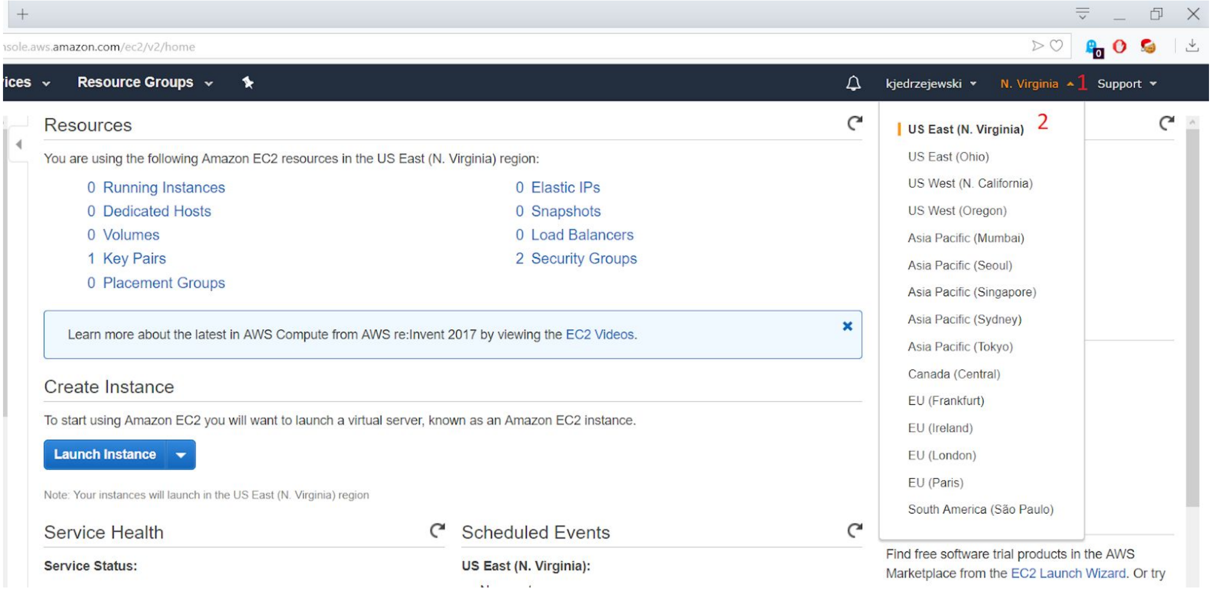
1. Redeeming a credit code

Go to https://console.aws.amazon.com/billing/home#/credits and activate your code.

2. Creating an EC2 instance

1. Go to the EC2 console (https://console.aws.amazon.com/ec2/v2/home)
2. Open the region menu (1) and select ***US East (N. Virginia)*** (2)



Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

0 Running Instances	0 Elastic IPs
0 Dedicated Hosts	0 Snapshots
0 Volumes	0 Load Balancers
1 Key Pairs	2 Security Groups
0 Placement Groups	

Learn more about the latest in AWS Compute from AWS re:Invent 2017 by viewing the EC2 Videos.

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US East (N. Virginia) region

Service Health

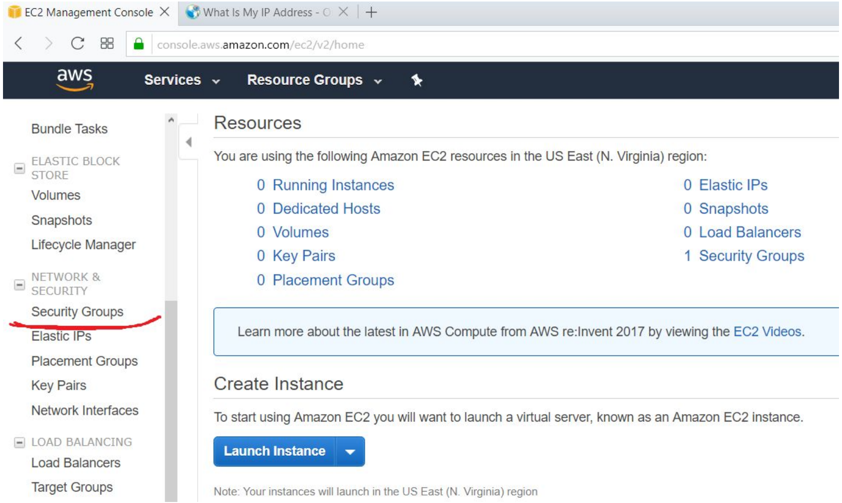
Service Status:

Scheduled Events

US East (N. Virginia):

Find free software trial products in the AWS Marketplace from the EC2 Launch Wizard. Or try

3. Use left sidebar to go to the ***Security Groups*** screen:



Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

0 Running Instances	0 Elastic IPs
0 Dedicated Hosts	0 Snapshots
0 Volumes	0 Load Balancers
0 Key Pairs	1 Security Groups
0 Placement Groups	

Learn more about the latest in AWS Compute from AWS re:Invent 2017 by viewing the EC2 Videos.

