual deviancy.  2. Greta Thumbelina is not an adult woman. 3. Speaking of
house fire, how much evidence did you destroy? Bill & Hillary Clinton House On Fire: Must-See Pictures From The Scene  https://wp.me/peIJr-4Eb1
1174835671455809536 2019-09-20 00:00:26 UTC <johnniearmadil1> Imagine being so defeatist that you think a youth who supports sustainability efforts is ‘brainwashed.’  I’m curi
ous to know if Janet here goes to church.
1174835667987304448 2019-09-20 00:00:25 UTC <MartyOTW> Unfortunately the members of the US Congress are incapable of listening to anyone or anything that doesn’t “donate” lots
 of money. And the money wants them to do nothing.
1174835647875571713 2019-09-20 00:00:20 UTC <buckscience> Greta is an amazing person.  https://twitter.com/BuffyWicks/status/1174634444923133952 …
1174835619878584321 2019-09-20 00:00:13 UTC <GreenerIdentity> "We can still fix this"   https://www.youtube.com/watch …  Great video from Greta and George. "Please take it & r
euse it" - spread the message.  #climatecrisis #ClimateChange
1174835595060944896 2019-09-20 00:00:07 UTC <Galaxysbod> I can't decide what does GRETA say????.
1174835571145019393 2019-09-20 00:00:02 UTC <JKrugenator> You opened up drilling in the arctic
ubuntu@ip-172-31-3-219:~$ twint -s 'greta' --since 2019-09-20 --until 2019-09-30 -l en -d true -o Sep2030Greta3.json74634444923133952 …

[mywindow30:bash*                                                                                                          "ip-172-31-3-219" 11:26 27-May-20
```

Figure 44: TMUC process for collection of Twitter data with TWINT

Time plot of collected data



In [1]:

```python
import twint
import nest_asyncio
import pandas as pd
nest_asyncio.apply()
import seaborn as sns
# package to clean text
import re
from nltk import FreqDist
pd.set_option('display.max_colwidth', -1)
```

In [2]:

```python
c = twint.Config()
c.Search = 'greta'
c.Lang = 'en'
c.Since = '2019-09-20'
c.Until = '2019-10-01'

c.Hide_output = True
c.Store_json = True
c.Output = 'Greta27MaySepJup.json'
c.Resume = 'GT_last2.csv'
c.Debug = True
```

In [ ]:

```python
twint.run.Search(c)
```

In [3]:

```python
df = pd.read_json('Greta27MaySepJup.json' , lines = True)
df.info()
```

...

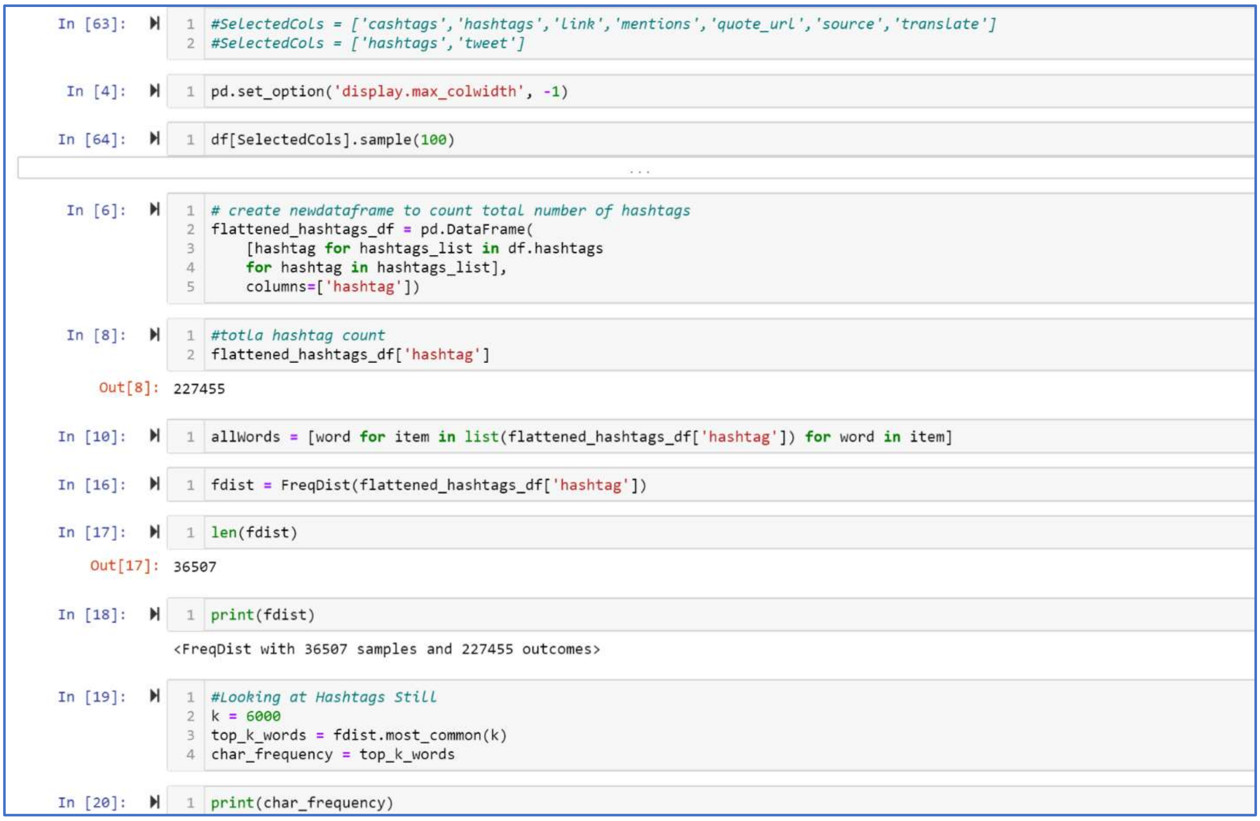
In [29]:

```python
%matplotlib inline
df['date'].value_counts().plot()
```

Out[29]:

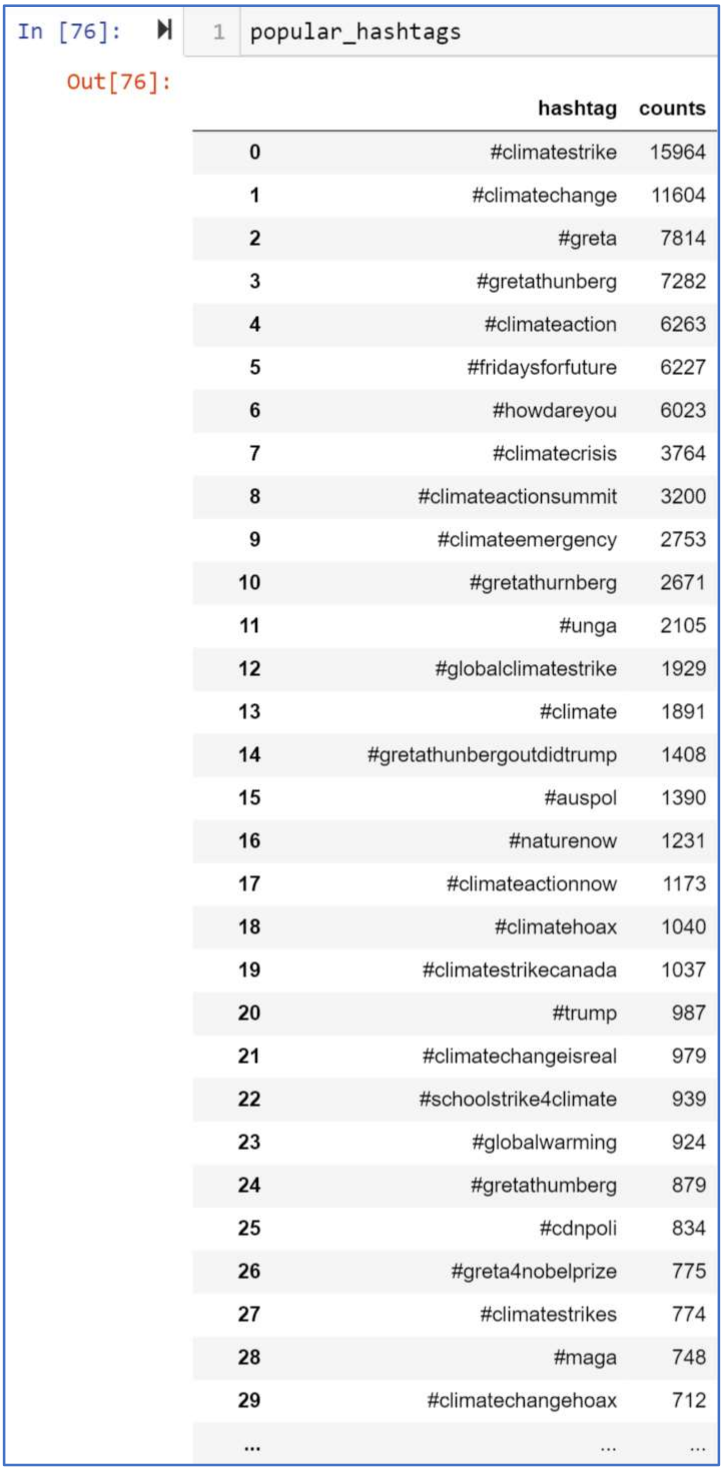
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbc6b80b4a8>
```

Figure 45: Initial exploration for the analysis of traffic



```
In [63]:  #SelectedCols = ['cashtags','hashtags','link','mentions','quote_url','source','translate']
          #SelectedCols = ['hashtags','tweet']

In [4]:   pd.set_option('display.max_colwidth', -1)

In [64]:  df[SelectedCols].sample(100)
          ...

In [6]:   # create newdataframe to count total number of hashtags
          flattened_hashtags_df = pd.DataFrame(
              [hashtag for hashtags_list in df.hashtags
              for hashtag in hashtags_list],
              columns=['hashtag'])

In [8]:   #totla hashtag count
          flattened_hashtags_df['hashtag']

Out[8]:   227455

In [10]:  allWords = [word for item in list(flattened_hashtags_df['hashtag']) for word in item]

In [16]:  fdist = FreqDist(flattened_hashtags_df['hashtag'])

In [17]:  len(fdist)

Out[17]:  36507

In [18]:  print(fdist)

          <FreqDist with 36507 samples and 227455 outcomes>

In [19]:  #Looking at Hashtags Still
          k = 6000
          top_k_words = fdist.most_common(k)
          char_frequency = top_k_words

In [20]:  print(char_frequency)
```

Figure 46: Examination of hashtag counts

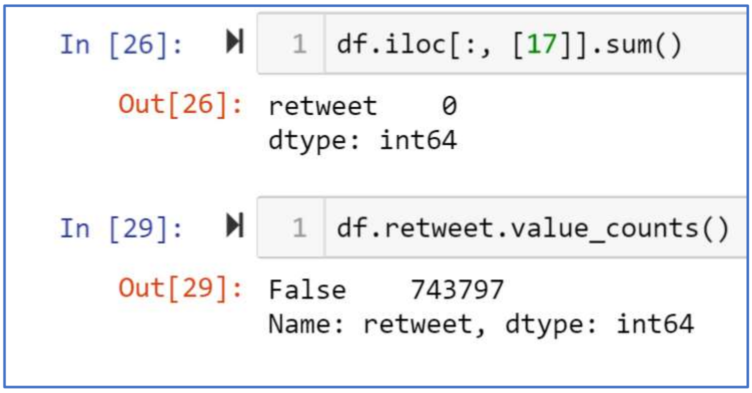


```
In [76]:   popular_hashtags
```

Out[76]:

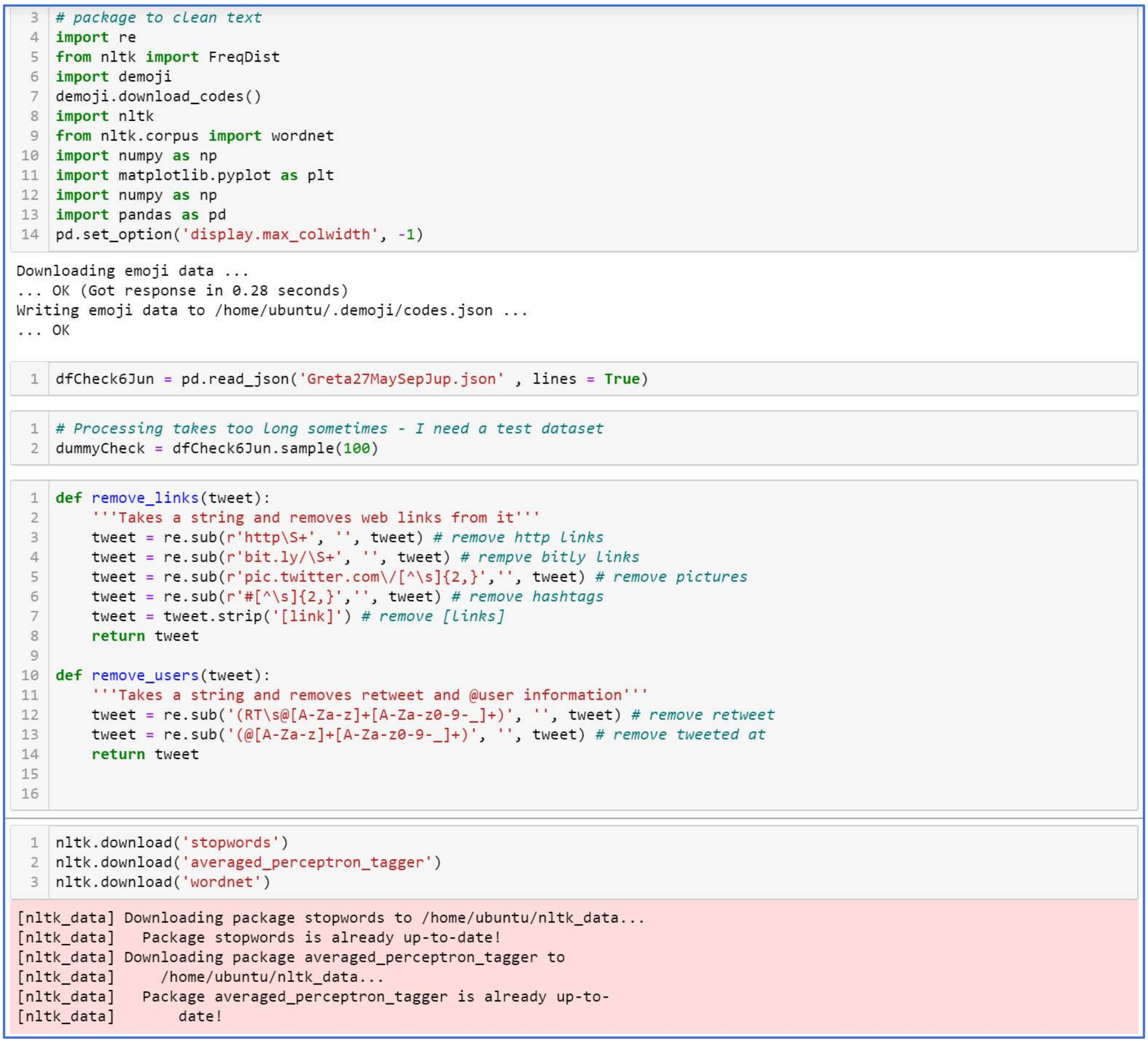
	hashtag	counts
0	#climatestrike	15964
1	#climatechange	11604
2	#greta	7814
3	#gretathunberg	7282
4	#climateaction	6263
5	#fridaysforfuture	6227
6	#howdareyou	6023
7	#climatecrisis	3764
8	#climateactionsummit	3200
9	#climateemergency	2753
10	#gretathurnberg	2671
11	#unga	2105
12	#globalclimatestrike	1929
13	#climate	1891
14	#gretathunbergoutdidtrump	1408
15	#auspol	1390
16	#naturenow	1231
17	#climateactionnow	1173
18	#climatehoax	1040
19	#climatestrikecanada	1037
20	#trump	987
21	#climatechangeisreal	979
22	#schoolstrike4climate	939
23	#globalwarming	924
24	#gretathumberg	879
25	#cdnpoli	834
26	#greta4nobelprize	775
27	#climatestrikes	774
28	#maga	748
29	#climatechangehoax	712
...	...	...

Figure 47: Most popular hashtags discovered



```
In [26]:   df.iloc[:, [17]].sum()

Out[26]:   retweet    0
           dtype: int64

In [29]:   df.retweet.value_counts()

Out[29]:   False    743797
           Name: retweet, dtype: int64
```

Figure 48: No retweets were found for the 740000 collected tweets



```
# package to clean text
import re
from nltk import FreqDist
import demoji
demoji.download_codes()
import nltk
from nltk.corpus import wordnet
import numpy as np
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
pd.set_option('display.max_colwidth', -1)
```

```
Downloading emoji data ...
... OK (Got response in 0.28 seconds)
Writing emoji data to /home/ubuntu/.demoji/codes.json ...
... OK
```

```
dfCheck6Jun = pd.read_json('Greta27MaySepJup.json' , lines = True)
```

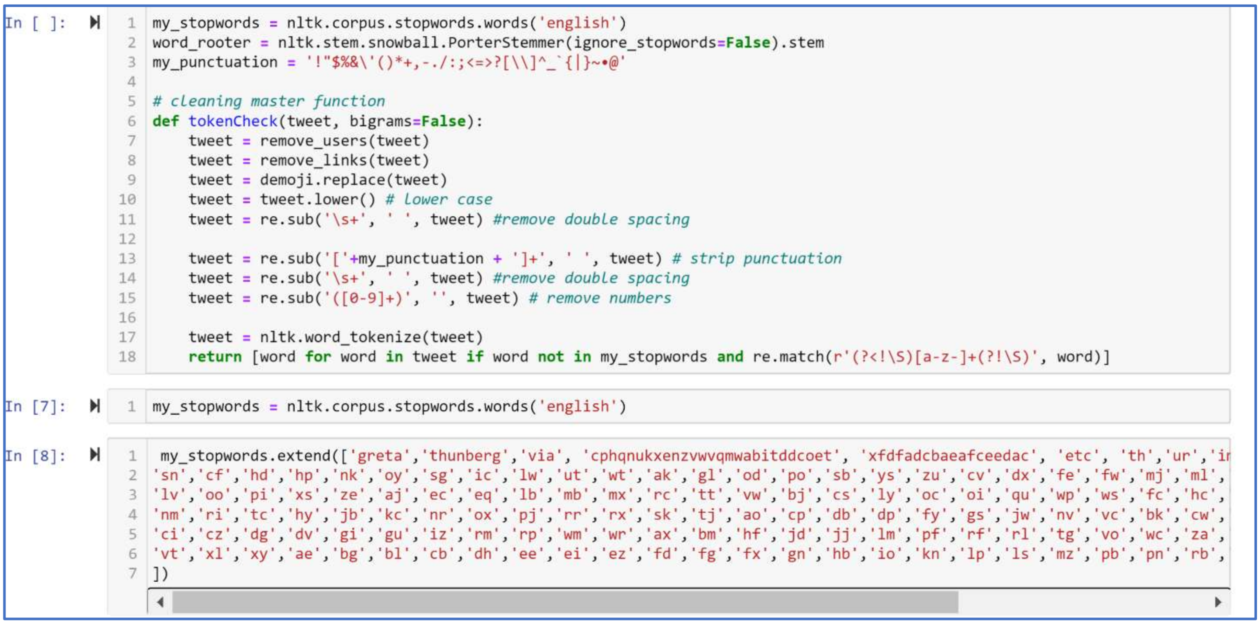
```
# Processing takes too long sometimes - I need a test dataset
dummyCheck = dfCheck6Jun.sample(100)
```

```
def remove_links(tweet):
    '''Takes a string and removes web links from it'''
    tweet = re.sub(r'http\S+', '', tweet) # remove http links
    tweet = re.sub(r'bit.ly/\S+', '', tweet) # rempve bitly links
    tweet = re.sub(r'pic.twitter.com\/[^\s]{2,}','', tweet) # remove pictures
    tweet = re.sub(r'#[^\s]{2,}','', tweet) # remove hashtags
    tweet = tweet.strip('[link]') # remove [links]
    return tweet

def remove_users(tweet):
    '''Takes a string and removes retweet and @user information'''
    tweet = re.sub('(RT\s@[A-Za-z]+[A-Za-z0-9-_]+)', '', tweet) # remove retweet
    tweet = re.sub('(@[A-Za-z]+[A-Za-z0-9-_]+)', '', tweet) # remove tweeted at
    return tweet


```

```
nltk.download('stopwords')
nltk.download('averaged_perceptron_tagger')
nltk.download('wordnet')
```

```
[nltk_data] Downloading package stopwords to /home/ubuntu/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /home/ubuntu/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

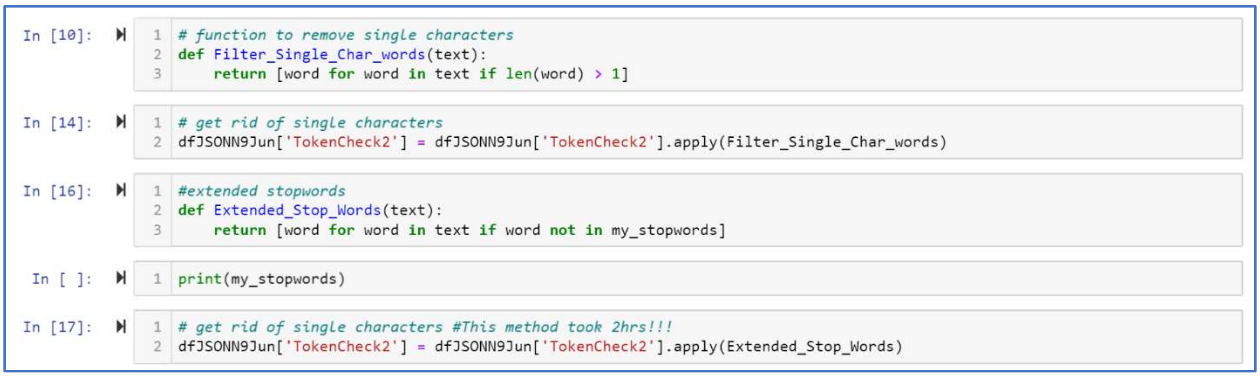
Figure 49: Functions used to clean tweets of metadata



```
In [ ]:
my_stopwords = nltk.corpus.stopwords.words('english')
word_rooter = nltk.stem.snowball.PorterStemmer(ignore_stopwords=False).stem
my_punctuation = '!"$%&\'()*+,-./:;<=>?[\\]^_`{|}~•@'

# cleaning master function
def tokenCheck(tweet, bigrams=False):
    tweet = remove_users(tweet)
    tweet = remove_links(tweet)
    tweet = demoji.replace(tweet)
    tweet = tweet.lower() # lower case
    tweet = re.sub('\s+', ' ', tweet) #remove double spacing

    tweet = re.sub('['+my_punctuation + ']+', ' ', tweet) # strip punctuation
    tweet = re.sub('\s+', ' ', tweet) #remove double spacing
    tweet = re.sub('([0-9]+)', '', tweet) # remove numbers

    tweet = nltk.word_tokenize(tweet)
    return [word for word in tweet if word not in my_stopwords and re.match(r'(?<!\S)[a-z-]+(?!\S)', word)]

In [7]:
my_stopwords = nltk.corpus.stopwords.words('english')