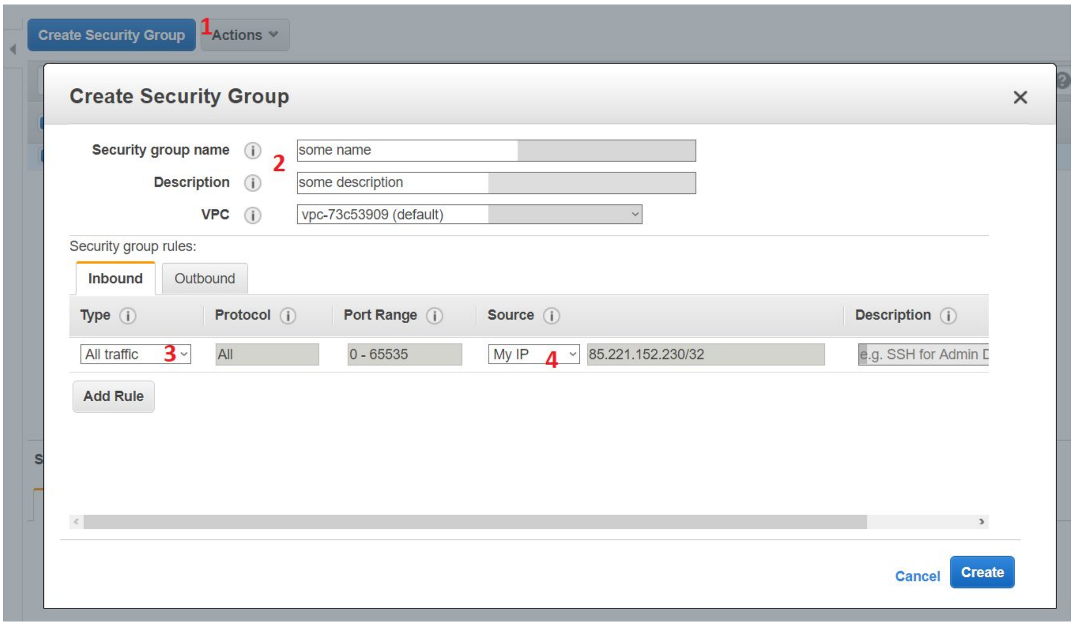
Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US East (N. Virginia) region

4. On that screen:
   (1) click ***Create Security Group*** button
   (2) enter a name and a description of the group

(3) Add a rule with type ***All traffic*** and source ***My IP*** (4)



Security group name: some name

Description: some description

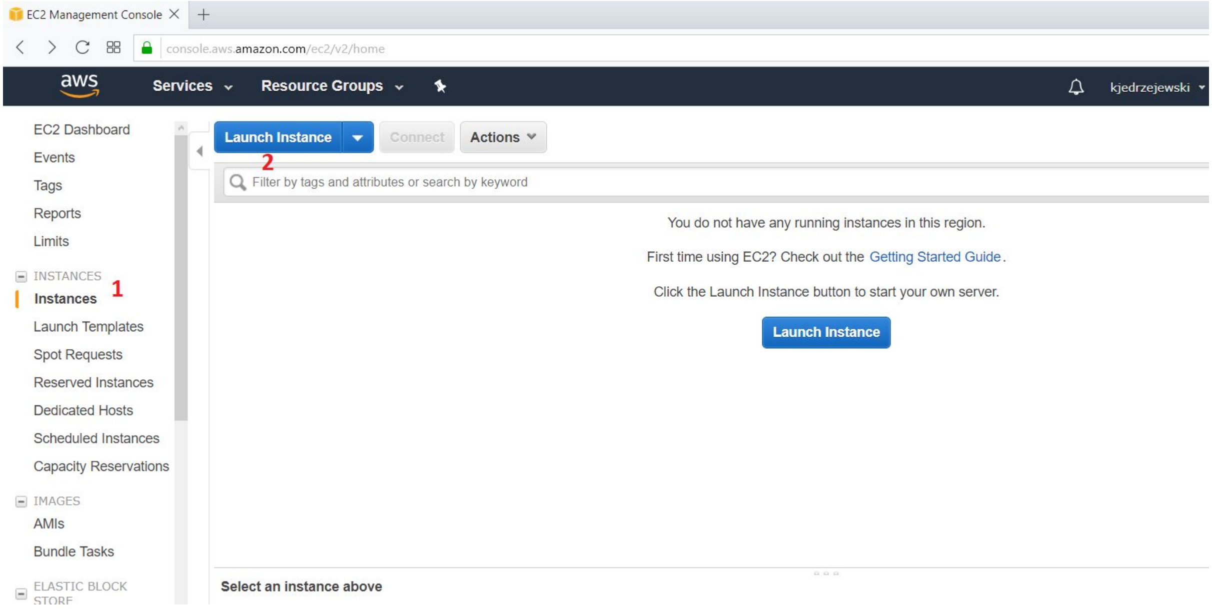
VPC: vpc-73c53909 (default)

Security group rules:

Inbound | Outbound

Type	Protocol	Port Range	Source		Description
All traffic	All	0 - 65535	My IP	85.221.152.230/32	e.g. SSH for Admin D

5. Now, go to the ***Instances*** (1) screen, and click ***Launch instance*** (2) button



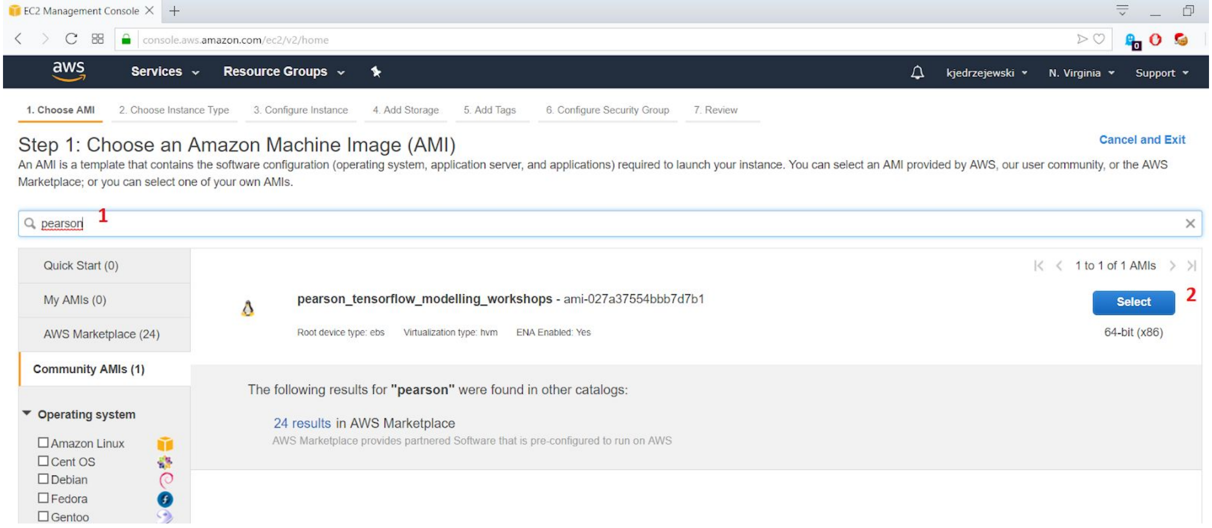
You do not have any running instances in this region.

First time using EC2? Check out the Getting Started Guide.

Click the Launch Instance button to start your own server.

Select an instance above

6. In the Step 1, select ***Community AMIs***, put “pearson” in the search box (1), and click the ***Select*** button (2).