In [8]:
 my_stopwords.extend(['greta','thunberg','via', 'cphqnukxenzvwvqmwabitddcoet', 'xfdfadcbaeafceedac', 'etc', 'th','ur','ir
'sn','cf','hd','hp','nk','oy','sg','ic','lw','ut','wt','ak','gl','od','po','sb','ys','zu','cv','dx','fe','fw','mj','ml',
'lv','oo','pi','xs','ze','aj','ec','eq','lb','mb','mx','rc','tt','vw','bj','cs','ly','oc','oi','qu','wp','ws','fc','hc',
'nm','ri','tc','hy','jb','kc','nr','ox','pj','rr','rx','sk','tj','ao','cp','db','dp','fy','gs','jw','nv','vc','bk','cw',
'ci','cz','dg','dv','gi','gu','iz','rm','rp','wm','wr','ax','bm','hf','jd','jj','lm','pf','rf','rl','tg','vo','wc','za',
'vt','xl','xy','ae','bg','bl','cb','dh','ee','ei','ez','fd','fg','fx','gn','hb','io','kn','lp','ls','mz','pb','pn','rb',
])
```

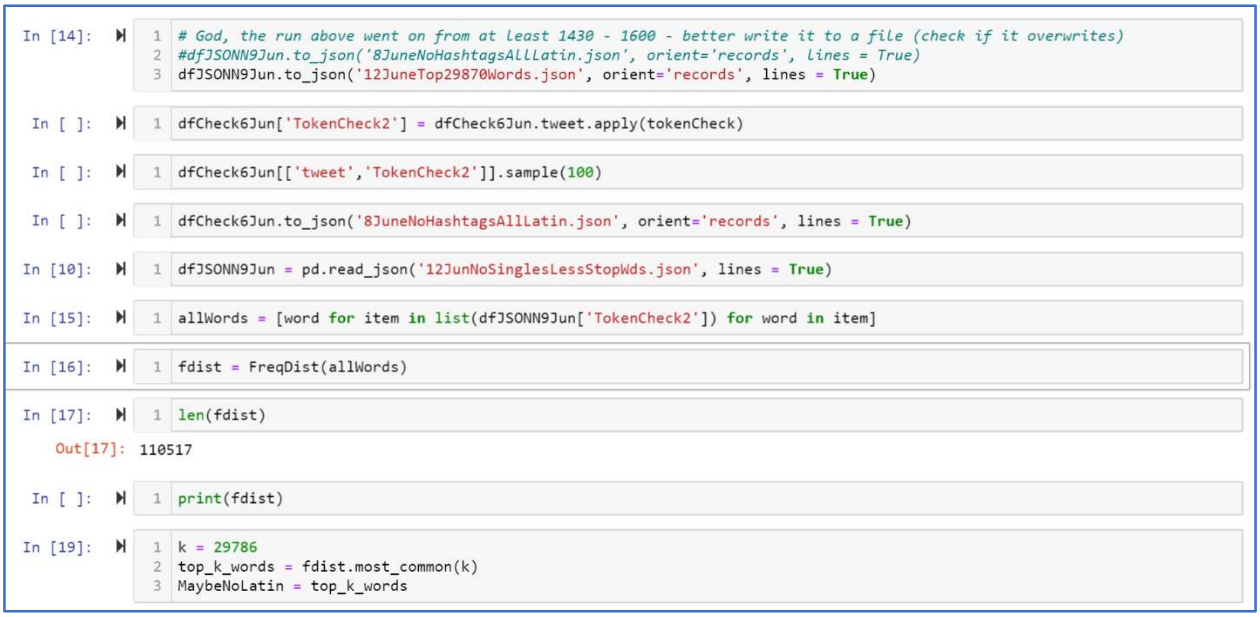
Figure 50: Central tweet processing method.

All numbers, punctuation and non-Latin characters are removed. Tweets are all converted to lower case



```
In [10]:
# function to remove single characters
def Filter_Single_Char_words(text):
    return [word for word in text if len(word) > 1]

In [14]:
# get rid of single characters
dfJSONN9Jun['TokenCheck2'] = dfJSONN9Jun['TokenCheck2'].apply(Filter_Single_Char_words)

In [16]:
#extended stopwords
def Extended_Stop_Words(text):
    return [word for word in text if word not in my_stopwords]

In [ ]:
print(my_stopwords)

In [17]:
# get rid of single characters #This method took 2hrs!!!
dfJSONN9Jun['TokenCheck2'] = dfJSONN9Jun['TokenCheck2'].apply(Extended_Stop_Words)
```

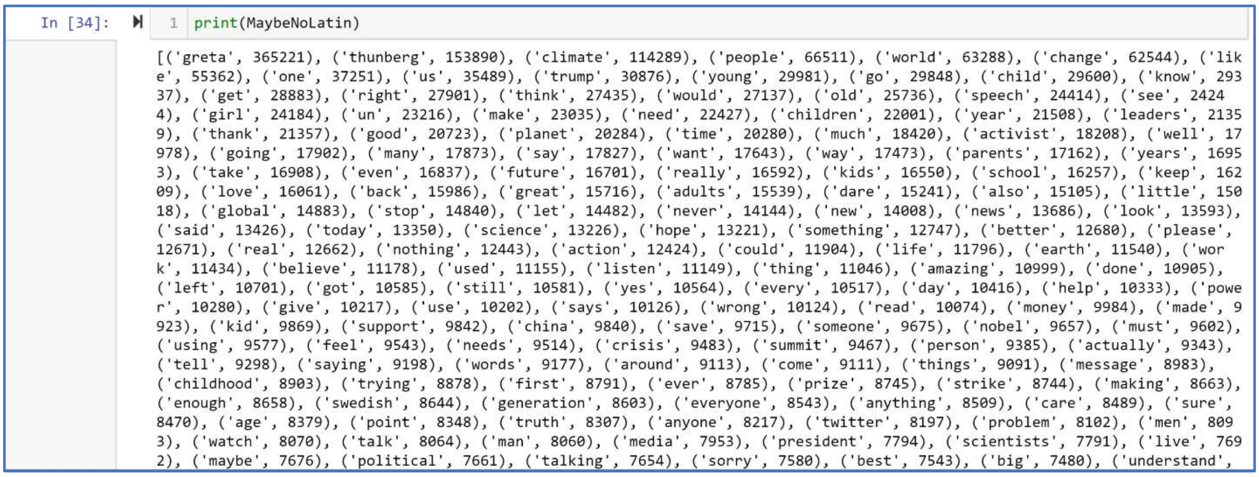
Figure 51: After initial preprocessing and examination, smaller more specific functions were run.



```
In [14]:  # God, the run above went on from at least 1430 - 1600 - better write it to a file (check if it overwrites)
          #dfJSONN9Jun.to_json('8JuneNoHashtagsAllLatin.json', orient='records', lines = True)
          dfJSONN9Jun.to_json('12JuneTop29870Words.json', orient='records', lines = True)

In [ ]:   dfCheck6Jun['TokenCheck2'] = dfCheck6Jun.tweet.apply(tokenCheck)

In [ ]:   dfCheck6Jun[['tweet','TokenCheck2']].sample(100)

In [ ]:   dfCheck6Jun.to_json('8JuneNoHashtagsAllLatin.json', orient='records', lines = True)

In [10]:  dfJSONN9Jun = pd.read_json('12JunNoSinglesLessStopWds.json', lines = True)

In [15]:  allWords = [word for item in list(dfJSONN9Jun['TokenCheck2']) for word in item]

In [16]:  fdist = FreqDist(allWords)

In [17]:  len(fdist)
Out[17]:  110517

In [ ]:   print(fdist)

In [19]:  k = 29786
          top_k_words = fdist.most_common(k)
          MaybeNoLatin = top_k_words
```

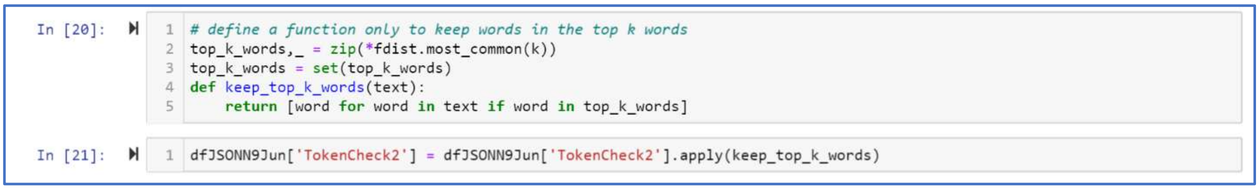
Figure 52: All unique tokens and their frequencies were examined.

29786 tokens had frequencies greater than 5



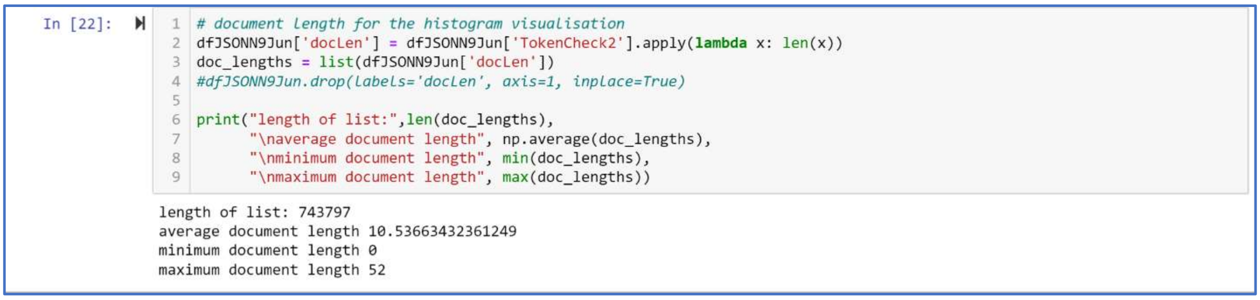
```
In [34]:  print(MaybeNoLatin)

[('greta', 365221), ('thunberg', 153890), ('climate', 114289), ('people', 66511), ('world', 63288), ('change', 62544), ('lik
e', 55362), ('one', 37251), ('us', 35489), ('trump', 30876), ('young', 29981), ('go', 29848), ('child', 29600), ('know', 293
37), ('get', 28883), ('right', 27901), ('think', 27435), ('would', 27137), ('old', 25736), ('speech', 24414), ('see', 2424
4), ('girl', 24184), ('un', 23216), ('make', 23035), ('need', 22427), ('children', 22001), ('year', 21508), ('leaders', 2135
9), ('thank', 21357), ('good', 20723), ('planet', 20284), ('time', 20280), ('much', 18420), ('activist', 18208), ('well', 17
978), ('going', 17902), ('many', 17873), ('say', 17827), ('want', 17643), ('way', 17473), ('parents', 17162), ('years', 1695
3), ('take', 16908), ('even', 16837), ('future', 16701), ('really', 16592), ('kids', 16550), ('school', 16257), ('keep', 162
09), ('love', 16061), ('back', 15986), ('great', 15716), ('adults', 15539), ('dare', 15241), ('also', 15105), ('little', 150
18), ('global', 14883), ('stop', 14840), ('let', 14482), ('never', 14144), ('new', 14008), ('news', 13686), ('look', 13593),
('said', 13426), ('today', 13350), ('science', 13226), ('hope', 13221), ('something', 12747), ('better', 12680), ('please',
12671), ('real', 12662), ('nothing', 12443), ('action', 12424), ('could', 11904), ('life', 11796), ('earth', 11540), ('wor
k', 11434), ('believe', 11178), ('used', 11155), ('listen', 11149), ('thing', 11046), ('amazing', 10999), ('done', 10905),
('left', 10701), ('got', 10585), ('still', 10581), ('yes', 10564), ('every', 10517), ('day', 10416), ('help', 10333), ('powe
r', 10280), ('give', 10217), ('use', 10202), ('says', 10126), ('wrong', 10124), ('read', 10074), ('money', 9984), ('made', 9
923), ('kid', 9869), ('support', 9842), ('china', 9840), ('save', 9715), ('someone', 9675), ('nobel', 9657), ('must', 9602),
('using', 9577), ('feel', 9543), ('needs', 9514), ('crisis', 9483), ('summit', 9467), ('person', 9385), ('actually', 9343),
('tell', 9298), ('saying', 9198), ('words', 9177), ('around', 9113), ('come', 9111), ('things', 9091), ('message', 8983),
('childhood', 8903), ('trying', 8878), ('first', 8791), ('ever', 8785), ('prize', 8745), ('strike', 8744), ('making', 8663),
('enough', 8658), ('swedish', 8644), ('generation', 8603), ('everyone', 8543), ('anything', 8509), ('care', 8489), ('sure',
8470), ('age', 8379), ('point', 8348), ('truth', 8307), ('anyone', 8217), ('twitter', 8197), ('problem', 8102), ('men', 809
3), ('watch', 8070), ('talk', 8064), ('man', 8060), ('media', 7953), ('president', 7794), ('scientists', 7791), ('live', 769
2), ('maybe', 7676), ('political', 7661), ('talking', 7654), ('sorry', 7580), ('best', 7543), ('big', 7480), ('understand',
```

Figure 53: Tokens and their Frequencies



```
In [20]:  # define a function only to keep words in the top k words
          top_k_words,_ = zip(*fdist.most_common(k))
          top_k_words = set(top_k_words)
          def keep_top_k_words(text):
              return [word for word in text if word in top_k_words]

In [21]:  dfJSONN9Jun['TokenCheck2'] = dfJSONN9Jun['TokenCheck2'].apply(keep_top_k_words)
```

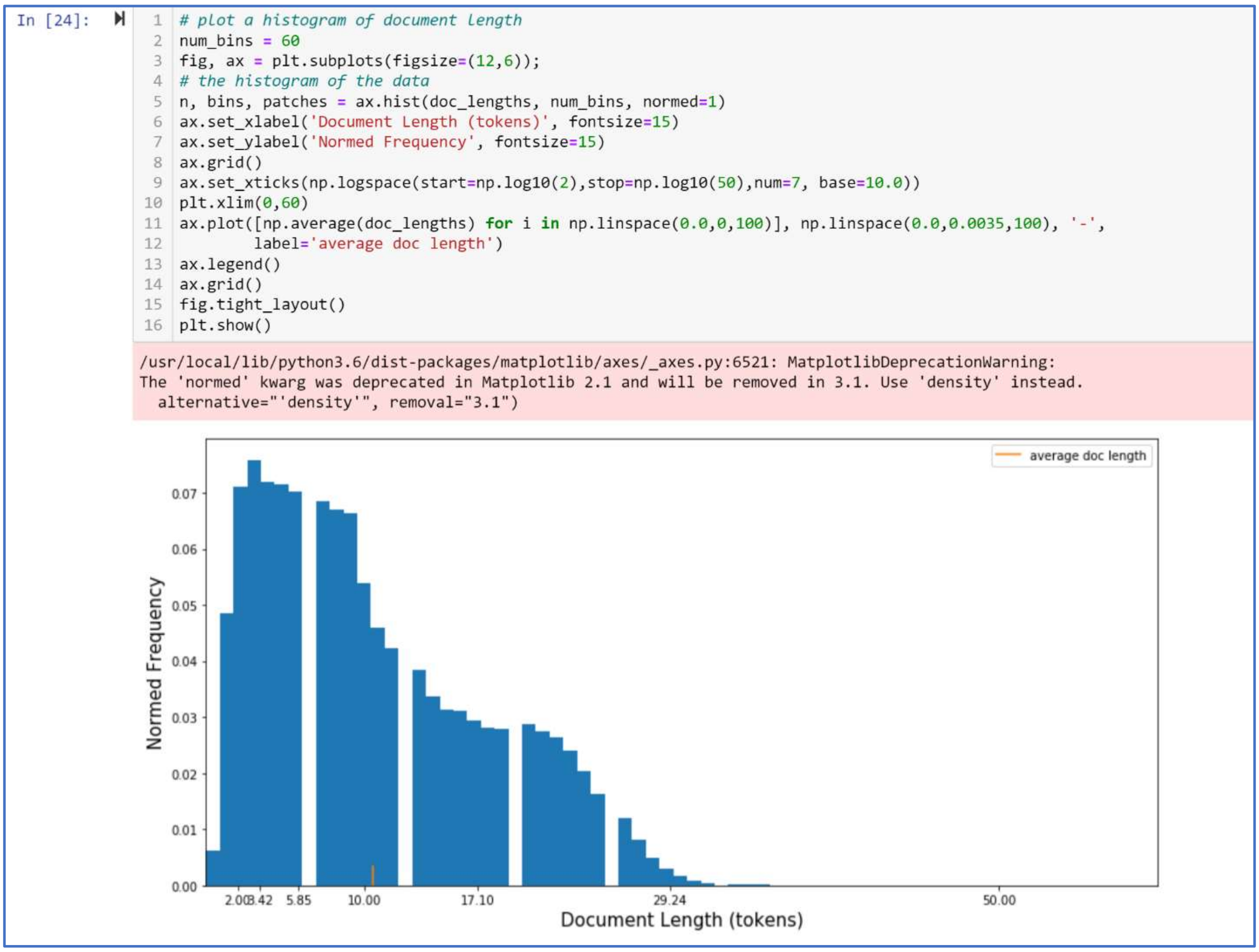
Figure 54: Method to only keep words in the corpus with frequencies greater than 5



```
In [22]:
# document length for the histogram visualisation
dfJSONN9Jun['docLen'] = dfJSONN9Jun['TokenCheck2'].apply(lambda x: len(x))
doc_lengths = list(dfJSONN9Jun['docLen'])
#dfJSONN9Jun.drop(labels='docLen', axis=1, inplace=True)

print("length of list:",len(doc_lengths),
      "\naverage document length", np.average(doc_lengths),
      "\nminimum document length", min(doc_lengths),
      "\nmaximum document length", max(doc_lengths))
```

```
length of list: 743797
average document length 10.53663432361249
minimum document length 0
maximum document length 52
```

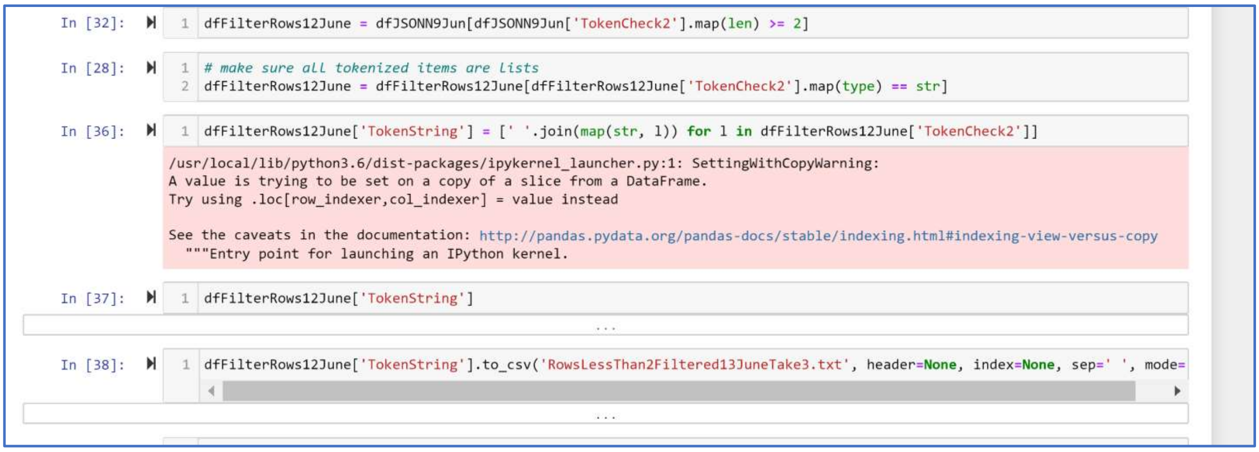
Figure 55: Examination of average document length



```
In [24]:
# plot a histogram of document length
num_bins = 60
fig, ax = plt.subplots(figsize=(12,6));
# the histogram of the data
n, bins, patches = ax.hist(doc_lengths, num_bins, normed=1)
ax.set_xlabel('Document Length (tokens)', fontsize=15)
ax.set_ylabel('Normed Frequency', fontsize=15)
ax.grid()
ax.set_xticks(np.logspace(start=np.log10(2),stop=np.log10(50),num=7, base=10.0))
plt.xlim(0,60)
ax.plot([np.average(doc_lengths) for i in np.linspace(0.0,0,100)], np.linspace(0.0,0.0035,100), '-',
         label='average doc length')
ax.legend()
ax.grid()
fig.tight_layout()
plt.show()
```

```
/usr/local/lib/python3.6/dist-packages/matplotlib/axes/_axes.py:6521: MatplotlibDeprecationWarning:
The 'normed' kwarg was deprecated in Matplotlib 2.1 and will be removed in 3.1. Use 'density' instead.
  alternative="'density'", removal="3.1")
```

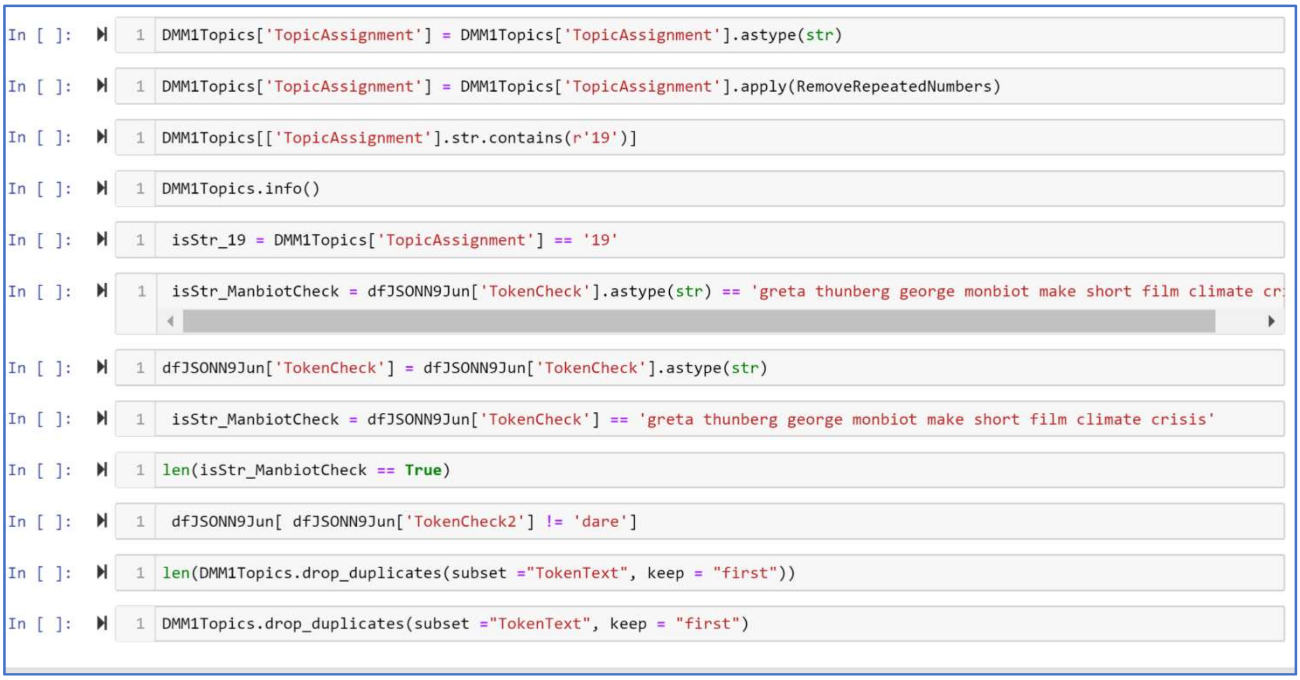
Figure 56: Document length distribution



```
In [32]:  dfFilterRows12June = dfJSONN9Jun[dfJSONN9Jun['TokenCheck2'].map(len) >= 2]

In [28]:  # make sure all tokenized items are lists
          dfFilterRows12June = dfFilterRows12June[dfFilterRows12June['TokenCheck2'].map(type) == str]

In [36]:  dfFilterRows12June['TokenString'] = [' '.join(map(str, l)) for l in dfFilterRows12June['TokenCheck2']]

/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
  """Entry point for launching an IPython kernel.