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

pearson 1

pearson_tensorflow_modelling_workshops - ami-027a37554bbb7d7b1

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

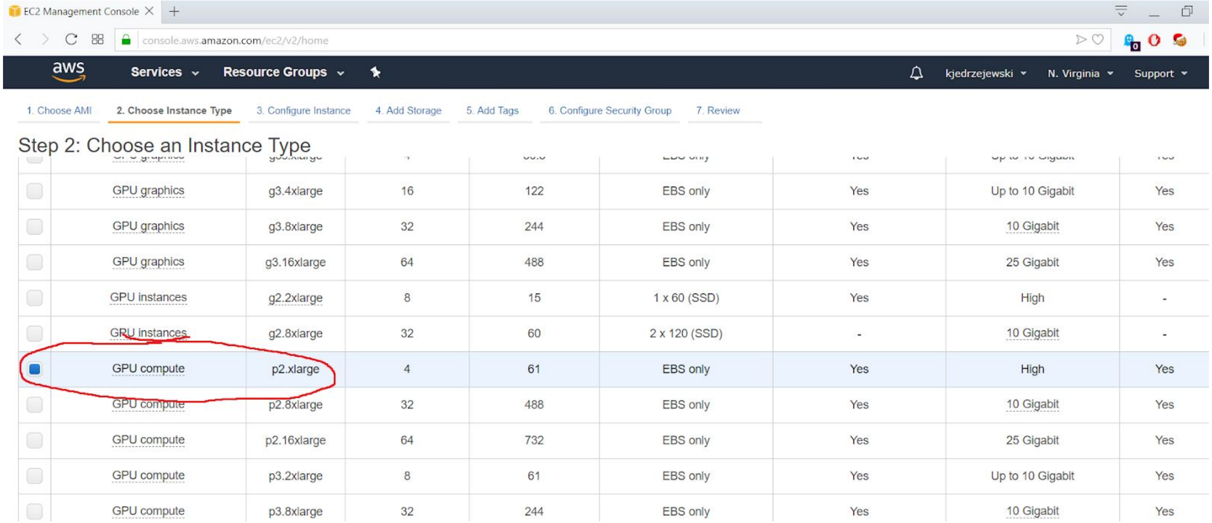
64-bit (x86)

The following results for **"pearson"** were found in other catalogs:

24 results in AWS Marketplace

AWS Marketplace provides partnered Software that is pre-configured to run on AWS

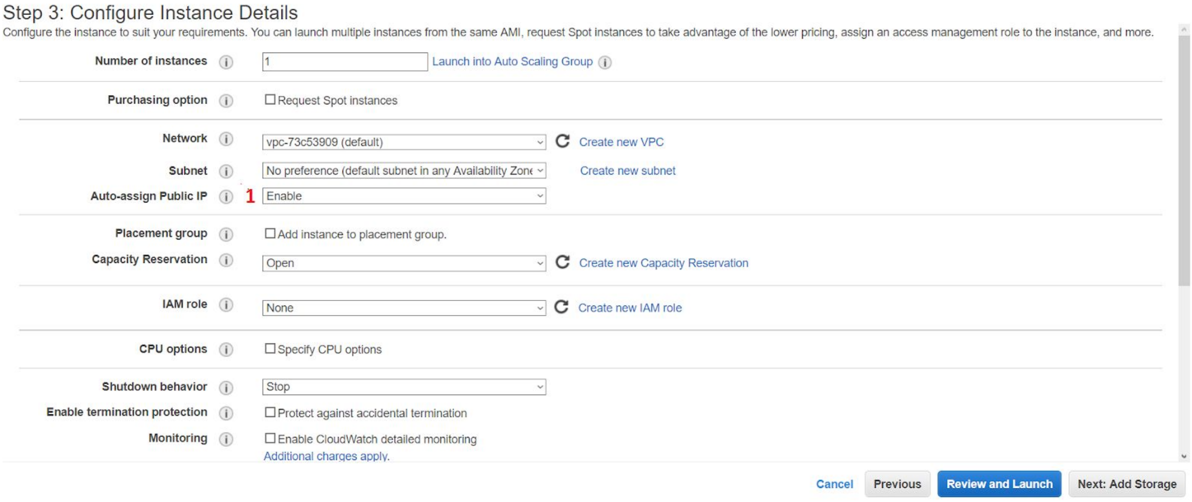
7. In the Step 2, select ***p2.xlarge***, and click the ***Next*** button. If you have the p2.xlarge instance limit at 0, select **t3.2xlarge**



Step 2: Choose an Instance Type

☐	GPU graphics	g3.4xlarge	16	122	EBS only	Yes	Up to 10 Gigabit	Yes
☐	GPU graphics	g3.8xlarge	32	244	EBS only	Yes	10 Gigabit	Yes
☐	GPU graphics	g3.16xlarge	64	488	EBS only	Yes	25 Gigabit	Yes
☐	GPU instances	g2.2xlarge	8	15	1 x 60 (SSD)	Yes	High	-
☐	GPU instances	g2.8xlarge	32	60	2 x 120 (SSD)	-	10 Gigabit	-
☑	GPU compute	p2.xlarge	4	61	EBS only	Yes	High	Yes
☐	GPU compute	p2.8xlarge	32	488	EBS only	Yes	10 Gigabit	Yes
☐	GPU compute	p2.16xlarge	64	732	EBS only	Yes	25 Gigabit	Yes
☐	GPU compute	p3.2xlarge	8	61	EBS only	Yes	Up to 10 Gigabit	Yes
☐	GPU compute	p3.8xlarge	32	244	EBS only	Yes	10 Gigabit	Yes