In [37]:  dfFilterRows12June['TokenString']
...

In [38]:  dfFilterRows12June['TokenString'].to_csv('RowsLessThan2Filtered13JuneTake3.txt', header=None, index=None, sep=' ', mode=
...
```

Figure 57: Create Token String Variable for Corpus format

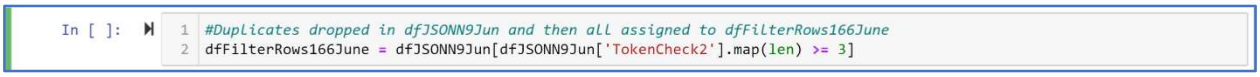
Only keep documents with 3 or more tokens.



```
In [ ]:  DMM1Topics['TopicAssignment'] = DMM1Topics['TopicAssignment'].astype(str)

In [ ]:  DMM1Topics['TopicAssignment'] = DMM1Topics['TopicAssignment'].apply(RemoveRepeatedNumbers)

In [ ]:  DMM1Topics[['TopicAssignment'].str.contains(r'19')]

In [ ]:  DMM1Topics.info()

In [ ]:   isStr_19 = DMM1Topics['TopicAssignment'] == '19'

In [ ]:   isStr_ManbiotCheck = dfJSONN9Jun['TokenCheck'].astype(str) == 'greta thunberg george monbiot make short film climate cr

In [ ]:  dfJSONN9Jun['TokenCheck'] = dfJSONN9Jun['TokenCheck'].astype(str)

In [ ]:   isStr_ManbiotCheck = dfJSONN9Jun['TokenCheck'] == 'greta thunberg george monbiot make short film climate crisis'

In [ ]:  len(isStr_ManbiotCheck == True)

In [ ]:   dfJSONN9Jun[ dfJSONN9Jun['TokenCheck2'] != 'dare']

In [ ]:  len(DMM1Topics.drop_duplicates(subset ="TokenText", keep = "first"))

In [ ]:  DMM1Topics.drop_duplicates(subset ="TokenText", keep = "first")
```

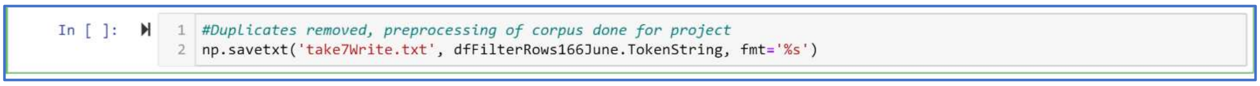
Figure 58: Corpus was examined for Duplicate Documents.

Despite no retweets, there were many long duplicated documents



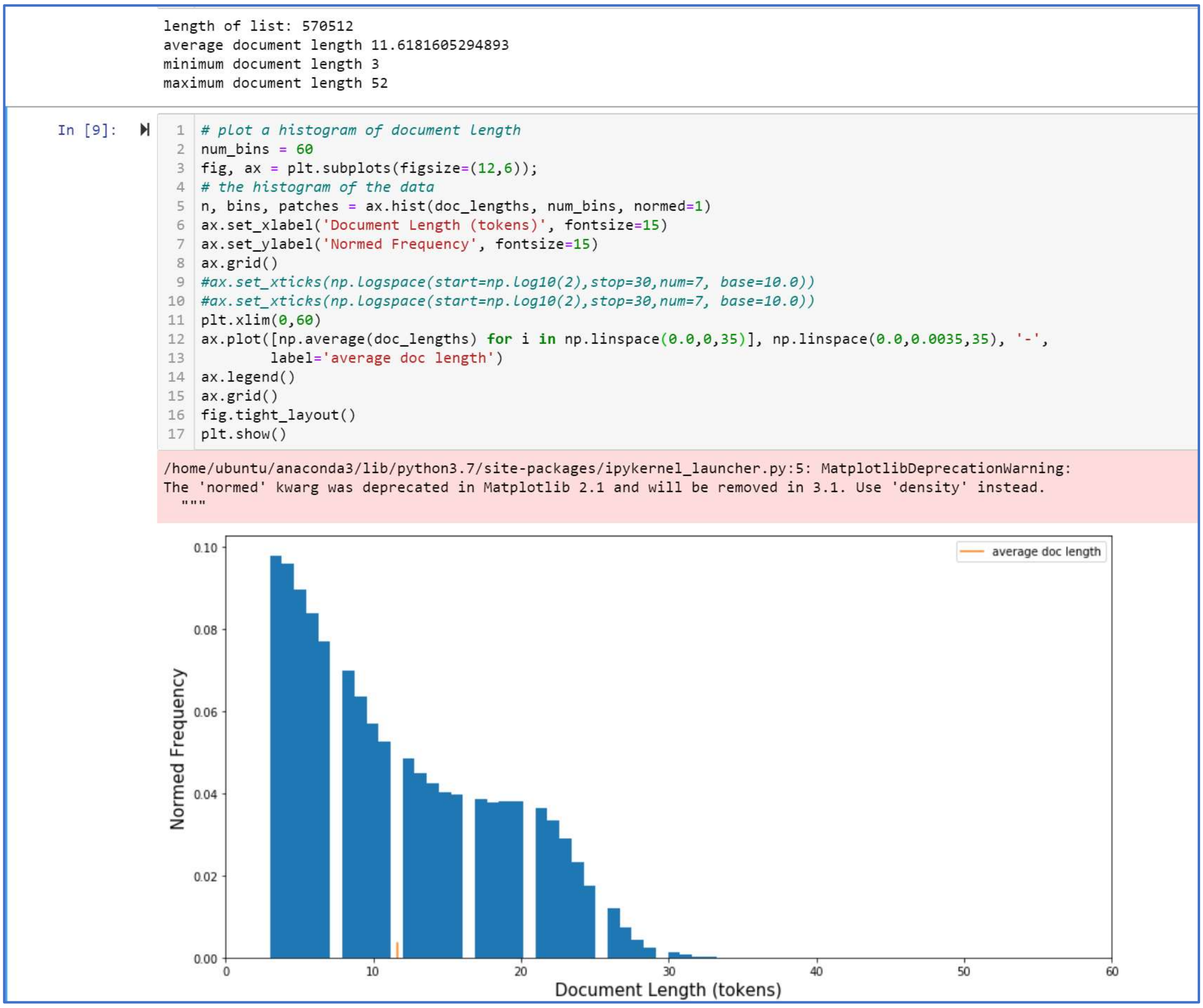
```
In [ ]:  #Duplicates dropped in dfJSONN9Jun and then all assigned to dfFilterRows166June
         dfFilterRows166June = dfJSONN9Jun[dfJSONN9Jun['TokenCheck2'].map(len) >= 3]
```

Figure 59: Tweets with no duplicates tweets were saved to a new dataframe



```
In [ ]:  #Duplicates removed, preprocessing of corpus done for project
         np.savetxt('take7Write.txt', dfFilterRows166June.TokenString, fmt='%s')
```

Figure 60: Final Corpus Saved



```
length of list: 570512
average document length 11.6181605294893
minimum document length 3
maximum document length 52
```

In [9]:

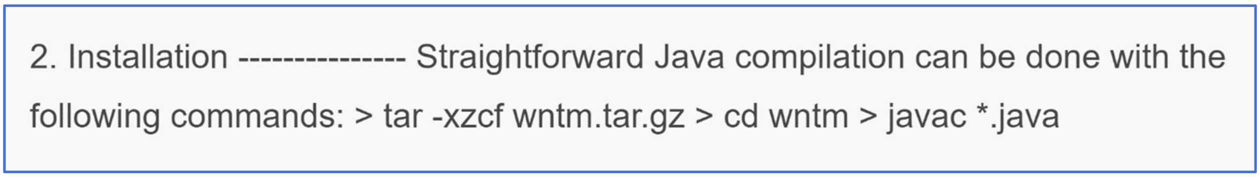
```
# plot a histogram of document length
num_bins = 60
fig, ax = plt.subplots(figsize=(12,6));
# the histogram of the data
n, bins, patches = ax.hist(doc_lengths, num_bins, normed=1)
ax.set_xlabel('Document Length (tokens)', fontsize=15)
ax.set_ylabel('Normed Frequency', fontsize=15)
ax.grid()
#ax.set_xticks(np.logspace(start=np.log10(2),stop=30,num=7, base=10.0))
#ax.set_xticks(np.logspace(start=np.log10(2),stop=30,num=7, base=10.0))
plt.xlim(0,60)
ax.plot([np.average(doc_lengths) for i in np.linspace(0.0,0,35)], np.linspace(0.0,0.0035,35), '-',
        label='average doc length')
ax.legend()
ax.grid()
fig.tight_layout()
plt.show()
```

```
/home/ubuntu/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:5: MatplotlibDeprecationWarning:
The 'normed' kwarg was deprecated in Matplotlib 2.1 and will be removed in 3.1. Use 'density' instead.
  """
```

Figure 61: Average document length after removal of duplicate tweets

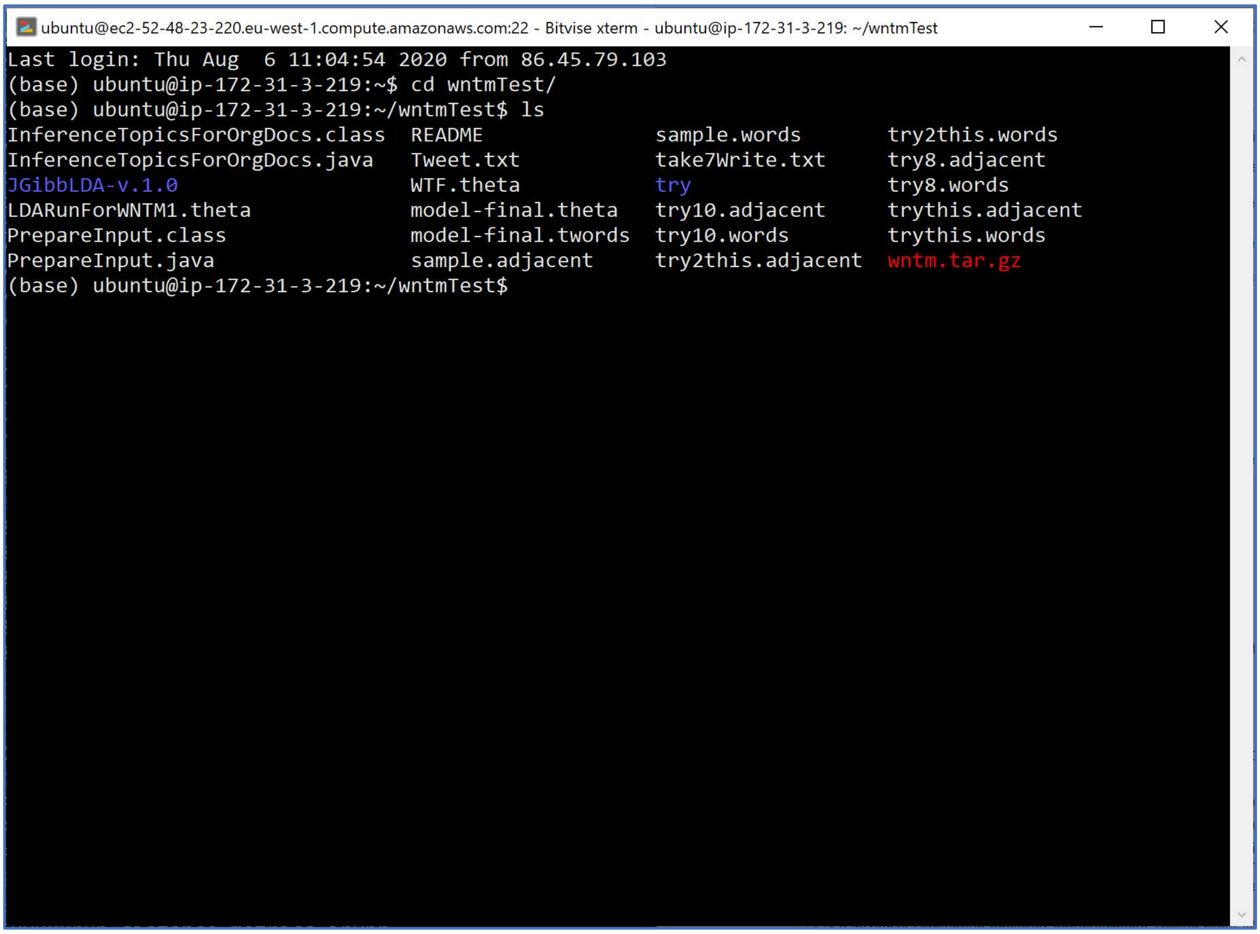
3 Implementation of Topic Models

WNTM[1]:



2. Installation --------------- Straightforward Java compilation can be done with the following commands: > tar -xzcf wntm.tar.gz > cd wntm > javac *.java

Figure 62: Installation of WNTM. Code is unzipped and then Compiled with Java

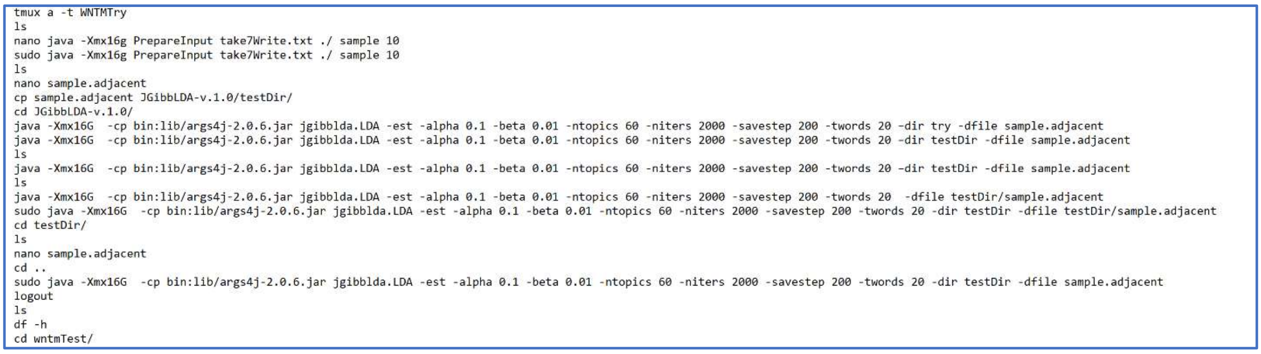


ubuntu@ec2-52-48-23-220.eu-west-1.compute.amazonaws.com:22 - Bitvise xterm - ubuntu@ip-172-31-3-219: ~/wntmTest

```
Last login: Thu Aug  6 11:04:54 2020 from 86.45.79.103
(base) ubuntu@ip-172-31-3-219:~$ cd wntmTest/
(base) ubuntu@ip-172-31-3-219:~/wntmTest$ ls
InferenceTopicsForOrgDocs.class  README               sample.words       try2this.words
InferenceTopicsForOrgDocs.java   Tweet.txt            take7Write.txt     try8.adjacent
JGibbLDA-v.1.0                   WTF.theta            try                try8.words
LDARunForWNTM1.theta             model-final.theta    try10.adjacent     trythis.adjacent
PrepareInput.class               model-final.twords   try10.words        trythis.words
PrepareInput.java                sample.adjacent      try2this.adjacent  wntm.tar.gz
(base) ubuntu@ip-172-31-3-219:~/wntmTest$
```

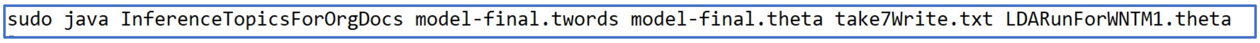
Figure 63: WNTM directory

[1] https://figshare.com/articles/Code_of_word_network_topic_model/5572591



```
tmux a -t WNTMTry
ls
nano java -Xmx16g PrepareInput take7Write.txt ./ sample 10
sudo java -Xmx16g PrepareInput take7Write.txt ./ sample 10
ls
nano sample.adjacent
cp sample.adjacent JGibbLDA-v.1.0/testDir/
cd JGibbLDA-v.1.0/
java -Xmx16G  -cp bin:lib/args4j-2.0.6.jar jgibblda.LDA -est -alpha 0.1 -beta 0.01 -ntopics 60 -niters 2000 -savestep 200 -twords 20 -dir try -dfile sample.adjacent
java -Xmx16G  -cp bin:lib/args4j-2.0.6.jar jgibblda.LDA -est -alpha 0.1 -beta 0.01 -ntopics 60 -niters 2000 -savestep 200 -twords 20 -dir testDir -dfile sample.adjacent
ls
java -Xmx16G  -cp bin:lib/args4j-2.0.6.jar jgibblda.LDA -est -alpha 0.1 -beta 0.01 -ntopics 60 -niters 2000 -savestep 200 -twords 20 -dir testDir -dfile sample.adjacent
ls
java -Xmx16G  -cp bin:lib/args4j-2.0.6.jar jgibblda.LDA -est -alpha 0.1 -beta 0.01 -ntopics 60 -niters 2000 -savestep 200 -twords 20  -dfile testDir/sample.adjacent
sudo java -Xmx16G  -cp bin:lib/args4j-2.0.6.jar jgibblda.LDA -est -alpha 0.1 -beta 0.01 -ntopics 60 -niters 2000 -savestep 200 -twords 20 -dir testDir -dfile testDir/sample.adjacent
cd testDir/
ls
nano sample.adjacent
cd ..
sudo java -Xmx16G  -cp bin:lib/args4j-2.0.6.jar jgibblda.LDA -est -alpha 0.1 -beta 0.01 -ntopics 60 -niters 2000 -savestep 200 -twords 20 -dir testDir -dfile sample.adjacent
logout
ls
df -h
cd wntmTest/
```

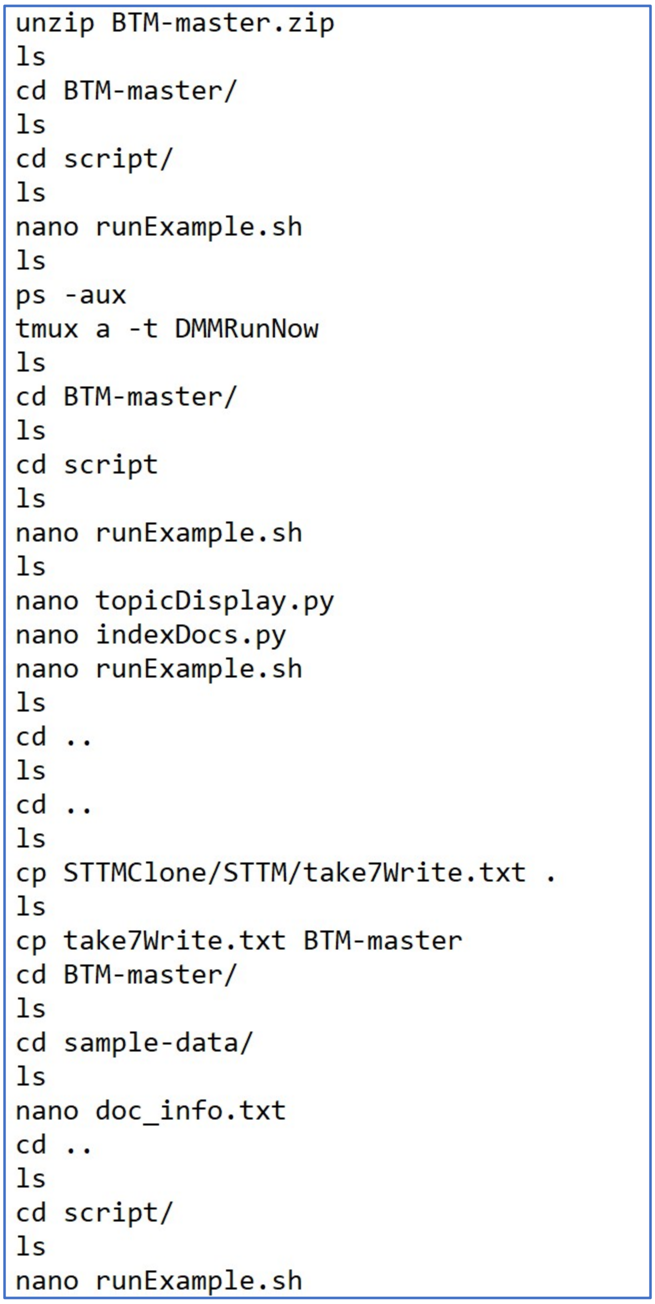
Figure 64: Creation of Word nodes for LDA topic generation



```
sudo java InferenceTopicsForOrgDocs model-final.twords model-final.theta take7Write.txt LDARunForWNTM1.theta
```

Figure 65: Inference of theta from LDA with Word Node Corpus onto the documents themselves

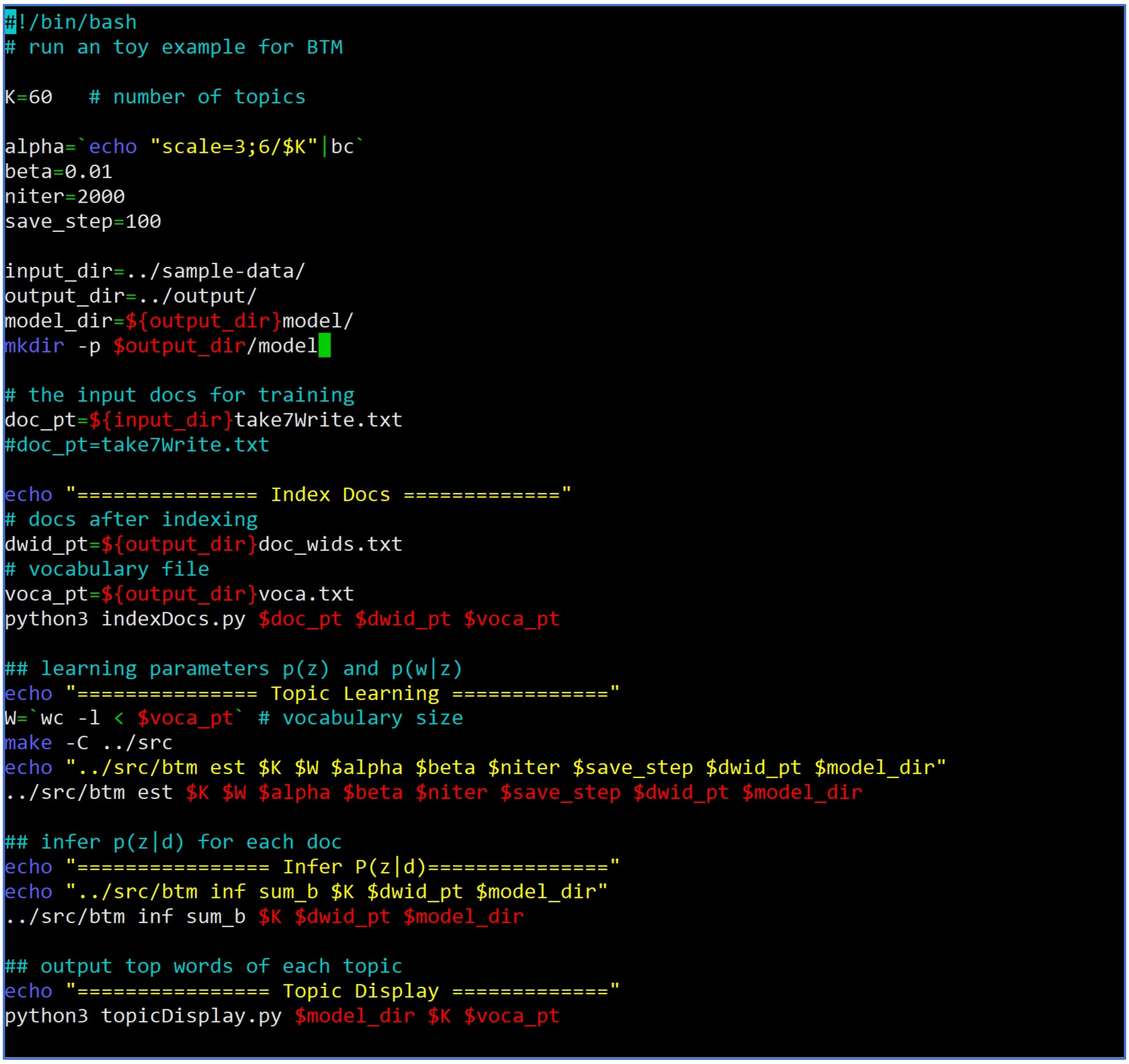
BTM[2]:



```
unzip BTM-master.zip
ls
cd BTM-master/
ls
cd script/
ls
nano runExample.sh
ls
ps -aux
tmux a -t DMMRunNow
ls
cd BTM-master/
ls
cd script
ls
nano runExample.sh
ls
nano topicDisplay.py
nano indexDocs.py
nano runExample.sh
ls
cd ..
ls
cd ..
ls
cp STTMClone/STTM/take7Write.txt .
ls
cp take7Write.txt BTM-master
cd BTM-master/
ls
cd sample-data/
ls
nano doc_info.txt
cd ..
ls
cd script/
ls
nano runExample.sh
```

Figure 66: Unzipping and running BTM

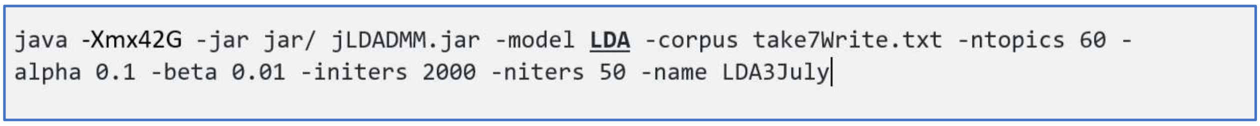
[2] https://github.com/xiaohuiyan/BTM/tree/master/src



```
#!/bin/bash
# run an toy example for BTM

K=60   # number of topics

alpha=`echo "scale=3;6/$K"|bc`
beta=0.01
niter=2000
save_step=100

input_dir=../sample-data/
output_dir=../output/
model_dir=${output_dir}model/
mkdir -p $output_dir/model

# the input docs for training
doc_pt=${input_dir}take7Write.txt
#doc_pt=take7Write.txt

echo "=============== Index Docs ============="
# docs after indexing
dwid_pt=${output_dir}doc_wids.txt
# vocabulary file
voca_pt=${output_dir}voca.txt
python3 indexDocs.py $doc_pt $dwid_pt $voca_pt

## learning parameters p(z) and p(w|z)
echo "=============== Topic Learning ============="
W=`wc -l < $voca_pt` # vocabulary size
make -C ../src
echo "../src/btm est $K $W $alpha $beta $niter $save_step $dwid_pt $model_dir"
../src/btm est $K $W $alpha $beta $niter $save_step $dwid_pt $model_dir

## infer p(z|d) for each doc
echo "================ Infer P(z|d)==============="
echo "../src/btm inf sum_b $K $dwid_pt $model_dir"
../src/btm inf sum_b $K $dwid_pt $model_dir

## output top words of each topic
echo "================ Topic Display ============="
python3 topicDisplay.py $model_dir $K $voca_pt
```

Figure 67: Parameter Specification for BTM. Corpus location and output location are also specified

LDA[3]:



```
java -Xmx42G -jar jar/ jLDADMM.jar -model LDA -corpus take7Write.txt -ntopics 60 -
alpha 0.1 -beta 0.01 -initers 2000 -niters 50 -name LDA3July
```

Figure 68: Execution of LDA algorithm. Parameters and JVM memory allocation are specified

[3] https://github.com/datquocnguyen/jLDADMM

DMM[4]:

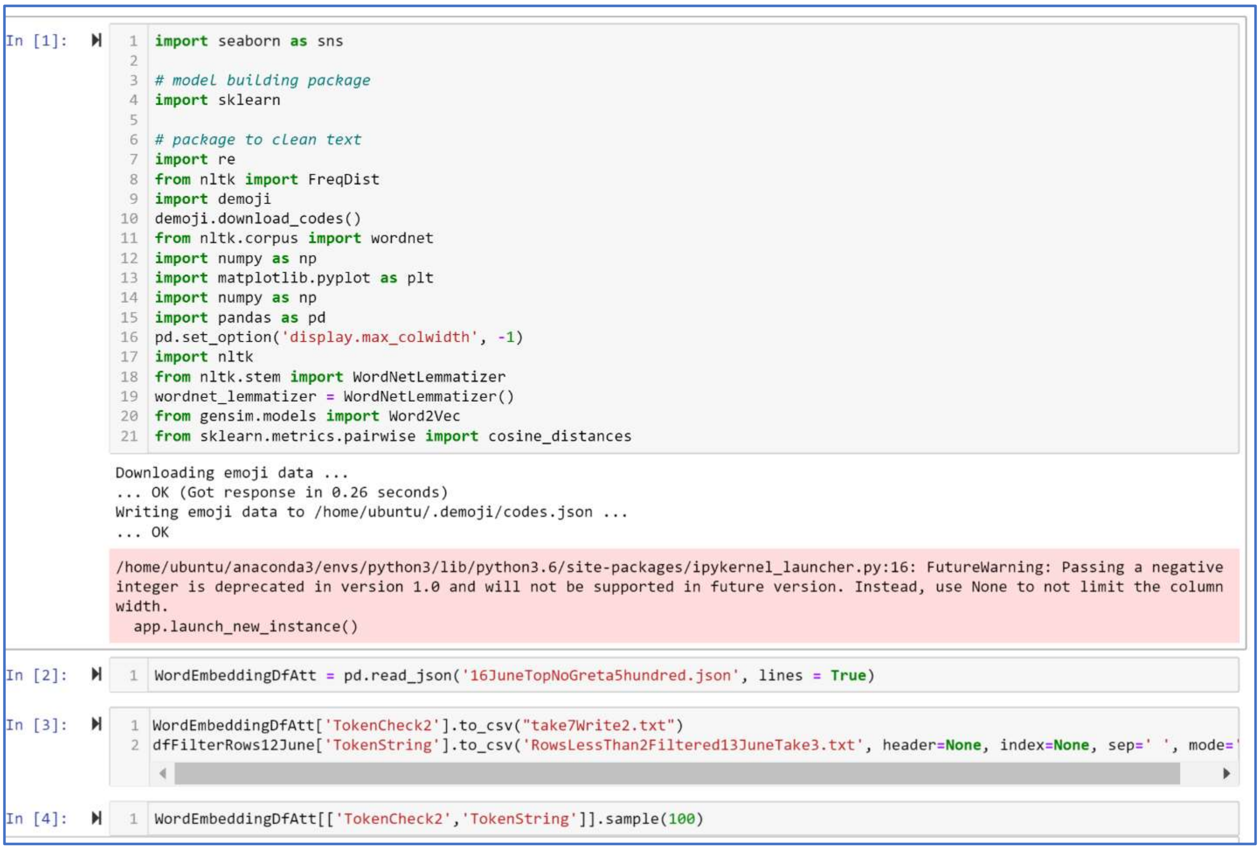
```
java -Xmx42G -jar jar/jLDADMM.jar -model DMM -corpus take7Write.txt -ntopics 60 -
alpha 0.1 -beta 0.01 -initers 2000 -niters 50 -name DMMDat3July
```

Figure 69: Execution of DMM algorithm. Parameters and JVM memory allocation are specified

LF-DMM[5]:

```
java -Xmx42G -jar jar/LFTM.jar -model LFDMM -corpus take7Write.txt -vectors
WordVec100skiptake1.txt -ntopics 60 -alpha 0.1 -beta 0.01 -lambda 0.5 -initers 2000 -
niters 200 -name LFDMM1July
```

Figure 70: Execution of LFDMM. Corpus, parameters and Trained word embedding are specified



```
In [1]:
import seaborn as sns

# model building package
import sklearn

# package to clean text
import re
from nltk import FreqDist
import demoji
demoji.download_codes()
from nltk.corpus import wordnet
import numpy as np
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
pd.set_option('display.max_colwidth', -1)
import nltk
from nltk.stem import WordNetLemmatizer
wordnet_lemmatizer = WordNetLemmatizer()
from gensim.models import Word2Vec
from sklearn.metrics.pairwise import cosine_distances
```

```
Downloading emoji data ...
... OK (Got response in 0.26 seconds)
Writing emoji data to /home/ubuntu/.demoji/codes.json ...
... OK
```

```
/home/ubuntu/anaconda3/envs/python3/lib/python3.6/site-packages/ipykernel_launcher.py:16: FutureWarning: Passing a negative
integer is deprecated in version 1.0 and will not be supported in future version. Instead, use None to not limit the column
width.
  app.launch_new_instance()
```

```
In [2]:
WordEmbeddingDfAtt = pd.read_json('16JuneTopNoGreta5hundred.json', lines = True)
```

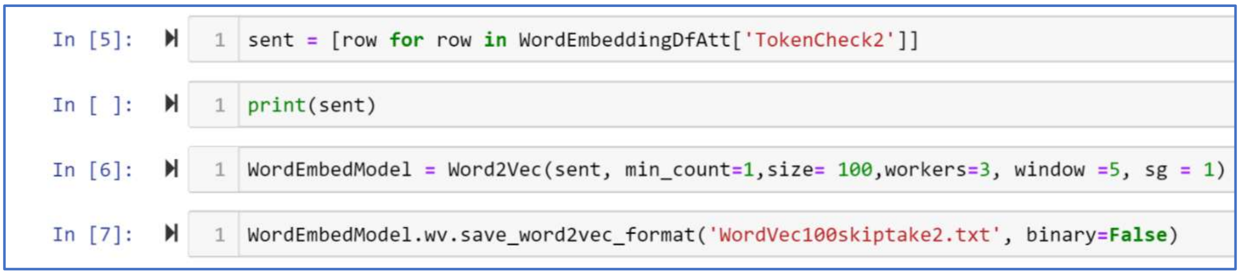
```
In [3]:
WordEmbeddingDfAtt['TokenCheck2'].to_csv("take7Write2.txt")
dfFilterRows12June['TokenString'].to_csv('RowsLessThan2Filtered13JuneTake3.txt', header=None, index=None, sep=' ', mode='
```

```
In [4]:
WordEmbeddingDfAtt[['TokenCheck2','TokenString']].sample(100)
```

Figure 71: Libraries for word embedding generation and corpus for word embedding

[4] https://github.com/datquocnguyen/jLDADMM

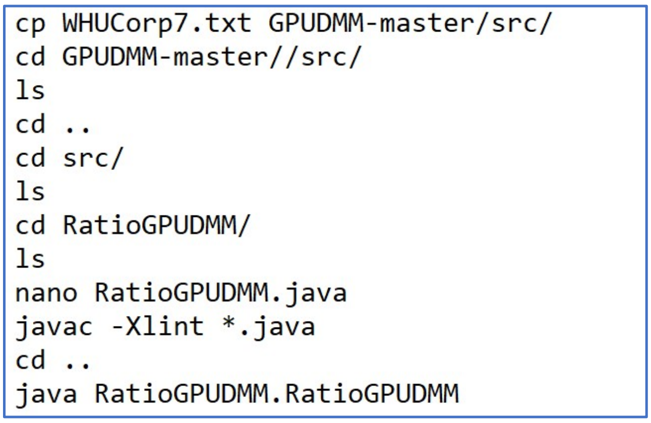
[5] https://github.com/datquocnguyen/LFTM



```
In [5]:  sent = [row for row in WordEmbeddingDfAtt['TokenCheck2']]

In [ ]:  print(sent)

In [6]:  WordEmbedModel = Word2Vec(sent, min_count=1,size= 100,workers=3, window =5, sg = 1)

In [7]:  WordEmbedModel.wv.save_word2vec_format('WordVec100skiptake2.txt', binary=False)
```

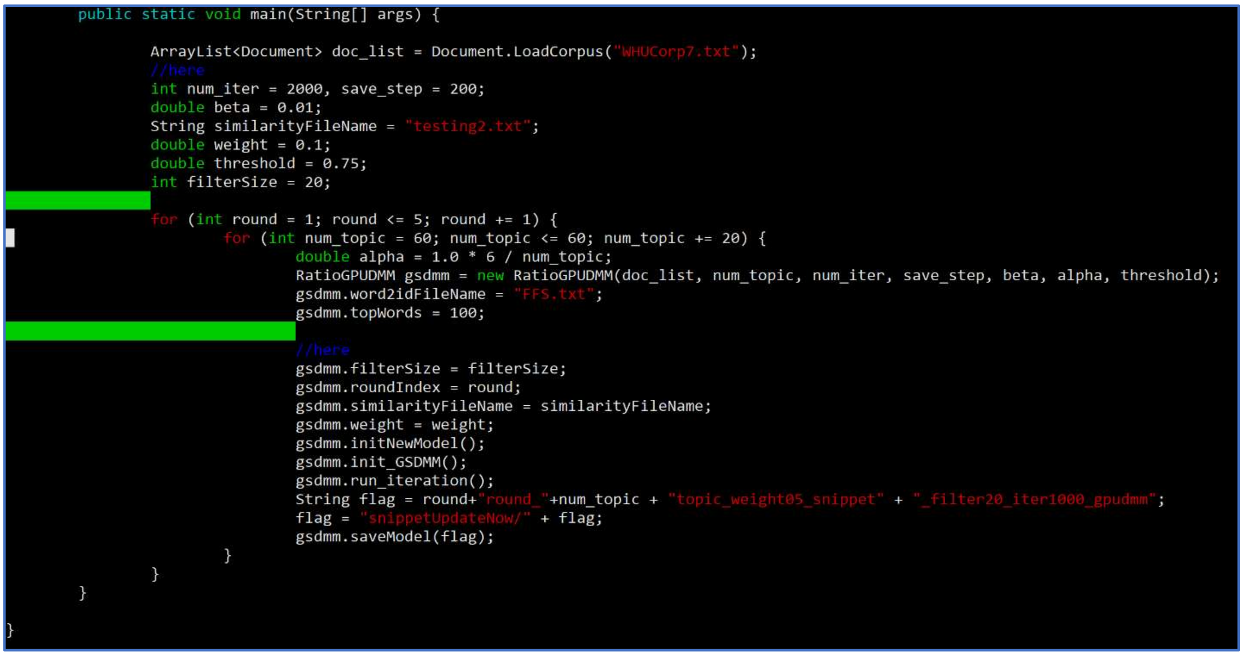
Figure 72: Word Embedding Training for LFDMM

GPU-DMM[6]:



```
cp WHUCorp7.txt GPUDMM-master/src/
cd GPUDMM-master//src/
ls
cd ..
cd src/
ls
cd RatioGPUDMM/
ls
nano RatioGPUDMM.java
javac -Xlint *.java
cd ..
java RatioGPUDMM.RatioGPUDMM
```

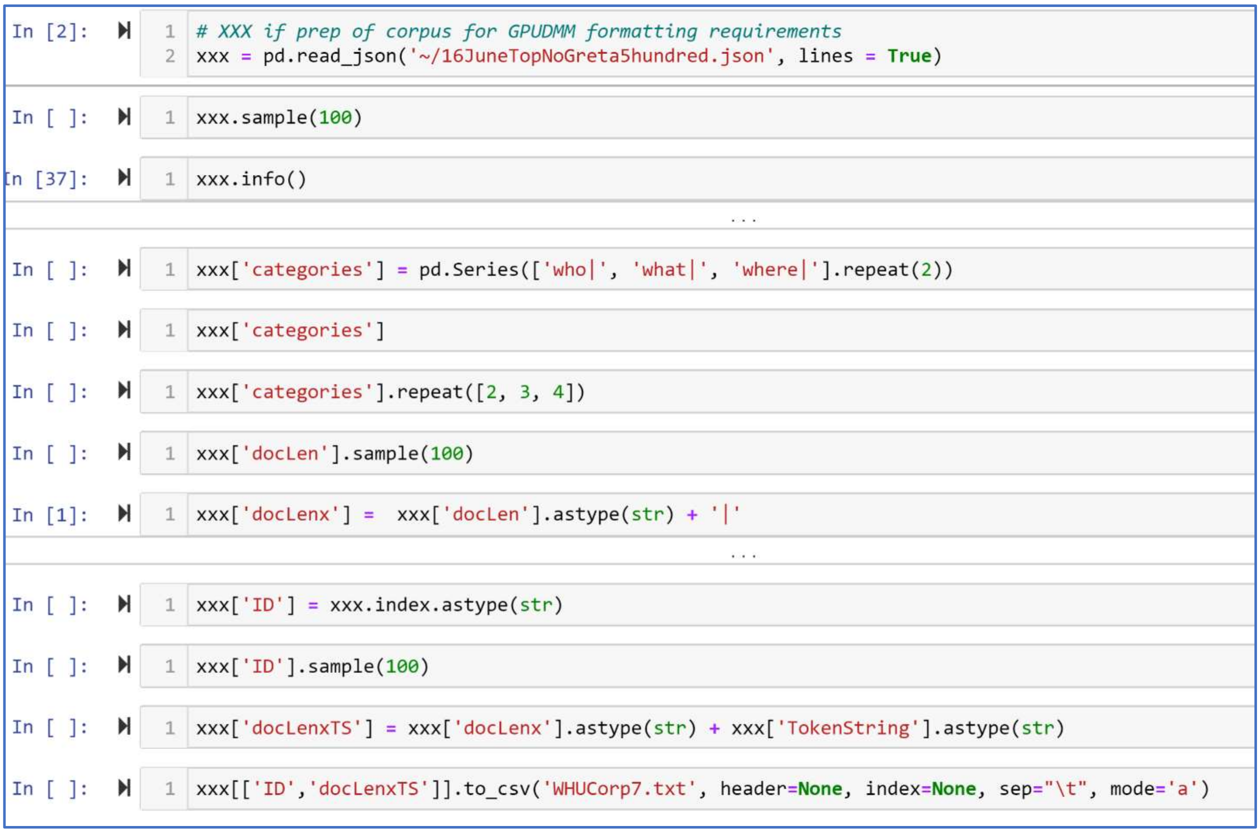
Figure 73: RatioGPUDMM parameters changed and code is recompiled



```
        public static void main(String[] args) {

                ArrayList<Document> doc_list = Document.LoadCorpus("WHUCorp7.txt");
                //here
                int num_iter = 2000, save_step = 200;
                double beta = 0.01;
                String similarityFileName = "testing2.txt";
                double weight = 0.1;
                double threshold = 0.75;
                int filterSize = 20;

                for (int round = 1; round <= 5; round += 1) {
                        for (int num_topic = 60; num_topic <= 60; num_topic += 20) {
                                double alpha = 1.0 * 6 / num_topic;
                                RatioGPUDMM gsdmm = new RatioGPUDMM(doc_list, num_topic, num_iter, save_step, beta, alpha, threshold);
                                gsdmm.word2idFileName = "FFS.txt";
                                gsdmm.topWords = 100;

                                //here
                                gsdmm.filterSize = filterSize;
                                gsdmm.roundIndex = round;
                                gsdmm.similarityFileName = similarityFileName;
                                gsdmm.weight = weight;
                                gsdmm.initNewModel();
                                gsdmm.init_GSDMM();
                                gsdmm.run_iteration();
                                String flag = round+"round_"+num_topic + "topic_weight05_snippet" + "_filter20_iter1000_gpudmm";
                                flag = "snippetUpdateNow/" + flag;
                                gsdmm.saveModel(flag);
                        }
                }
        }

}
```

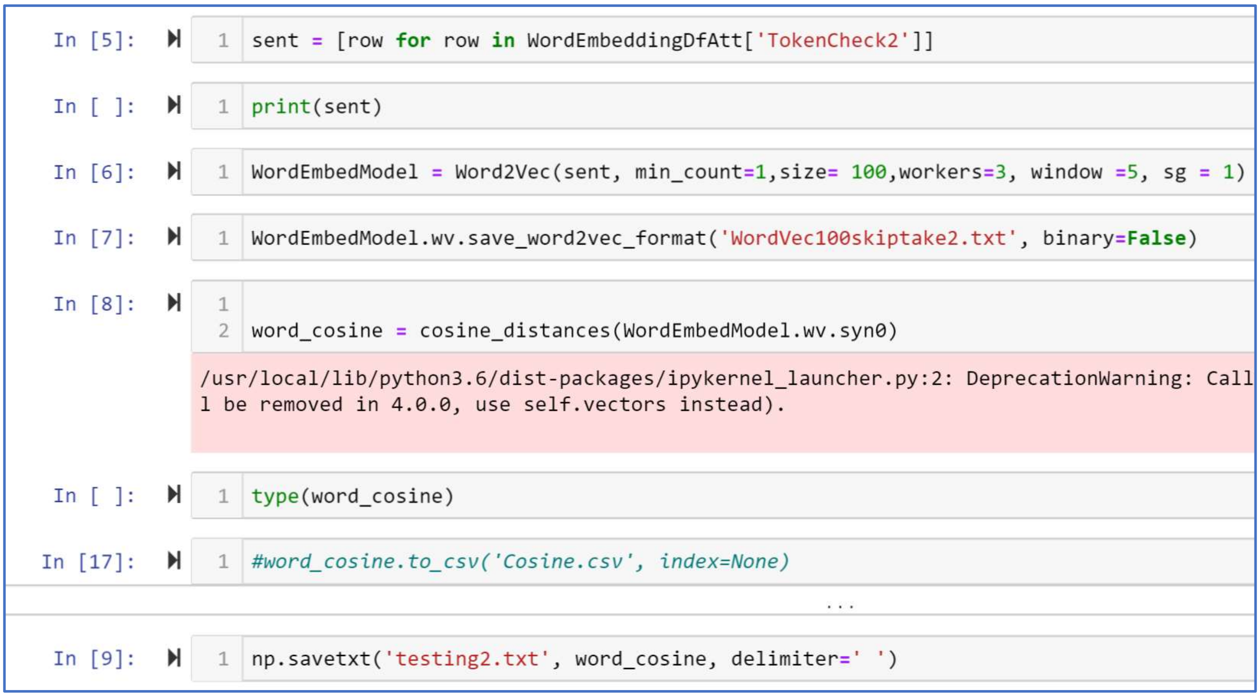
Figure 74: Specification of parameters for GPUDMM

[6] https://github.com/WHUIR/GPUDMM



```
In [2]:  # XXX if prep of corpus for GPUDMM formatting requirements
         xxx = pd.read_json('~/16JuneTopNoGreta5hundred.json', lines = True)

In [ ]:  xxx.sample(100)

In [37]: xxx.info()
...

In [ ]:  xxx['categories'] = pd.Series(['who|', 'what|', 'where|'].repeat(2))

In [ ]:  xxx['categories']

In [ ]:  xxx['categories'].repeat([2, 3, 4])

In [ ]:  xxx['docLen'].sample(100)

In [1]:  xxx['docLenx'] =  xxx['docLen'].astype(str) + '|'
...