8. In the Step 3, set ***Auto-assign Public IP*** to ***Enable***, and click the ***Next*** button



Step 3: Configure Instance Details

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

Number of instances 1 Launch into Auto Scaling Group

Purchasing option □ Request Spot instances

Network vpc-73c53909 (default) Create new VPC

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

Auto-assign Public IP 1 Enable

Placement group □ Add instance to placement group.

Capacity Reservation Open Create new Capacity Reservation

IAM role None Create new IAM role

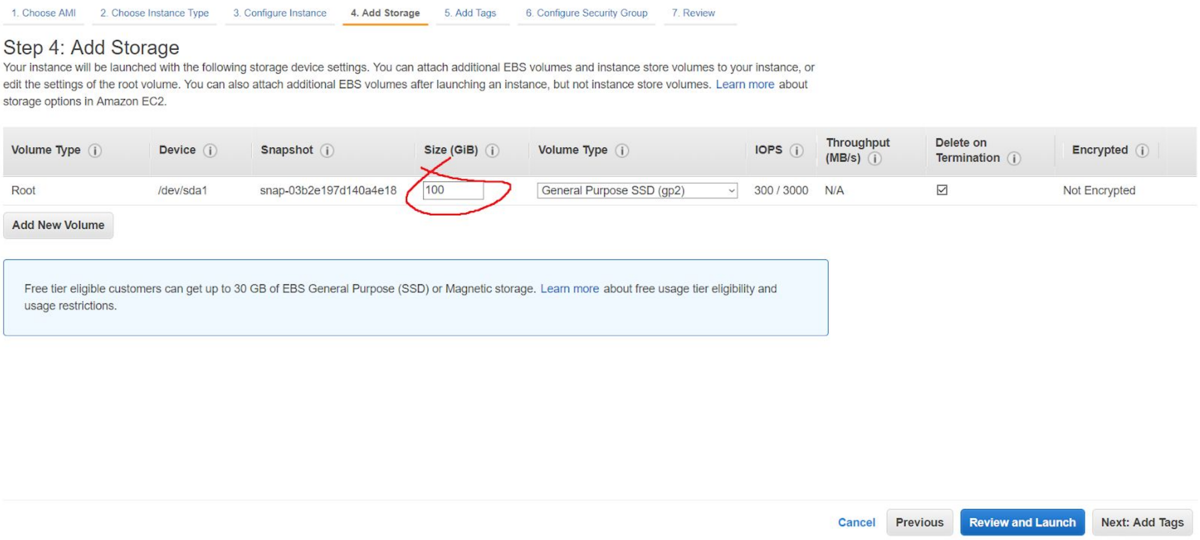
CPU options □ Specify CPU options

Shutdown behavior Stop

Enable termination protection □ Protect against accidental termination

Monitoring □ Enable CloudWatch detailed monitoring
Additional charges apply.

9. In the Step 4, leave 100 GB as the storage size, and click the ***Next*** button



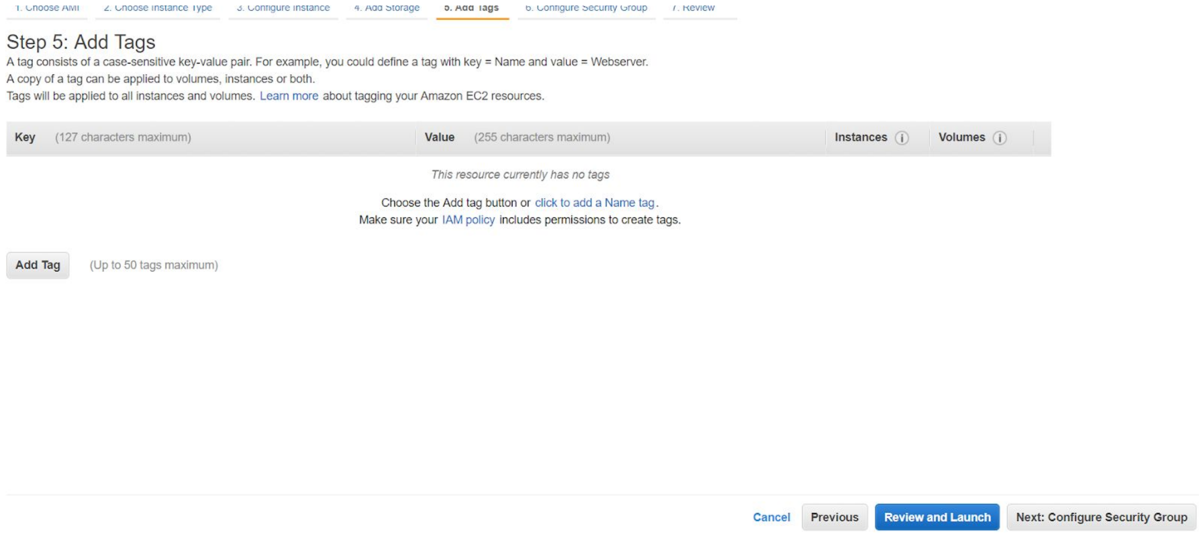
Step 4: Add Storage

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	Encrypted
Root	/dev/sda1	snap-03b2e197d140a4e18	100	General Purpose SSD (gp2)	300 / 3000	N/A	☑	Not Encrypted

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.

10. In the Step 5, just click the ***Next*** button



Step 5: Add Tags

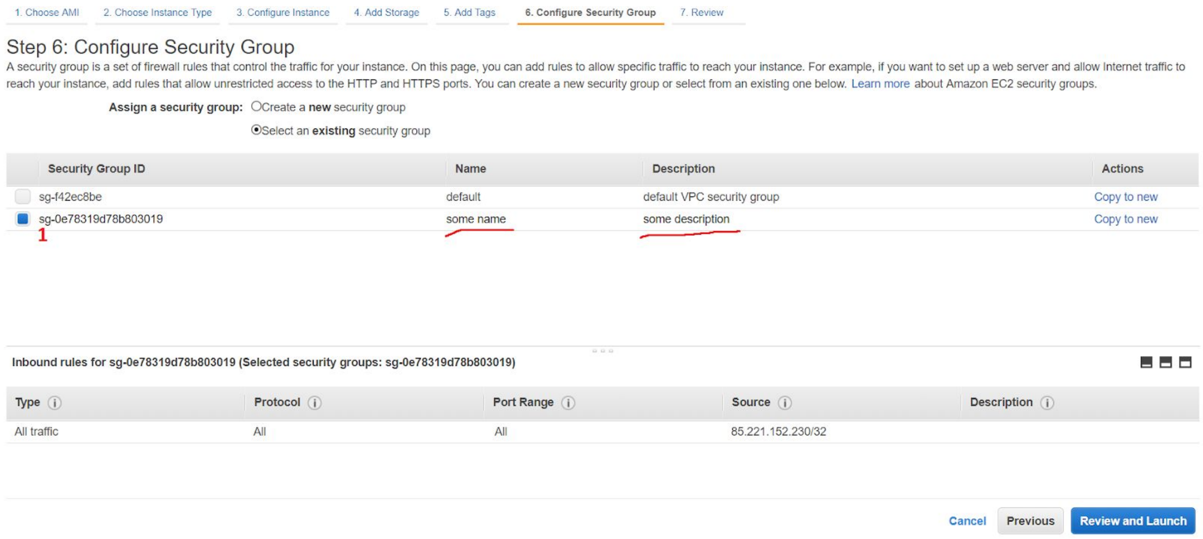
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)	Instances	Volumes

This resource currently has no tags

Choose the Add tag button or click to add a Name tag.
Make sure your IAM policy includes permissions to create tags.

11. In the Step 6, switch to ***Select an existing security group***, and select a group you created earlier (in the point 2.3). Click the ***Review and Launch*** button



Step 6: Configure Security Group

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

Assign a security group: ○Create a **new** security group
●Select an **existing** security group