In [ ]:  xxx['ID'] = xxx.index.astype(str)

In [ ]:  xxx['ID'].sample(100)

In [ ]:  xxx['docLenxTS'] = xxx['docLenx'].astype(str) + xxx['TokenString'].astype(str)

In [ ]:  xxx[['ID','docLenxTS']].to_csv('WHUCorp7.txt', header=None, index=None, sep="\t", mode='a')
```

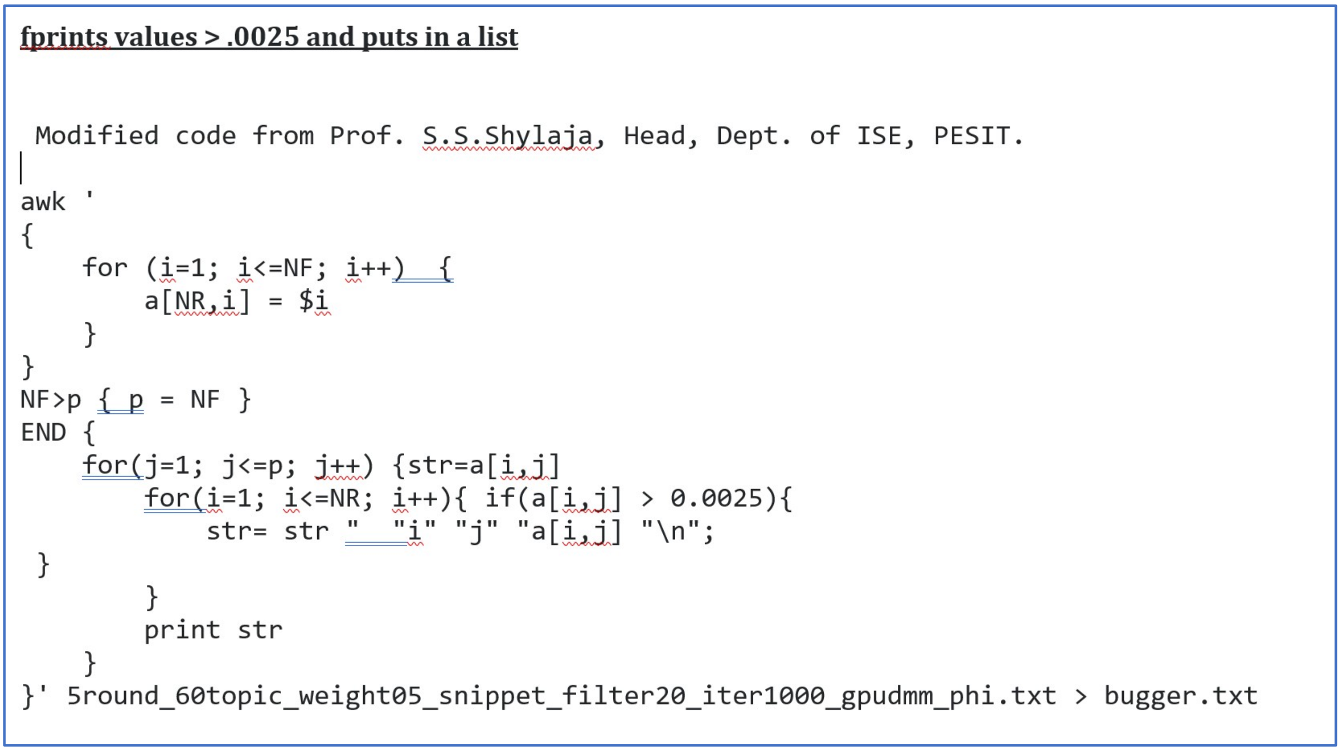
Figure 75: Corpus Format Preparation for GPUDMM



```
In [5]:  sent = [row for row in WordEmbeddingDfAtt['TokenCheck2']]

In [ ]:  print(sent)

In [6]:  WordEmbedModel = Word2Vec(sent, min_count=1,size= 100,workers=3, window =5, sg = 1)

In [7]:  WordEmbedModel.wv.save_word2vec_format('WordVec100skiptake2.txt', binary=False)

In [8]:  
         word_cosine = cosine_distances(WordEmbedModel.wv.syn0)

/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:2: DeprecationWarning: Call
l be removed in 4.0.0, use self.vectors instead).

In [ ]:  type(word_cosine)

In [17]: #word_cosine.to_csv('Cosine.csv', index=None)
...

In [9]:  np.savetxt('testing2.txt', word_cosine, delimiter=' ')
```

Figure 76: Word Embedding was converted to cosine similarities, as required by GPUDMM

Finding Top 20 weighted words per topic from phi matrix – the matrix was 29000 columns by 60 rows and had to be transformed to extract the words per topic:



fprints values > .0025 and puts in a list

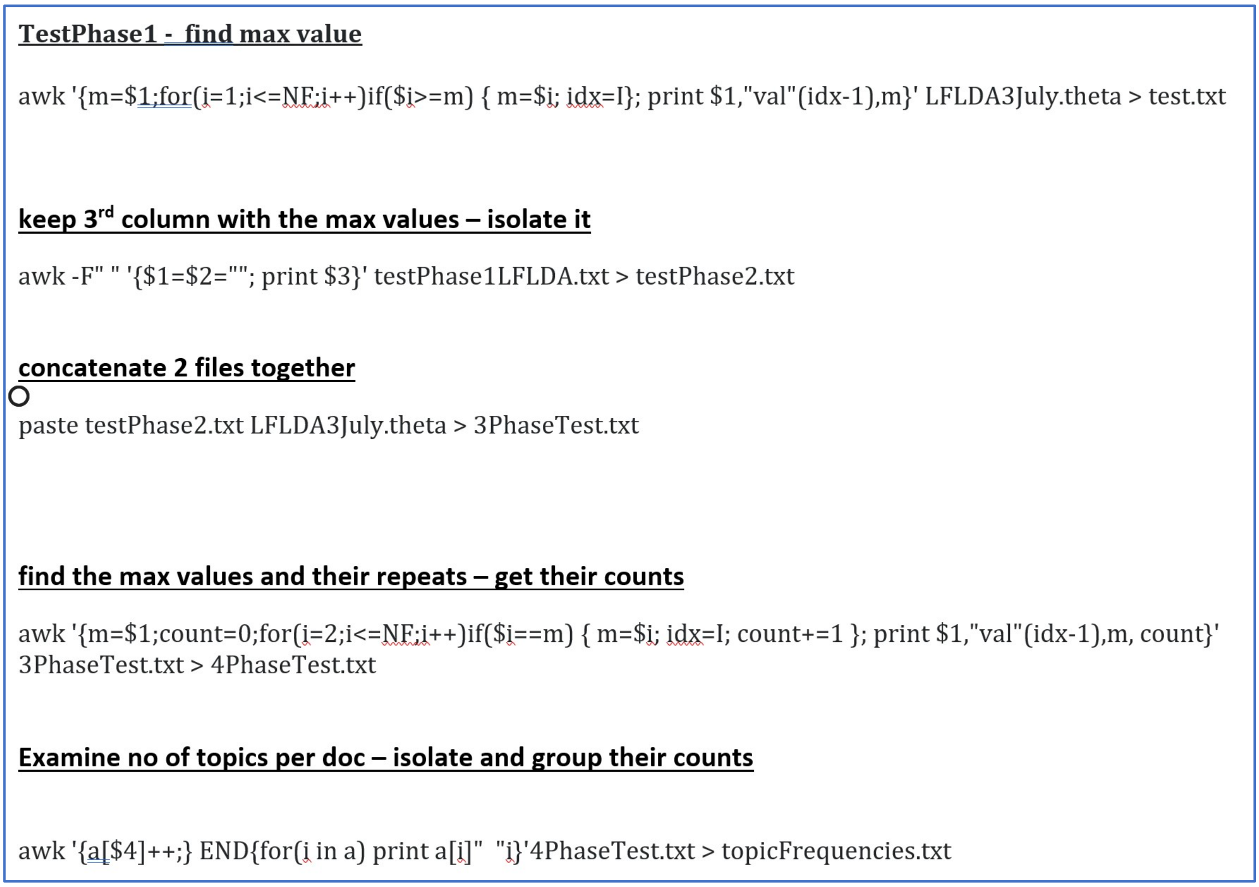
```
 Modified code from Prof. S.S.Shylaja, Head, Dept. of ISE, PESIT.

awk '
{
    for (i=1; i<=NF; i++)  {
        a[NR,i] = $i
    }
}
NF>p { p = NF }
END {
    for(j=1; j<=p; j++) {str=a[i,j]
        for(i=1; i<=NR; i++){ if(a[i,j] > 0.0025){
            str= str "   "i" "j" "a[i,j] "\n";
 }
        }
        print str
    }
}' 5round_60topic_weight05_snippet_filter20_iter1000_gpudmm_phi.txt > bugger.txt
```

Figure 77: Code to Invert word topic matrix to optimise file

```
sort -t$'\t' -k1,1n -k3'\t'3n phiVals1.txt > phiVals1.Sorted3.txt

sort -t$'\t' -k1,1n phiVals1.txt > phiVals1.Sorted3.txt
```

Figure 78: Sort words and weights by topic for output of topic model

4 Evaluation of Topic Models



TestPhase1 - find max value

```
awk '{m=$1;for(i=1;i<=NF;i++)if($i>=m) { m=$i; idx=I}; print $1,"val"(idx-1),m}' LFLDA3July.theta > test.txt
```

keep 3rd column with the max values – isolate it

```
awk -F" " '{$1=$2=""; print $3}' testPhase1LFLDA.txt > testPhase2.txt
```

concatenate 2 files together

```
paste testPhase2.txt LFLDA3July.theta > 3PhaseTest.txt
```

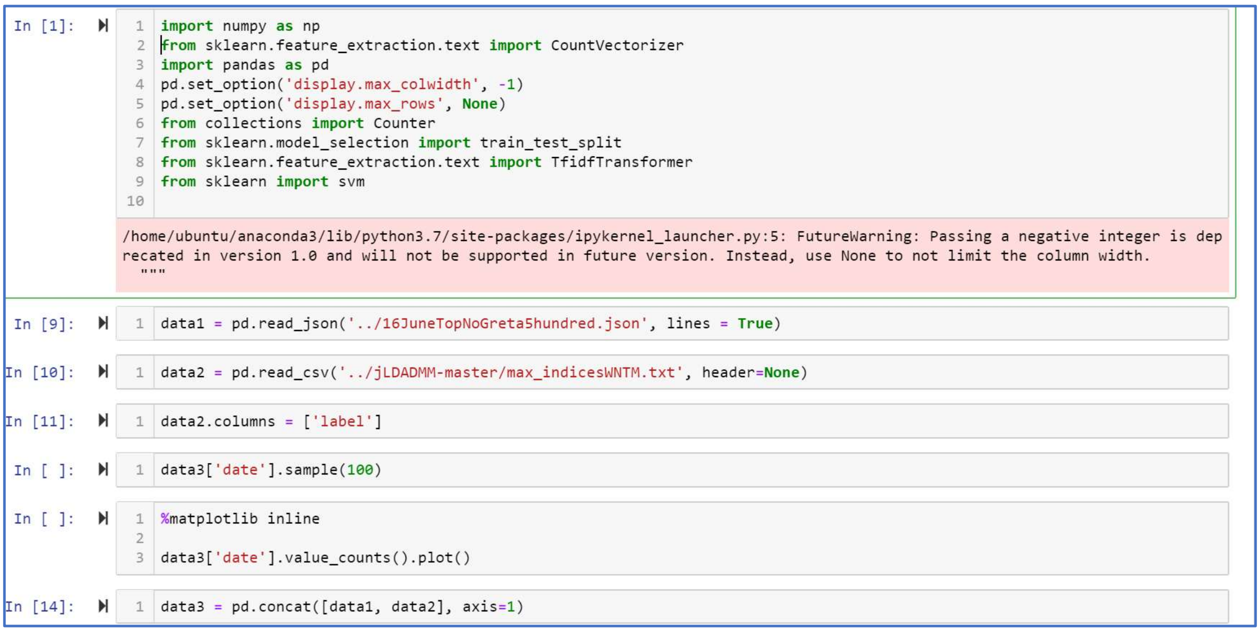
find the max values and their repeats – get their counts

```
awk '{m=$1;count=0;for(i=2;i<=NF;i++)if($i==m) { m=$i; idx=I; count+=1 }; print $1,"val"(idx-1),m, count}'
3PhaseTest.txt > 4PhaseTest.txt
```

Examine no of topics per doc – isolate and group their counts

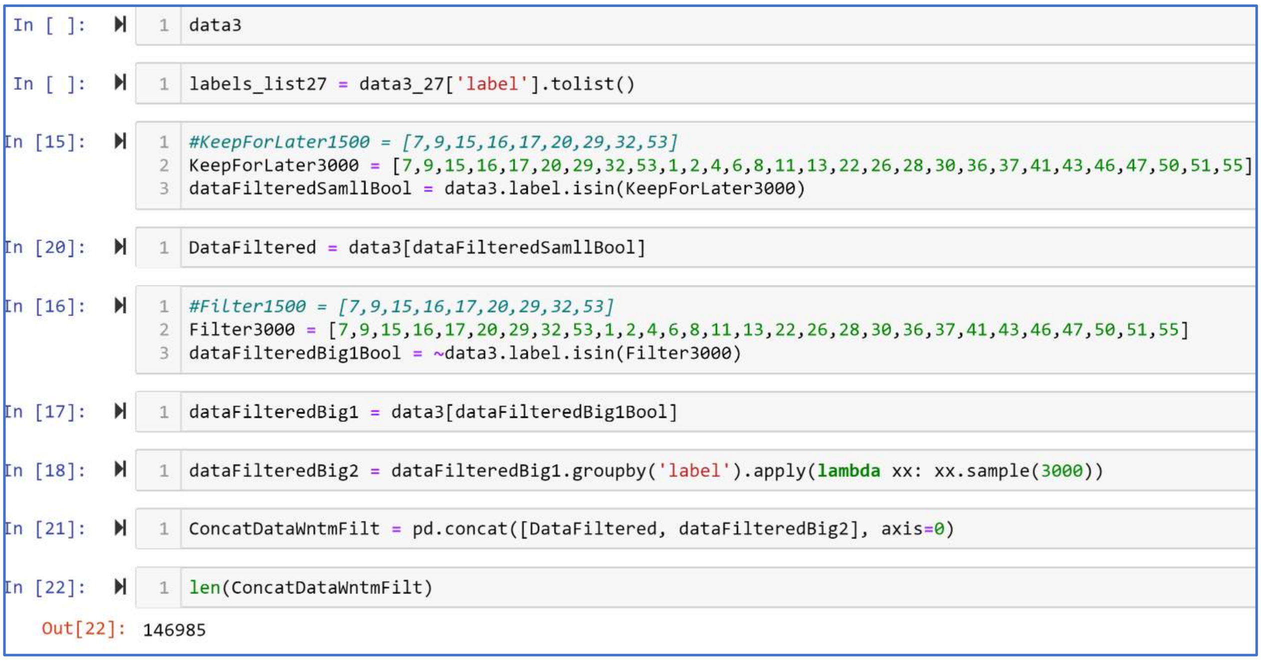
```
awk '{a[$4]++;} END{for(i in a) print a[i]"  "i}'4PhaseTest.txt > topicFrequencies.txt
```

Figure 79: Awk Commands to find document counts per Topic



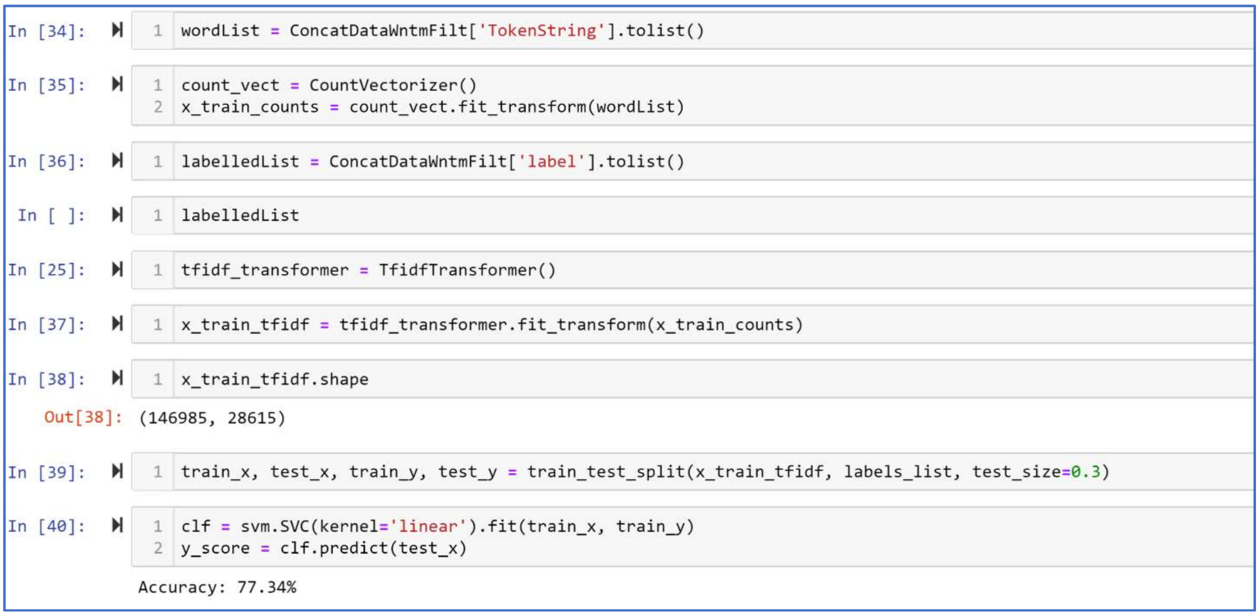
```
In [1]:
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
import pandas as pd
pd.set_option('display.max_colwidth', -1)
pd.set_option('display.max_rows', None)
from collections import Counter
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn import svm

/home/ubuntu/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:5: FutureWarning: Passing a negative integer is dep
recated in version 1.0 and will not be supported in future version. Instead, use None to not limit the column width.
  """

In [9]:
data1 = pd.read_json('../16JuneTopNoGreta5hundred.json', lines = True)

In [10]:
data2 = pd.read_csv('../jLDADMM-master/max_indicesWNTM.txt', header=None)

In [11]:
data2.columns = ['label']

In [ ]:
data3['date'].sample(100)

In [ ]:
%matplotlib inline

data3['date'].value_counts().plot()

In [14]:
data3 = pd.concat([data1, data2], axis=1)
```

Figure 80: Joining assigned document labels to corpus



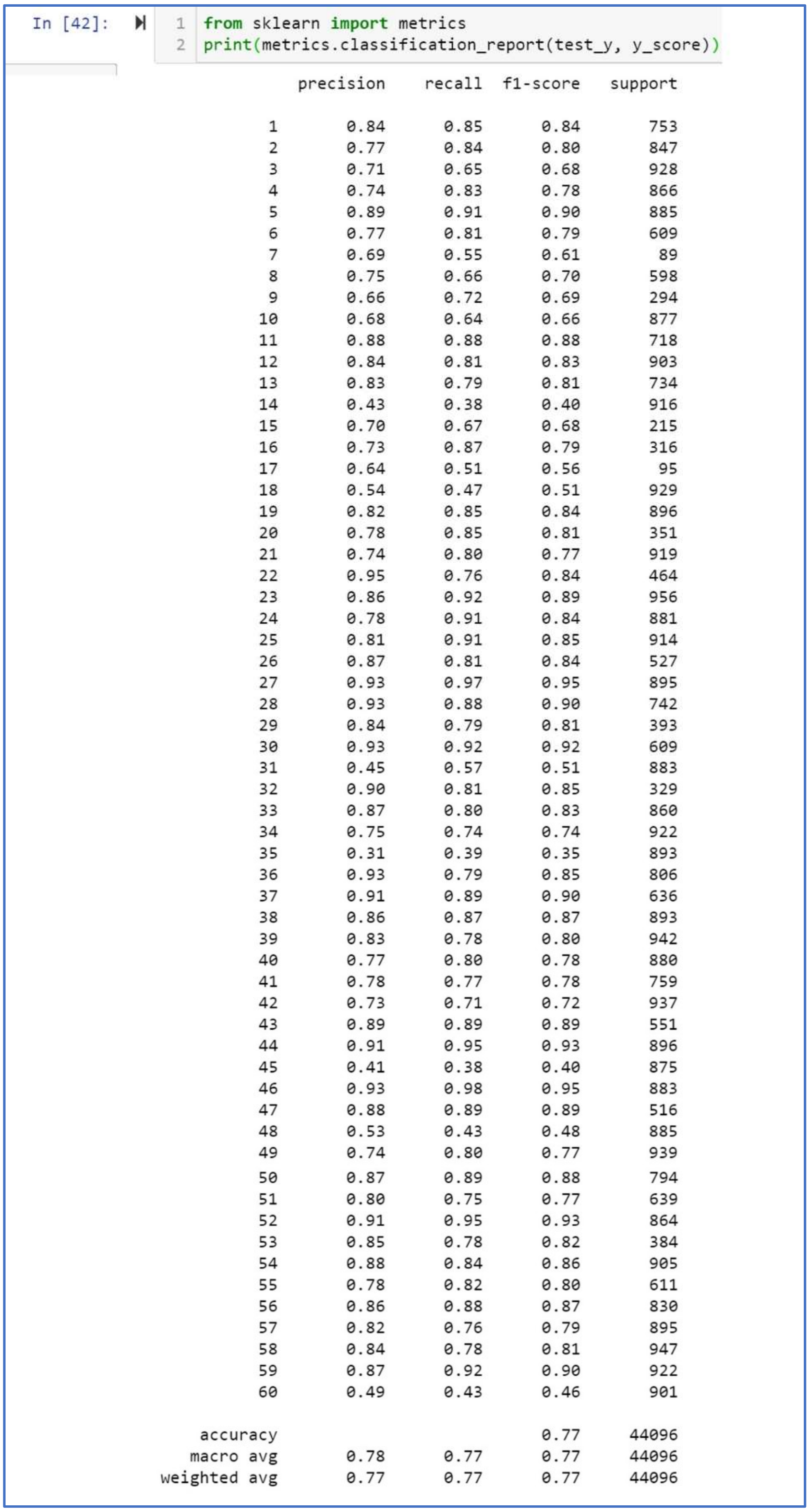
```
In [ ]:   data3

In [ ]:   labels_list27 = data3_27['label'].tolist()

In [15]:  #KeepForLater1500 = [7,9,15,16,17,20,29,32,53]
          KeepForLater3000 = [7,9,15,16,17,20,29,32,53,1,2,4,6,8,11,13,22,26,28,30,36,37,41,43,46,47,50,51,55]
          dataFilteredSamllBool = data3.label.isin(KeepForLater3000)

In [20]:  DataFiltered = data3[dataFilteredSamllBool]

In [16]:  #Filter1500 = [7,9,15,16,17,20,29,32,53]
          Filter3000 = [7,9,15,16,17,20,29,32,53,1,2,4,6,8,11,13,22,26,28,30,36,37,41,43,46,47,50,51,55]
          dataFilteredBig1Bool = ~data3.label.isin(Filter3000)

In [17]:  dataFilteredBig1 = data3[dataFilteredBig1Bool]

In [18]:  dataFilteredBig2 = dataFilteredBig1.groupby('label').apply(lambda xx: xx.sample(3000))

In [21]:  ConcatDataWntmFilt = pd.concat([DataFiltered, dataFilteredBig2], axis=0)

In [22]:  len(ConcatDataWntmFilt)
Out[22]:  146985
```

Figure 81: Addressing Class imbalance in WNTM dataset for SVC classification



```
In [34]:  wordList = ConcatDataWntmFilt['TokenString'].tolist()

In [35]:  count_vect = CountVectorizer()
          x_train_counts = count_vect.fit_transform(wordList)

In [36]:  labelledList = ConcatDataWntmFilt['label'].tolist()

In [ ]:   labelledList

In [25]:  tfidf_transformer = TfidfTransformer()

In [37]:  x_train_tfidf = tfidf_transformer.fit_transform(x_train_counts)