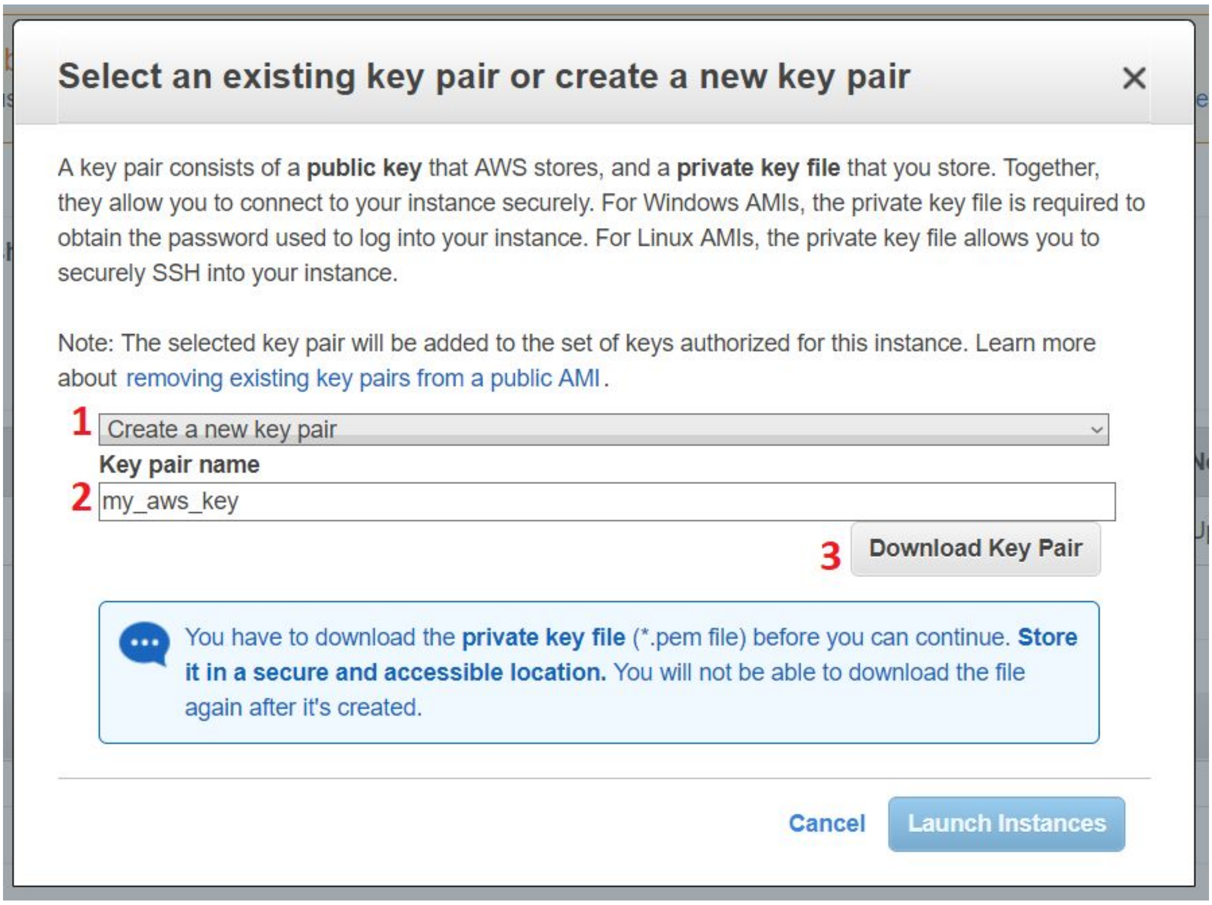
	Security Group ID	Name	Description	Actions
□	sg-f42ec8be	default	default VPC security group	Copy to new
■	sg-0e78319d78b803019	some name	some description	Copy to new

Inbound rules for sg-0e78319d78b803019 (Selected security groups: sg-0e78319d78b803019)

Type	Protocol	Port Range	Source	Description
All traffic	All	All	85.221.152.230/32	

12. In the Step 7, just click ***Launch***

13. In the window that will pop-up:
    (1) select ***Create a new key pair***
    (2) put the name of your key
    (3) click ***Download Key Pair***, as save it in some safe place. You will need it soon.
    Click ***Launch*** instance



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 .