In [38]:  x_train_tfidf.shape
Out[38]:  (146985, 28615)

In [39]:  train_x, test_x, train_y, test_y = train_test_split(x_train_tfidf, labels_list, test_size=0.3)

In [40]:  clf = svm.SVC(kernel='linear').fit(train_x, train_y)
          y_score = clf.predict(test_x)

          Accuracy: 77.34%
```

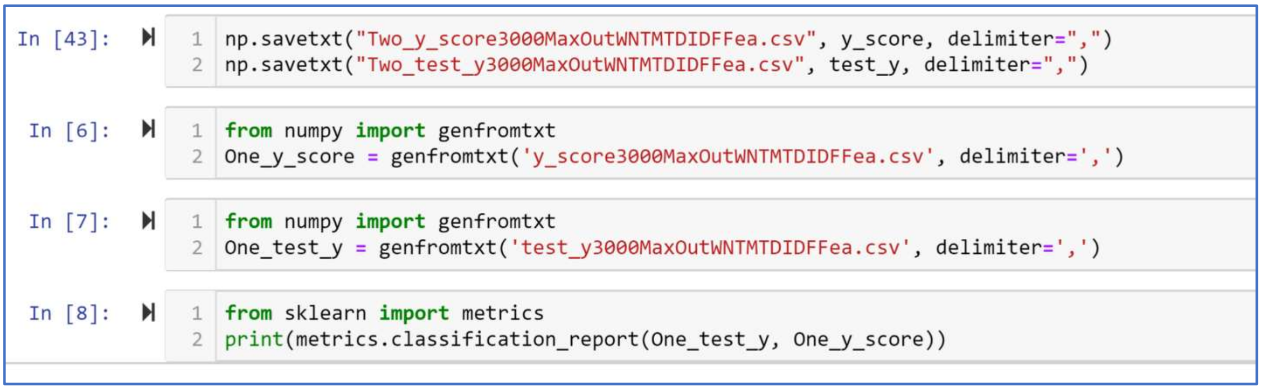
Figure 82: Classification model training



```
from sklearn import metrics
print(metrics.classification_report(test_y, y_score))
```

```
              precision    recall  f1-score   support

           1       0.84      0.85      0.84       753
           2       0.77      0.84      0.80       847
           3       0.71      0.65      0.68       928
           4       0.74      0.83      0.78       866
           5       0.89      0.91      0.90       885
           6       0.77      0.81      0.79       609
           7       0.69      0.55      0.61        89
           8       0.75      0.66      0.70       598
           9       0.66      0.72      0.69       294
          10       0.68      0.64      0.66       877
          11       0.88      0.88      0.88       718
          12       0.84      0.81      0.83       903
          13       0.83      0.79      0.81       734
          14       0.43      0.38      0.40       916
          15       0.70      0.67      0.68       215
          16       0.73      0.87      0.79       316
          17       0.64      0.51      0.56        95
          18       0.54      0.47      0.51       929
          19       0.82      0.85      0.84       896
          20       0.78      0.85      0.81       351
          21       0.74      0.80      0.77       919
          22       0.95      0.76      0.84       464
          23       0.86      0.92      0.89       956
          24       0.78      0.91      0.84       881
          25       0.81      0.91      0.85       914
          26       0.87      0.81      0.84       527
          27       0.93      0.97      0.95       895
          28       0.93      0.88      0.90       742
          29       0.84      0.79      0.81       393
          30       0.93      0.92      0.92       609
          31       0.45      0.57      0.51       883
          32       0.90      0.81      0.85       329
          33       0.87      0.80      0.83       860
          34       0.75      0.74      0.74       922
          35       0.31      0.39      0.35       893
          36       0.93      0.79      0.85       806
          37       0.91      0.89      0.90       636
          38       0.86      0.87      0.87       893
          39       0.83      0.78      0.80       942
          40       0.77      0.80      0.78       880
          41       0.78      0.77      0.78       759
          42       0.73      0.71      0.72       937
          43       0.89      0.89      0.89       551
          44       0.91      0.95      0.93       896
          45       0.41      0.38      0.40       875
          46       0.93      0.98      0.95       883
          47       0.88      0.89      0.89       516
          48       0.53      0.43      0.48       885
          49       0.74      0.80      0.77       939
          50       0.87      0.89      0.88       794
          51       0.80      0.75      0.77       639
          52       0.91      0.95      0.93       864
          53       0.85      0.78      0.82       384
          54       0.88      0.84      0.86       905
          55       0.78      0.82      0.80       611
          56       0.86      0.88      0.87       830
          57       0.82      0.76      0.79       895
          58       0.84      0.78      0.81       947
          59       0.87      0.92      0.90       922
          60       0.49      0.43      0.46       901

    accuracy                           0.77     44096
   macro avg       0.78      0.77      0.77     44096
weighted avg       0.77      0.77      0.77     44096
```

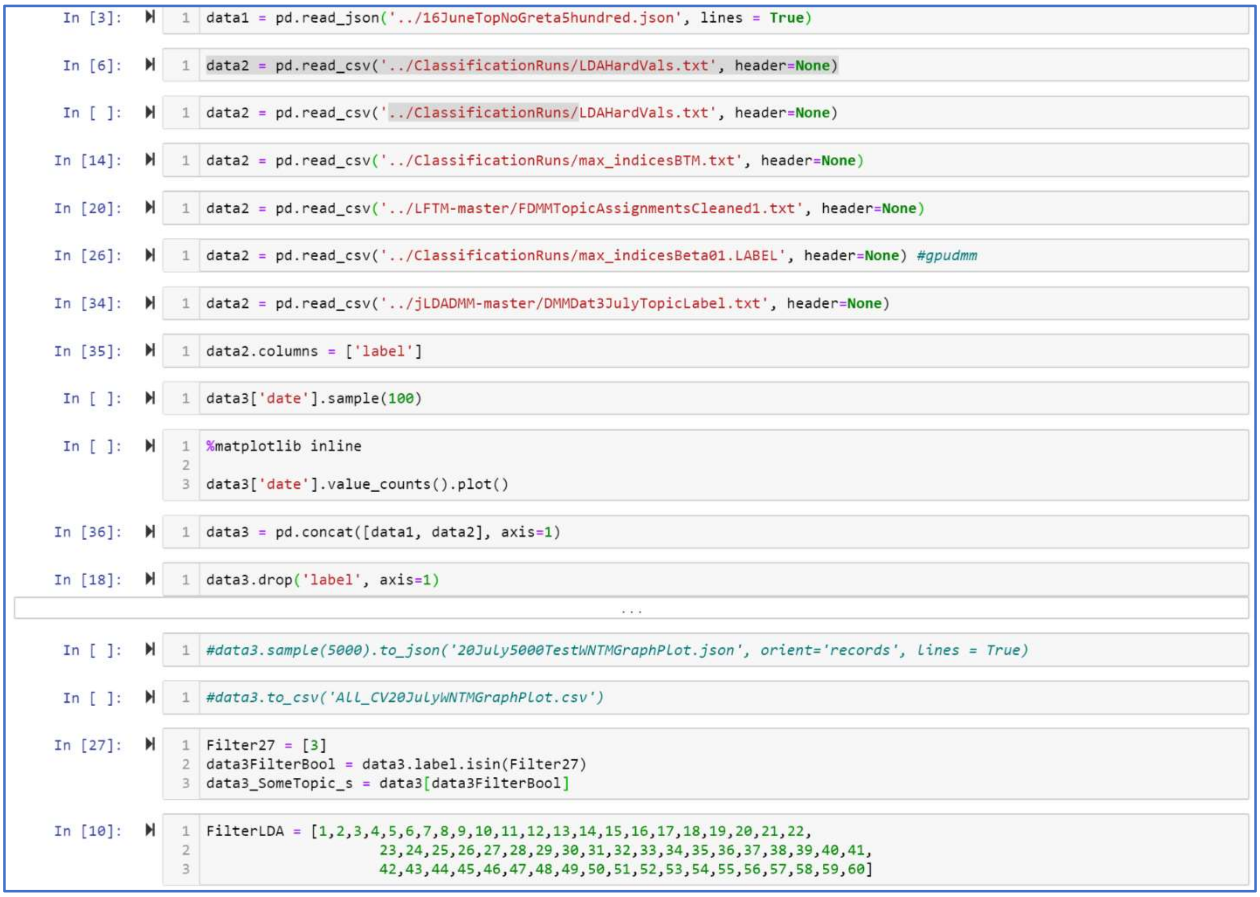
Figure 83: Classification recall results per topic



```
In [43]:  np.savetxt("Two_y_score3000MaxOutWNTMTDIDFFea.csv", y_score, delimiter=",")
           np.savetxt("Two_test_y3000MaxOutWNTMTDIDFFea.csv", test_y, delimiter=",")

In [6]:    from numpy import genfromtxt
           One_y_score = genfromtxt('y_score3000MaxOutWNTMTDIDFFea.csv', delimiter=',')

In [7]:    from numpy import genfromtxt
           One_test_y = genfromtxt('test_y3000MaxOutWNTMTDIDFFea.csv', delimiter=',')

In [8]:    from sklearn import metrics
           print(metrics.classification_report(One_test_y, One_y_score))
```

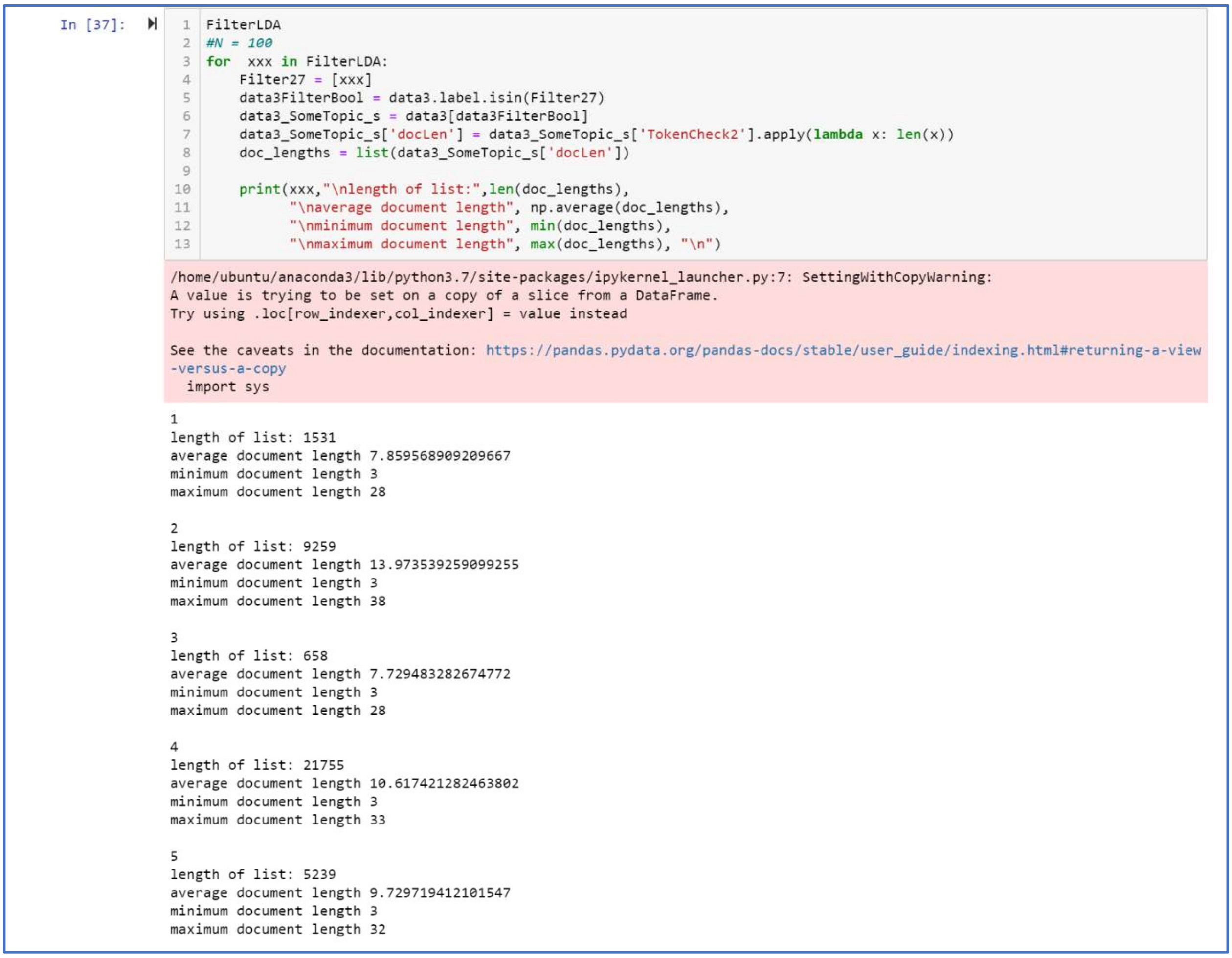
Figure 84: Saving recall result to file

It was decided to set the coherence window to 15 upon examination of average document length per topic. For all models, a few topics had average document length counts slightly above 15. To accommodate these topics in the coherence windows, the size of the context window was set to 15.



```
In [3]:   data1 = pd.read_json('../16JuneTopNoGreta5hundred.json', lines = True)

In [6]:   data2 = pd.read_csv('../ClassificationRuns/LDAHardVals.txt', header=None)

In [ ]:   data2 = pd.read_csv('../ClassificationRuns/LDAHardVals.txt', header=None)

In [14]:  data2 = pd.read_csv('../ClassificationRuns/max_indicesBTM.txt', header=None)

In [20]:  data2 = pd.read_csv('../LFTM-master/FDMMTopicAssignmentsCleaned1.txt', header=None)

In [26]:  data2 = pd.read_csv('../ClassificationRuns/max_indicesBeta01.LABEL', header=None) #gpudmm

In [34]:  data2 = pd.read_csv('../jLDADMM-master/DMMDat3JulyTopicLabel.txt', header=None)

In [35]:  data2.columns = ['label']

In [ ]:   data3['date'].sample(100)

In [ ]:   %matplotlib inline

          data3['date'].value_counts().plot()

In [36]:  data3 = pd.concat([data1, data2], axis=1)

In [18]:  data3.drop('label', axis=1)
...

In [ ]:   #data3.sample(5000).to_json('20July5000TestWNTMGraphPlot.json', orient='records', lines = True)

In [ ]:   #data3.to_csv('All_CV20JulyWNTMGraphPlot.csv')

In [27]:  Filter27 = [3]
          data3FilterBool = data3.label.isin(Filter27)
          data3_SomeTopic_s = data3[data3FilterBool]

In [10]:  FilterLDA = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
                      23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,
                      42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60]
```

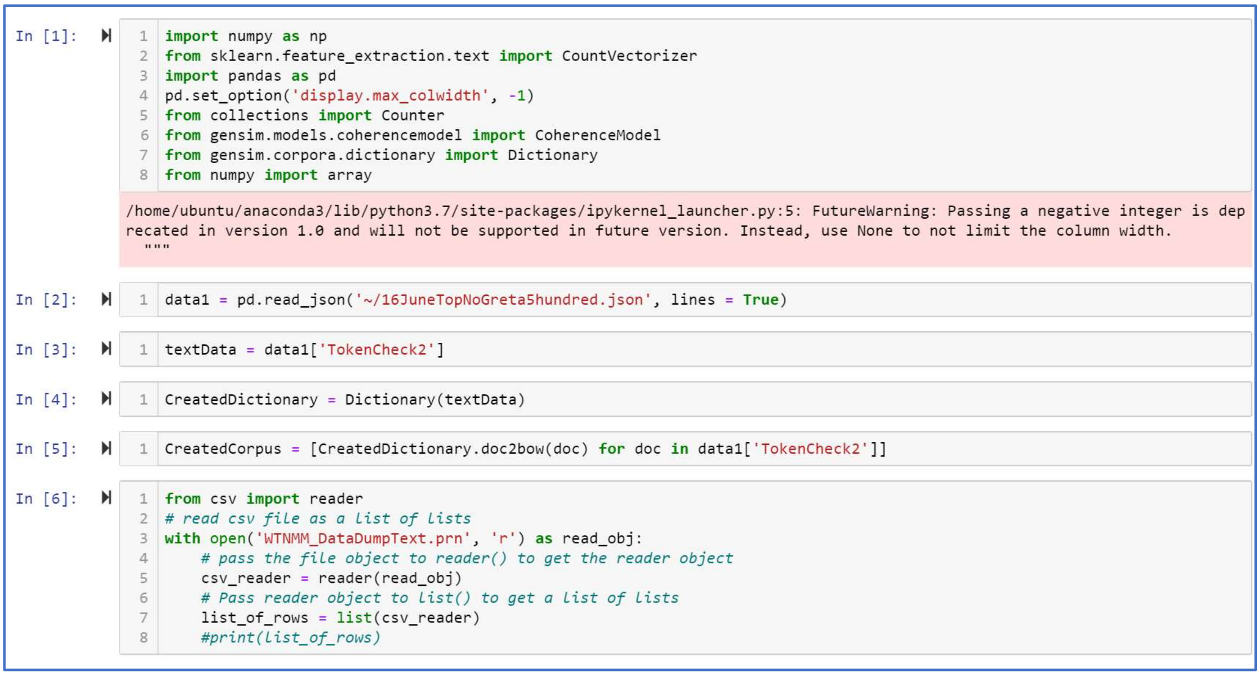
Figure 85: Loading of consecutive label allocation per topic model method



```
In [37]:
FilterLDA
#N = 100
for  xxx in FilterLDA:
    Filter27 = [xxx]
    data3FilterBool = data3.label.isin(Filter27)
    data3_SomeTopic_s = data3[data3FilterBool]
    data3_SomeTopic_s['docLen'] = data3_SomeTopic_s['TokenCheck2'].apply(lambda x: len(x))
    doc_lengths = list(data3_SomeTopic_s['docLen'])

    print(xxx,"\nlength of list:",len(doc_lengths),
          "\naverage document length", np.average(doc_lengths),
          "\nminimum document length", min(doc_lengths),
          "\nmaximum document length", max(doc_lengths), "\n")
```

```
/home/ubuntu/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
-versus-a-copy
  import sys

1
length of list: 1531
average document length 7.859568909209667
minimum document length 3
maximum document length 28

2
length of list: 9259
average document length 13.973539259099255
minimum document length 3
maximum document length 38

3
length of list: 658
average document length 7.729483282674772
minimum document length 3
maximum document length 28

4
length of list: 21755
average document length 10.617421282463802
minimum document length 3
maximum document length 33

5
length of list: 5239
average document length 9.729719412101547
minimum document length 3
maximum document length 32
```

Figure 86: Average document length per topic for each method

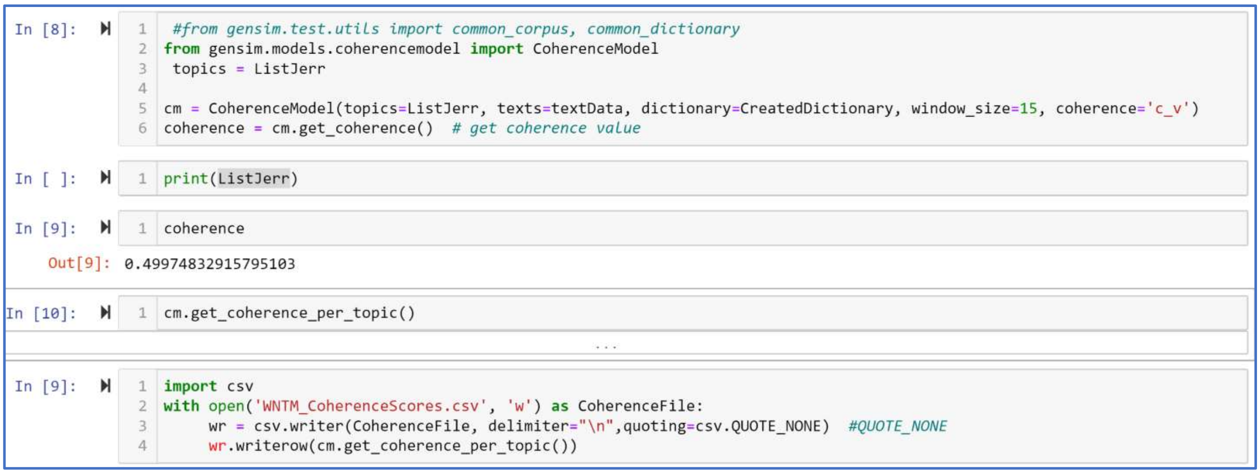


```
In [1]:
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
import pandas as pd
pd.set_option('display.max_colwidth', -1)
from collections import Counter
from gensim.models.coherencemodel import CoherenceModel
from gensim.corpora.dictionary import Dictionary
from numpy import array
```

```
/home/ubuntu/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:5: FutureWarning: Passing a negative integer is dep
recated in version 1.0 and will not be supported in future version. Instead, use None to not limit the column width.
  """
```

```
In [2]:
data1 = pd.read_json('~/16JuneTopNoGreta5hundred.json', lines = True)

In [3]:
textData = data1['TokenCheck2']

In [4]:
CreatedDictionary = Dictionary(textData)

In [5]:
CreatedCorpus = [CreatedDictionary.doc2bow(doc) for doc in data1['TokenCheck2']]

In [6]:
from csv import reader
# read csv file as a list of lists
with open('WTNMM_DataDumpText.prn', 'r') as read_obj:
    # pass the file object to reader() to get the reader object
    csv_reader = reader(read_obj)
    # Pass reader object to list() to get a list of lists
    list_of_rows = list(csv_reader)
    #print(list_of_rows)
```

Figure 87: Data preparation for coherence testing



```
In [8]:
 #from gensim.test.utils import common_corpus, common_dictionary
from gensim.models.coherencemodel import CoherenceModel
 topics = ListJerr

cm = CoherenceModel(topics=ListJerr, texts=textData, dictionary=CreatedDictionary, window_size=15, coherence='c_v')
coherence = cm.get_coherence()  # get coherence value
```

```
In [ ]:
print(ListJerr)
```

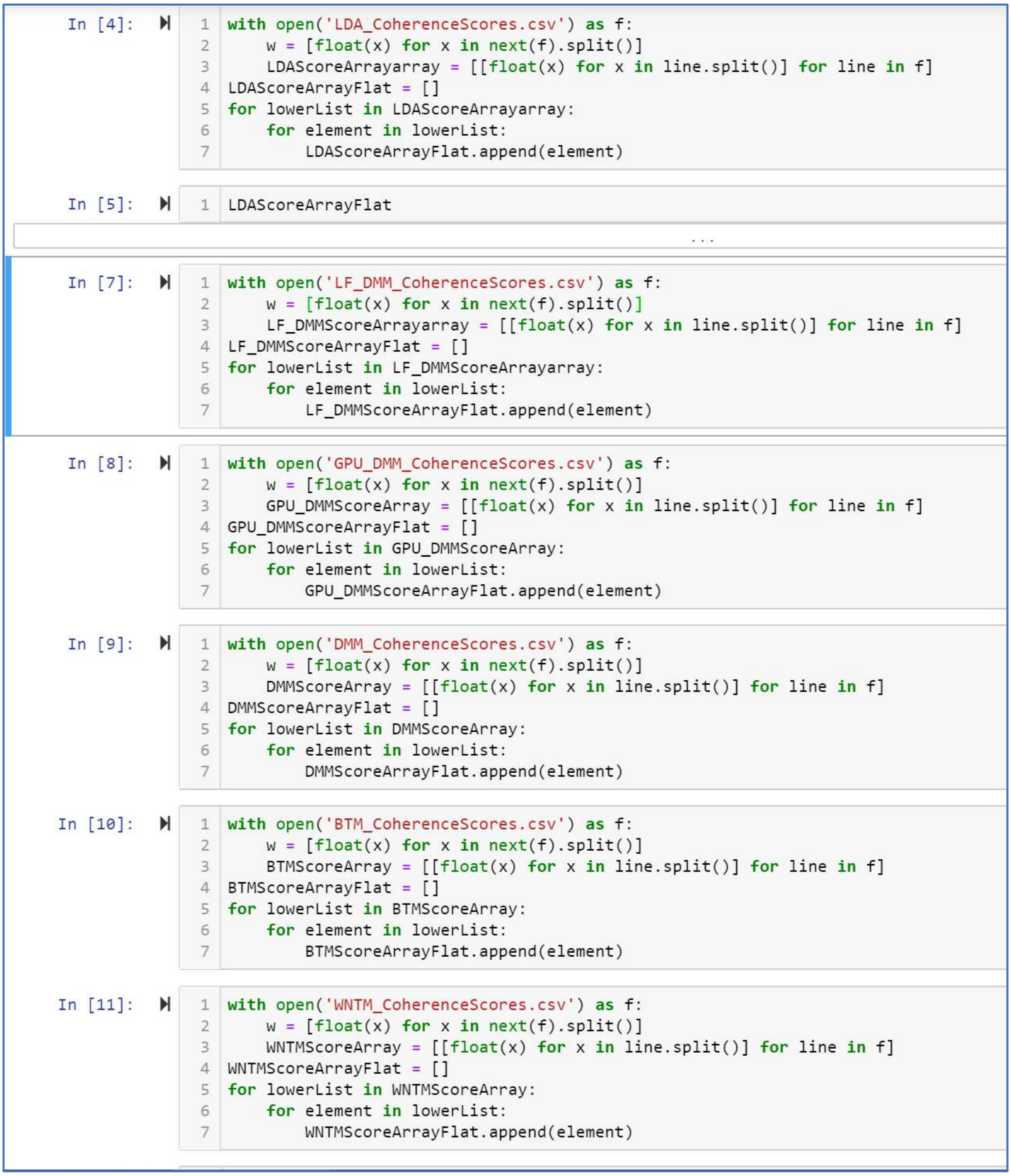
```
In [9]:
coherence
```

```
Out[9]: 0.49974832915795103
```

```
In [10]:
cm.get_coherence_per_topic()
```

```
In [9]:
import csv
with open('WNTM_CoherenceScores.csv', 'w') as CoherenceFile:
     wr = csv.writer(CoherenceFile, delimiter="\n",quoting=csv.QUOTE_NONE)  #QUOTE_NONE
     wr.writerow(cm.get_coherence_per_topic())
```

Figure 88: Meta-weighted average coherence for WNTM topics



In [4]:
```
with open('LDA_CoherenceScores.csv') as f:
    w = [float(x) for x in next(f).split()]
    LDAScoreArrayarray = [[float(x) for x in line.split()] for line in f]
LDAScoreArrayFlat = []
for lowerList in LDAScoreArrayarray:
    for element in lowerList:
        LDAScoreArrayFlat.append(element)
```

In [5]:
```
LDAScoreArrayFlat
```

...

In [7]:
```
with open('LF_DMM_CoherenceScores.csv') as f:
    w = [float(x) for x in next(f).split()]
    LF_DMMScoreArrayarray = [[float(x) for x in line.split()] for line in f]
LF_DMMScoreArrayFlat = []
for lowerList in LF_DMMScoreArrayarray:
    for element in lowerList:
        LF_DMMScoreArrayFlat.append(element)
```

In [8]:
```
with open('GPU_DMM_CoherenceScores.csv') as f:
    w = [float(x) for x in next(f).split()]
    GPU_DMMScoreArray = [[float(x) for x in line.split()] for line in f]
GPU_DMMScoreArrayFlat = []
for lowerList in GPU_DMMScoreArray:
    for element in lowerList:
        GPU_DMMScoreArrayFlat.append(element)
```

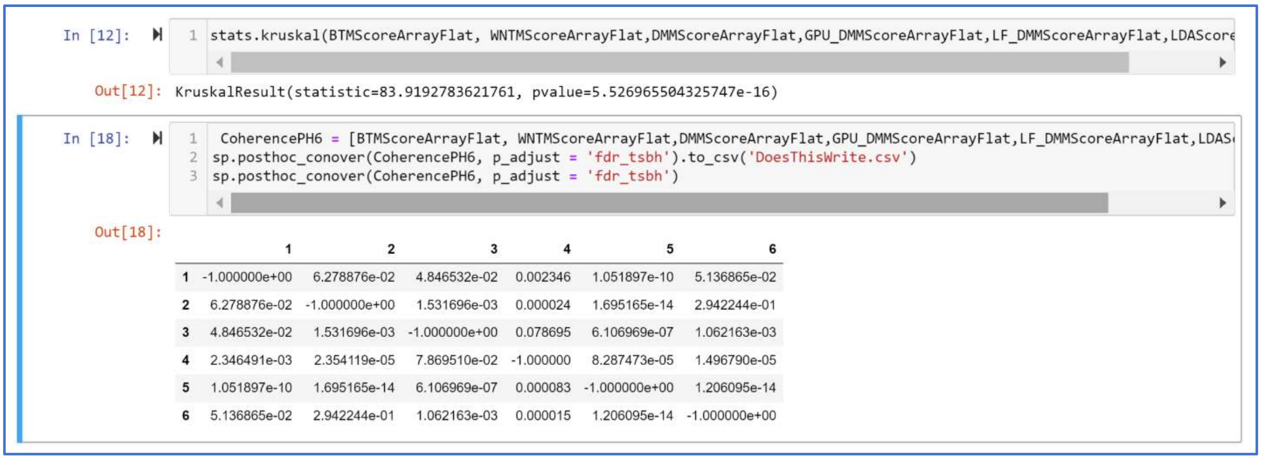
In [9]:
```
with open('DMM_CoherenceScores.csv') as f:
    w = [float(x) for x in next(f).split()]
    DMMScoreArray = [[float(x) for x in line.split()] for line in f]
DMMScoreArrayFlat = []
for lowerList in DMMScoreArray:
    for element in lowerList:
        DMMScoreArrayFlat.append(element)
```

In [10]:
```
with open('BTM_CoherenceScores.csv') as f:
    w = [float(x) for x in next(f).split()]
    BTMScoreArray = [[float(x) for x in line.split()] for line in f]
BTMScoreArrayFlat = []
for lowerList in BTMScoreArray:
    for element in lowerList:
        BTMScoreArrayFlat.append(element)
```

In [11]:
```
with open('WNTM_CoherenceScores.csv') as f:
    w = [float(x) for x in next(f).split()]
    WNTMScoreArray = [[float(x) for x in line.split()] for line in f]
WNTMScoreArrayFlat = []
for lowerList in WNTMScoreArray:
    for element in lowerList:
        WNTMScoreArrayFlat.append(element)
```

Figure 89: Recall based on Topic Proportions (extracted plot of documents < 7000)

Loading of saved coherence score distributions per topic model method into Lists for post-hoc analysis

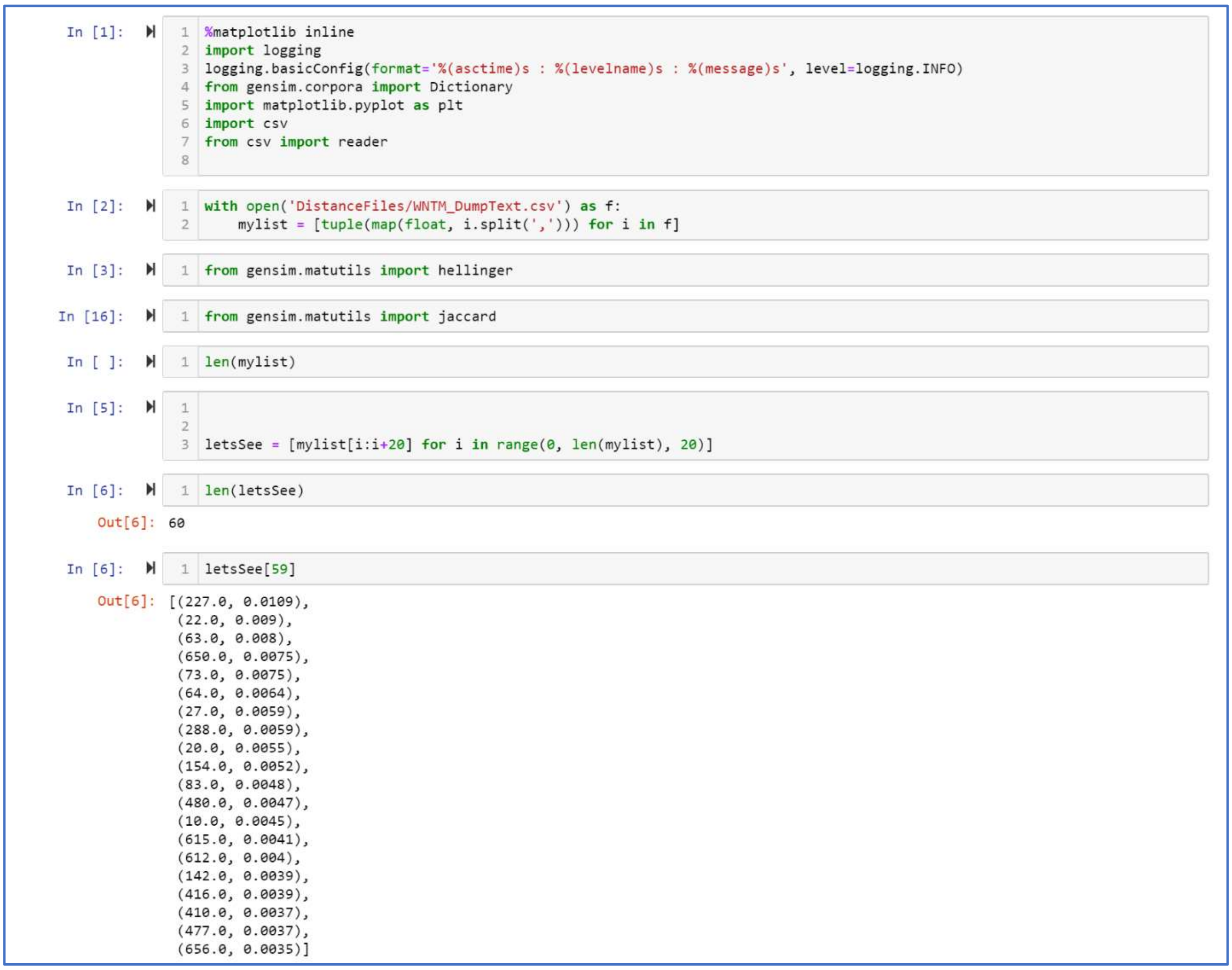


```
In [12]:
stats.kruskal(BTMScoreArrayFlat, WNTMScoreArrayFlat,DMMScoreArrayFlat,GPU_DMMScoreArrayFlat,LF_DMMScoreArrayFlat,LDAScore

Out[12]: KruskalResult(statistic=83.9192783621761, pvalue=5.526965504325747e-16)
```

```
In [18]:
 CoherencePH6 = [BTMScoreArrayFlat, WNTMScoreArrayFlat,DMMScoreArrayFlat,GPU_DMMScoreArrayFlat,LF_DMMScoreArrayFlat,LDAS
sp.posthoc_conover(CoherencePH6, p_adjust = 'fdr_tsbh').to_csv('DoesThisWrite.csv')
sp.posthoc_conover(CoherencePH6, p_adjust = 'fdr_tsbh')

Out[18]:
```

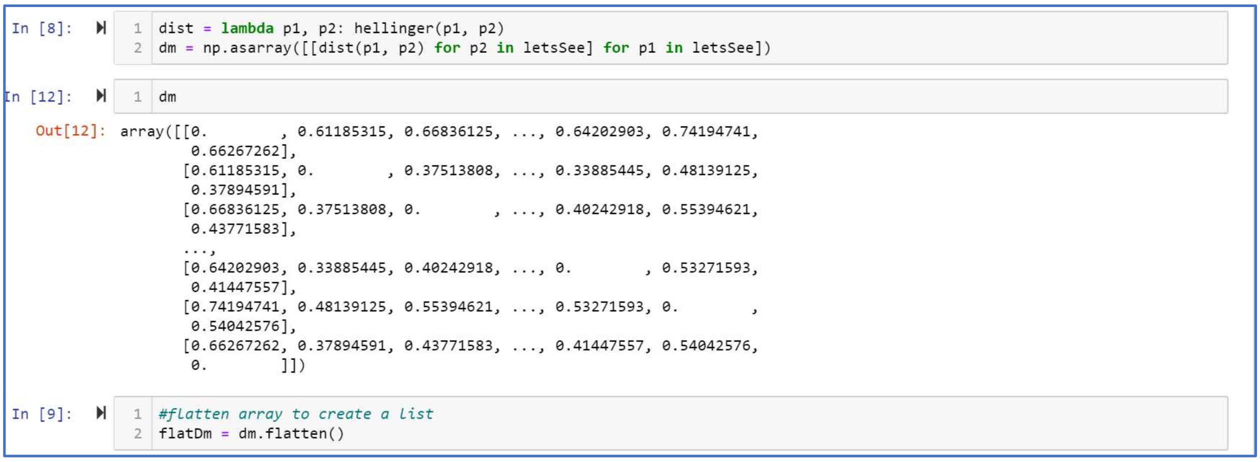
	1	2	3	4	5	6
1	-1.000000e+00	6.278876e-02	4.846532e-02	0.002346	1.051897e-10	5.136865e-02
2	6.278876e-02	-1.000000e+00	1.531696e-03	0.000024	1.695165e-14	2.942244e-01
3	4.846532e-02	1.531696e-03	-1.000000e+00	0.078695	6.106969e-07	1.062163e-03
4	2.346491e-03	2.354119e-05	7.869510e-02	-1.000000	8.287473e-05	1.496790e-05
5	1.051897e-10	1.695165e-14	6.106969e-07	0.000083	-1.000000e+00	1.206095e-14
6	5.136865e-02	2.942244e-01	1.062163e-03	0.000015	1.206095e-14	-1.000000e+00

Figure 90: Post-hoc testing on coherence scores



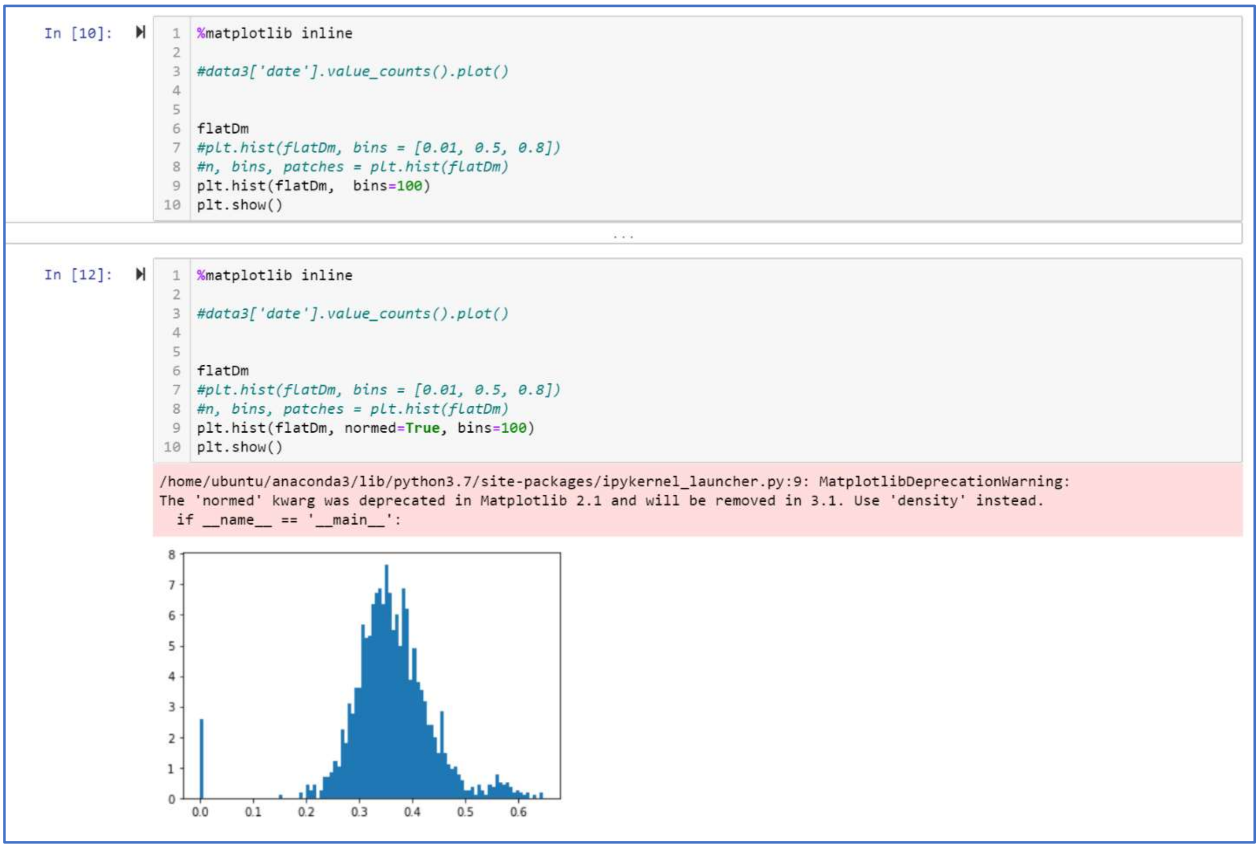
```
In [1]:
%matplotlib inline
import logging
logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.INFO)
from gensim.corpora import Dictionary
import matplotlib.pyplot as plt
import csv
from csv import reader


In [2]:
with open('DistanceFiles/WNTM_DumpText.csv') as f:
    mylist = [tuple(map(float, i.split(','))) for i in f]

In [3]:
from gensim.matutils import hellinger

In [16]:
from gensim.matutils import jaccard

In [ ]:
len(mylist)

In [5]:


letsSee = [mylist[i:i+20] for i in range(0, len(mylist), 20)]

In [6]:
len(letsSee)

Out[6]: 60

In [6]:
letsSee[59]

Out[6]: [(227.0, 0.0109),
 (22.0, 0.009),
 (63.0, 0.008),
 (650.0, 0.0075),
 (73.0, 0.0075),
 (64.0, 0.0064),
 (27.0, 0.0059),
 (288.0, 0.0059),
 (20.0, 0.0055),
 (154.0, 0.0052),
 (83.0, 0.0048),
 (480.0, 0.0047),
 (10.0, 0.0045),
 (615.0, 0.0041),
 (612.0, 0.004),
 (142.0, 0.0039),
 (416.0, 0.0039),
 (410.0, 0.0037),
 (477.0, 0.0037),
 (656.0, 0.0035)]
```

Figure 91: Generation of distances amongst topics with their word weightings



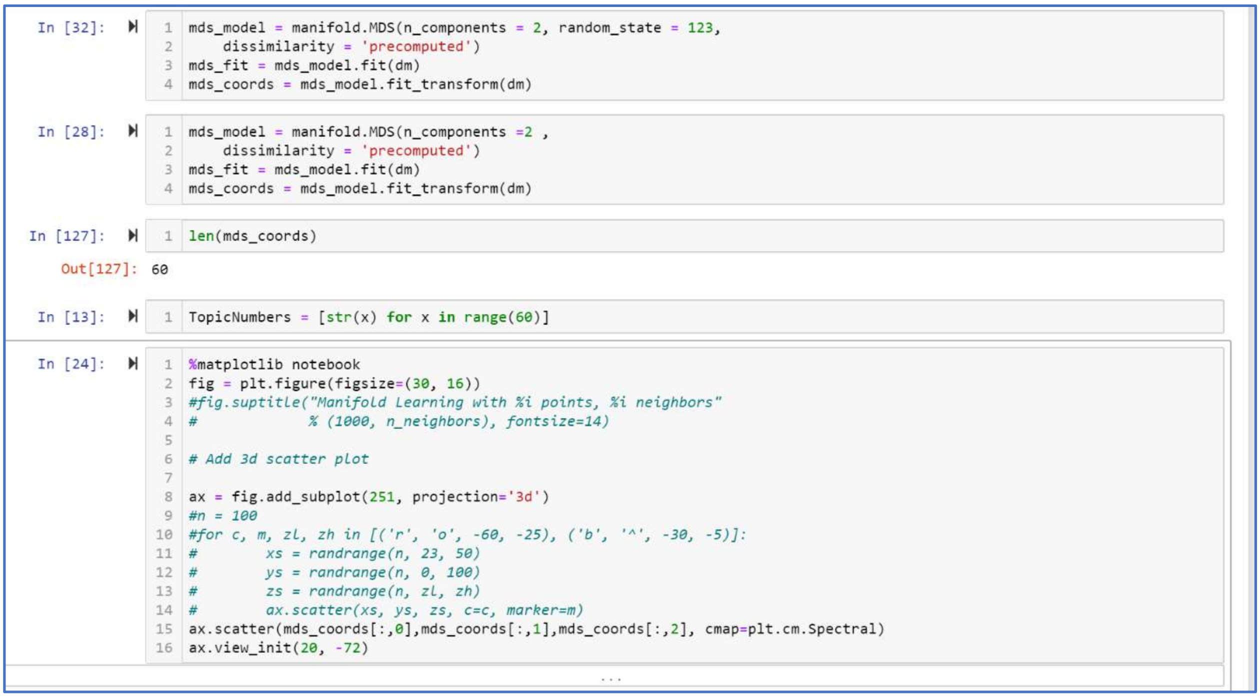
```
In [8]:  dist = lambda p1, p2: hellinger(p1, p2)
          dm = np.asarray([[dist(p1, p2) for p2 in letsSee] for p1 in letsSee])

In [12]:  dm

Out[12]: array([[0.        , 0.61185315, 0.66836125, ..., 0.64202903, 0.74194741,
                  0.66267262],
                 [0.61185315, 0.        , 0.37513808, ..., 0.33885445, 0.48139125,
                  0.37894591],
                 [0.66836125, 0.37513808, 0.        , ..., 0.40242918, 0.55394621,
                  0.43771583],
                 ...,
                 [0.64202903, 0.33885445, 0.40242918, ..., 0.        , 0.53271593,
                  0.41447557],
                 [0.74194741, 0.48139125, 0.55394621, ..., 0.53271593, 0.        ,
                  0.54042576],
                 [0.66267262, 0.37894591, 0.43771583, ..., 0.41447557, 0.54042576,
                  0.        ]])

In [9]:  #flatten array to create a list
         flatDm = dm.flatten()
```

Figure 92: Array flattening in preparation for Multi-dimensional Scaling



```
In [10]:  %matplotlib inline

          #data3['date'].value_counts().plot()


          flatDm
          #plt.hist(flatDm, bins = [0.01, 0.5, 0.8])
          #n, bins, patches = plt.hist(flatDm)
          plt.hist(flatDm,  bins=100)
          plt.show()

In [12]:  %matplotlib inline

          #data3['date'].value_counts().plot()


          flatDm
          #plt.hist(flatDm, bins = [0.01, 0.5, 0.8])
          #n, bins, patches = plt.hist(flatDm)
          plt.hist(flatDm, normed=True, bins=100)
          plt.show()

/home/ubuntu/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:9: MatplotlibDeprecationWarning:
The 'normed' kwarg was deprecated in Matplotlib 2.1 and will be removed in 3.1. Use 'density' instead.
  if __name__ == '__main__':
```

Figure 93: Distribution of distance similarities



```
In [32]:
mds_model = manifold.MDS(n_components = 2, random_state = 123,
    dissimilarity = 'precomputed')
mds_fit = mds_model.fit(dm)
mds_coords = mds_model.fit_transform(dm)
```

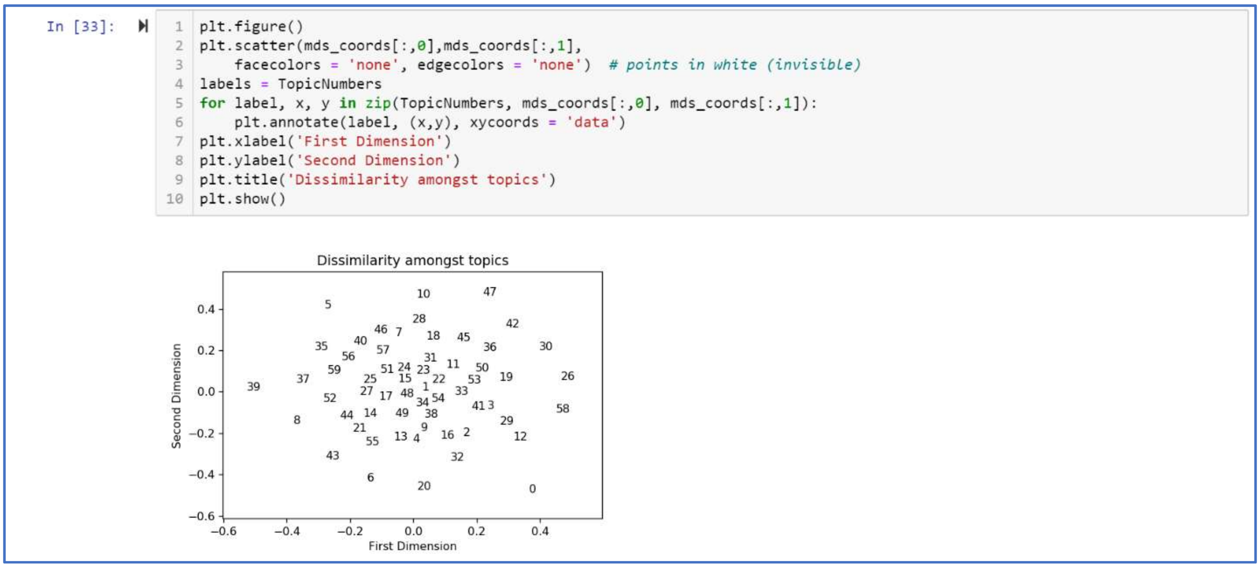
```
In [28]:
mds_model = manifold.MDS(n_components =2 ,
    dissimilarity = 'precomputed')
mds_fit = mds_model.fit(dm)
mds_coords = mds_model.fit_transform(dm)
```

```
In [127]:
len(mds_coords)
Out[127]: 60
```

```
In [13]:
TopicNumbers = [str(x) for x in range(60)]
```

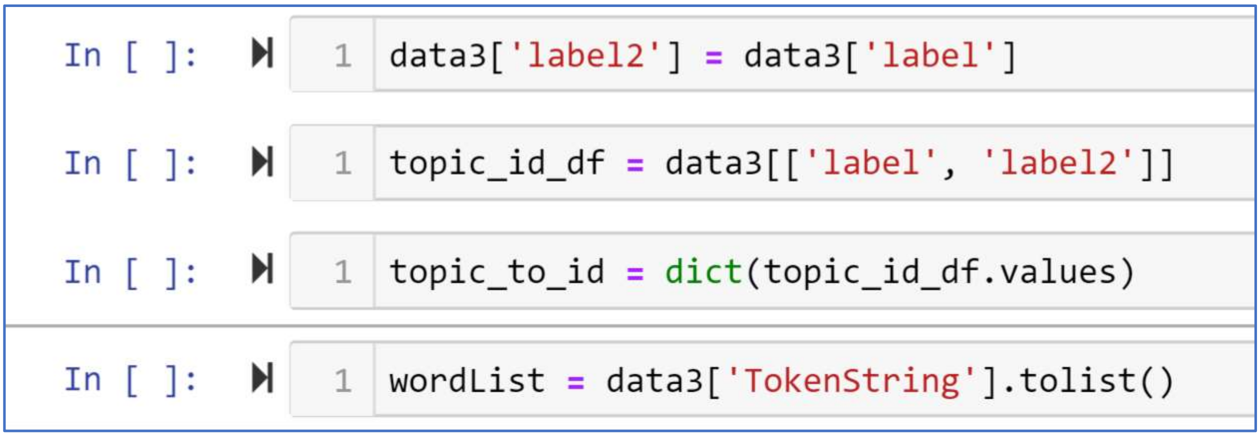
```
In [24]:
%matplotlib notebook
fig = plt.figure(figsize=(30, 16))
#fig.suptitle("Manifold Learning with %i points, %i neighbors"
#             % (1000, n_neighbors), fontsize=14)

# Add 3d scatter plot

ax = fig.add_subplot(251, projection='3d')
#n = 100
#for c, m, zl, zh in [('r', 'o', -60, -25), ('b', '^', -30, -5)]:
#       xs = randrange(n, 23, 50)
#       ys = randrange(n, 0, 100)
#       zs = randrange(n, zl, zh)
#       ax.scatter(xs, ys, zs, c=c, marker=m)
ax.scatter(mds_coords[:,0],mds_coords[:,1],mds_coords[:,2], cmap=plt.cm.Spectral)
ax.view_init(20, -72)
```

Figure 94: Multidimensional Scaling specification



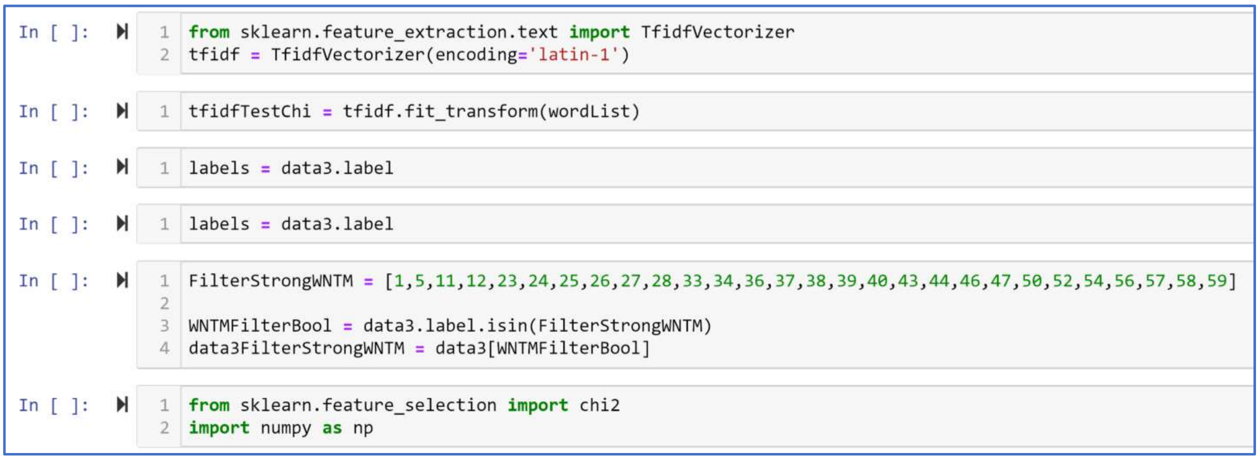
```
In [33]:
plt.figure()
plt.scatter(mds_coords[:,0],mds_coords[:,1],
    facecolors = 'none', edgecolors = 'none')  # points in white (invisible)
labels = TopicNumbers
for label, x, y in zip(TopicNumbers, mds_coords[:,0], mds_coords[:,1]):
    plt.annotate(label, (x,y), xycoords = 'data')
plt.xlabel('First Dimension')
plt.ylabel('Second Dimension')
plt.title('Dissimilarity amongst topics')
plt.show()
```

Figure 95: Multidimensional scaling plot projected onto two dimensions



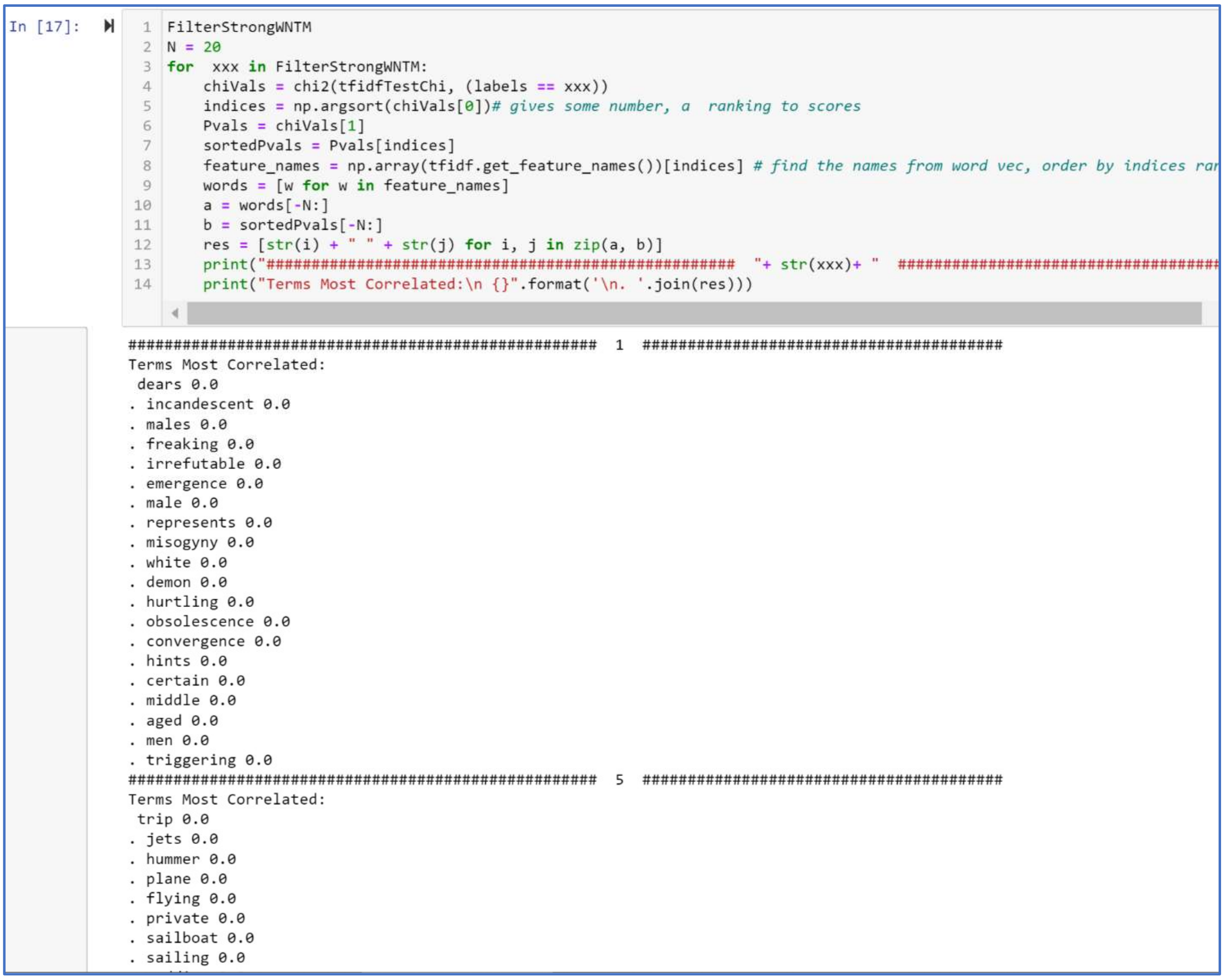
```
In [ ]: data3['label2'] = data3['label']

In [ ]: topic_id_df = data3[['label', 'label2']]

In [ ]: topic_to_id = dict(topic_id_df.values)

In [ ]: wordList = data3['TokenString'].tolist()
```

Figure 96: Creation of Word Features to be used in Chi test



```
In [ ]: from sklearn.feature_extraction.text import TfidfVectorizer
        tfidf = TfidfVectorizer(encoding='latin-1')

In [ ]: tfidfTestChi = tfidf.fit_transform(wordList)

In [ ]: labels = data3.label

In [ ]: labels = data3.label

In [ ]: FilterStrongWNTM = [1,5,11,12,23,24,25,26,27,28,33,34,36,37,38,39,40,43,44,46,47,50,52,54,56,57,58,59]

        WNTMFilterBool = data3.label.isin(FilterStrongWNTM)
        data3FilterStrongWNTM = data3[WNTMFilterBool]

In [ ]: from sklearn.feature_selection import chi2
        import numpy as np
```

Figure 97: Word vector preparation for examination of significant words per topic according to Chi^2 test of independence



```
In [17]:
FilterStrongWNTM
N = 20
for  xxx in FilterStrongWNTM:
    chiVals = chi2(tfidfTestChi, (labels == xxx))
    indices = np.argsort(chiVals[0])# gives some number, a  ranking to scores
    Pvals = chiVals[1]
    sortedPvals = Pvals[indices]
    feature_names = np.array(tfidf.get_feature_names())[indices] # find the names from word vec, order by indices ran
    words = [w for w in feature_names]
    a = words[-N:]
    b = sortedPvals[-N:]
    res = [str(i) + " " + str(j) for i, j in zip(a, b)]
    print("####################################################  "+ str(xxx)+ "  ######################################
    print("Terms Most Correlated:\n {}".format('\n. '.join(res)))
```

```
####################################################  1  ########################################
Terms Most Correlated:
 dears 0.0
. incandescent 0.0
. males 0.0
. freaking 0.0
. irrefutable 0.0
. emergence 0.0
. male 0.0
. represents 0.0
. misogyny 0.0
. white 0.0
. demon 0.0
. hurtling 0.0
. obsolescence 0.0
. convergence 0.0
. hints 0.0
. certain 0.0
. middle 0.0
. aged 0.0
. men 0.0
. triggering 0.0
####################################################  5  ########################################
Terms Most Correlated:
 trip 0.0
. jets 0.0
. hummer 0.0
. plane 0.0
. flying 0.0
. private 0.0
. sailboat 0.0
. sailing 0.0
```

Figure 98: Chi-squared test and relevant significant words per topic

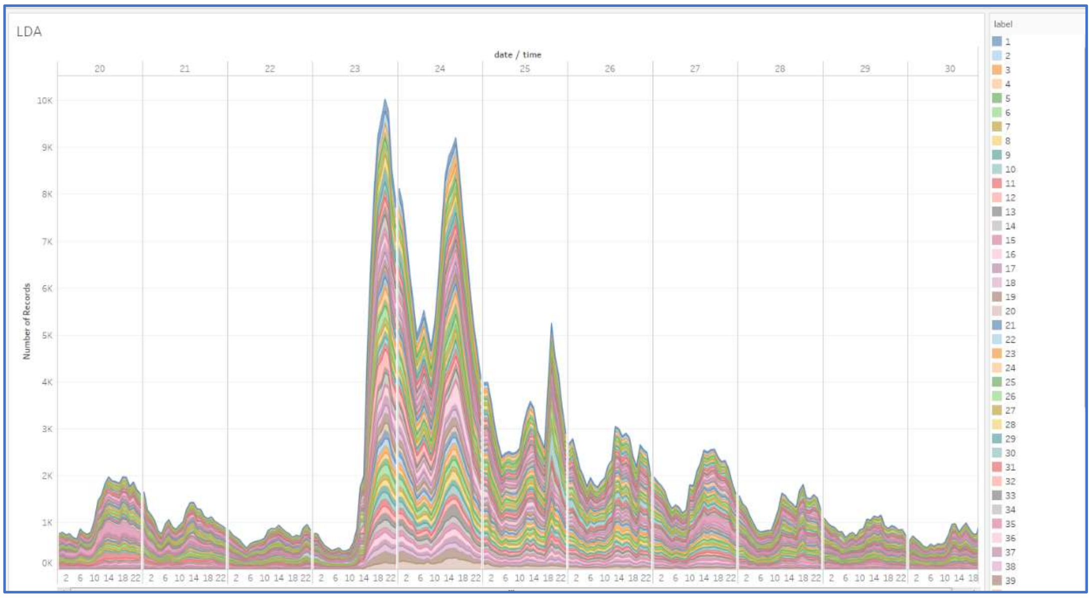


Figure 99: Time series plot of the topics modelled by LDA

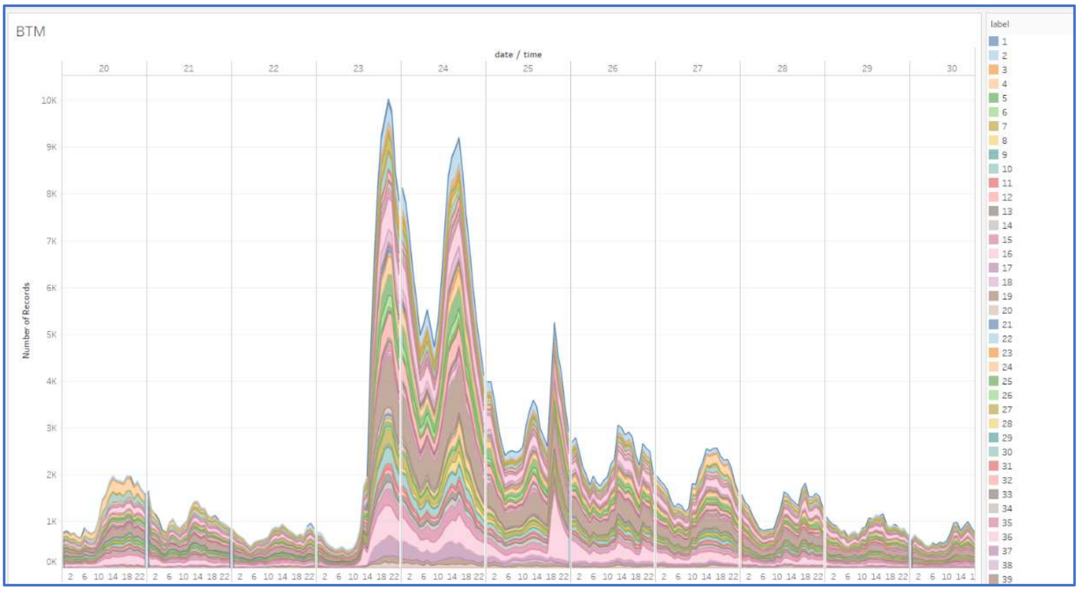


Figure 100: Time series plot of the topics modelled by BTM

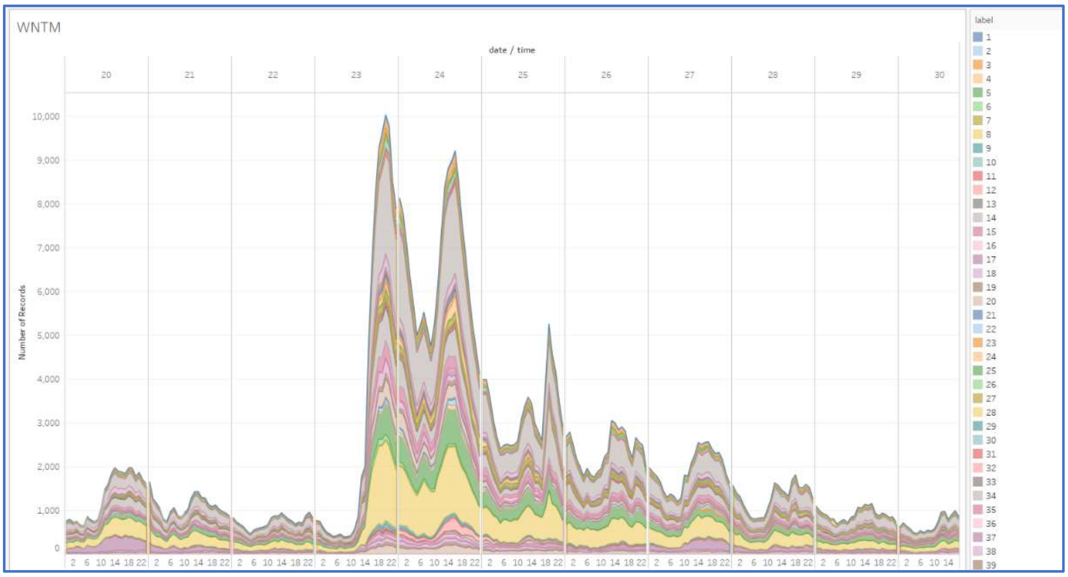


Figure 101: Time series plot of the topics modelled by WNTM

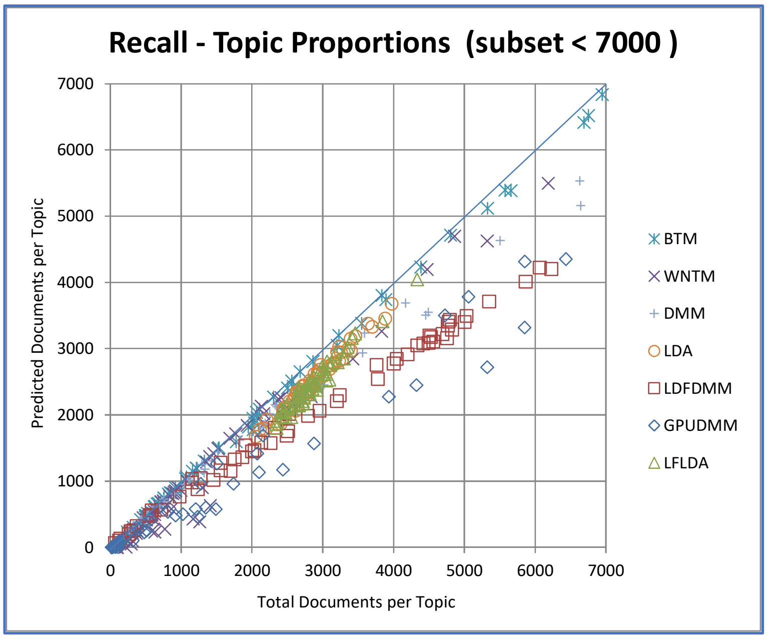


Figure 102: Recall based on Topic Proportions (extracted plot of documents < 7000)

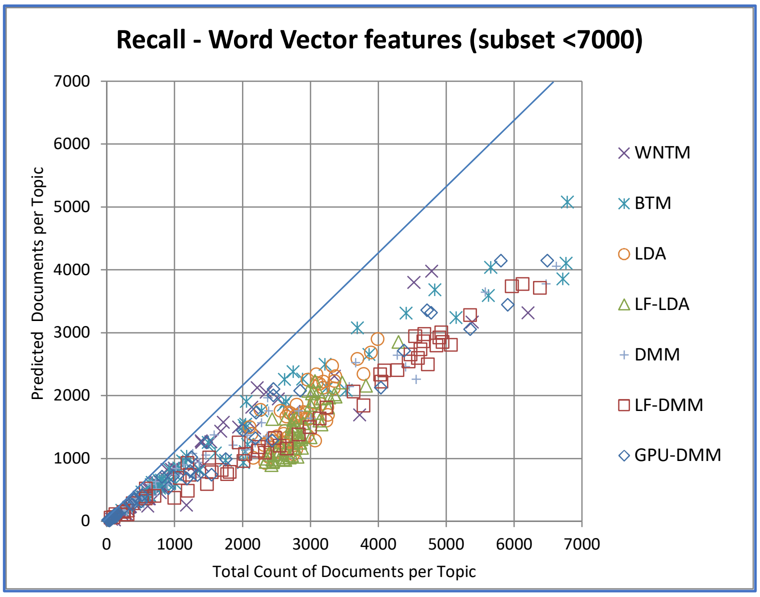


Figure 103: Recall based on Word Vectors (extracted plot of documents < 7000)