1 Create a new key pair

Key pair name

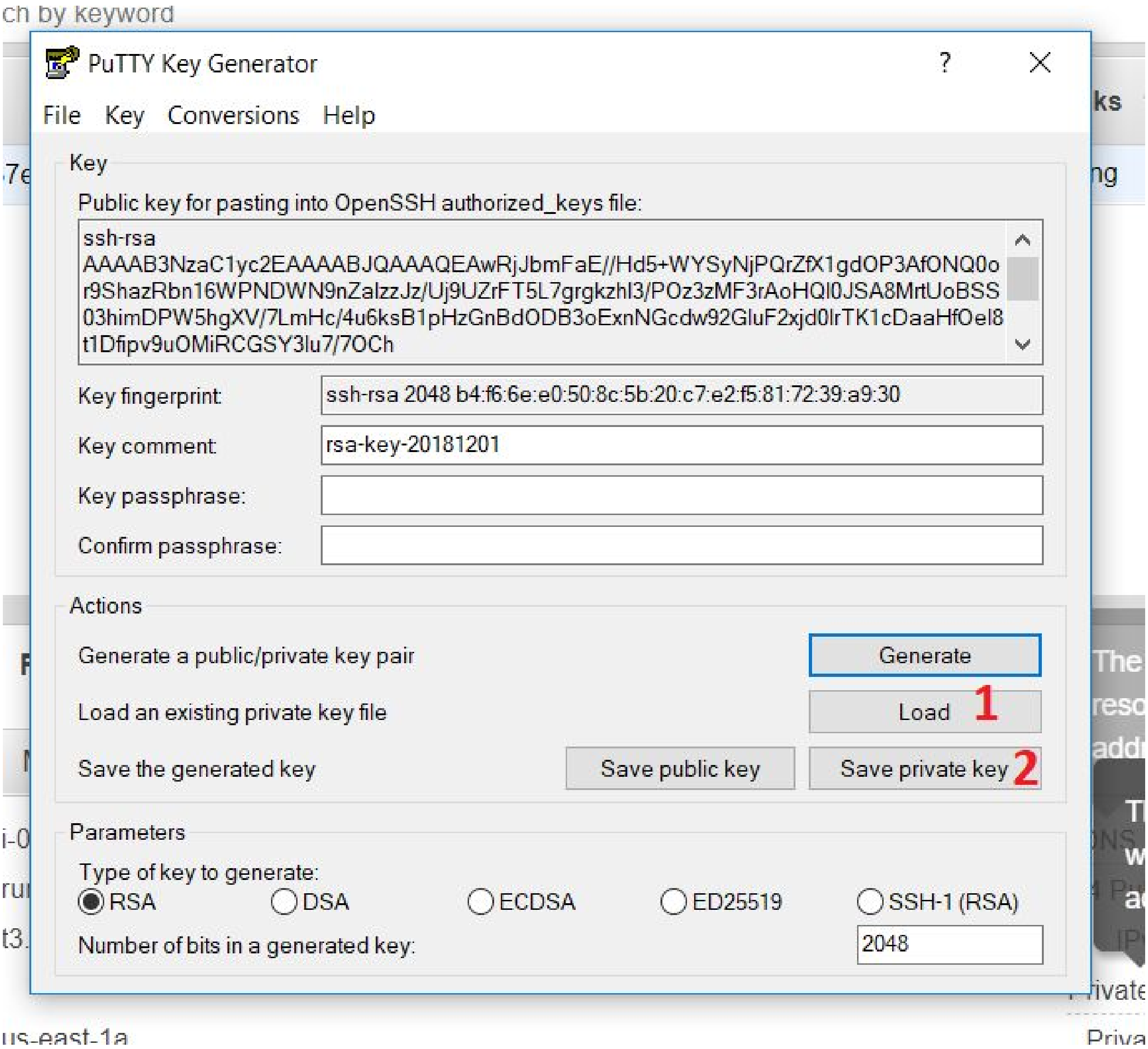
2 my_aws_key

3 Download Key Pair

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

3. Windows - Connecting to an EC2 instance - from

1. Go to the EC2 console (https://console.aws.amazon.com/ec2/v2/home)
2. Open PuTTYgen. (1) Load the .pem file you downloaded in point 3.13. (2) Save private key as the .ppk file.



Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAwRjJbmFaE//Hd5+WYSyNjPQrZfX1gdOP3AfONQ0o
r9ShazRbn16WPNDWN9nZalzzJz/Uj9UZrFT5L7grgkzhl3/POz3zMF3rAoHQI0JSA8MrtUoBSS
03himDPW5hgXV/7LmHc/4u6ksB1pHzGnBdODB3oExnNGcdw92GluF2xjd0lrTK1cDaaHf0el8
t1Dfipv9uOMiRCGSY3lu7/7OCh
```

Key fingerprint: ssh-rsa 2048 b4:f6:6e:e0:50:8c:5b:20:c7:e2:f5:81:72:39:a9:30

Key comment: rsa-key-20181201

Key passphrase:

Confirm passphrase:

Generate a public/private key pair

Load an existing private key file

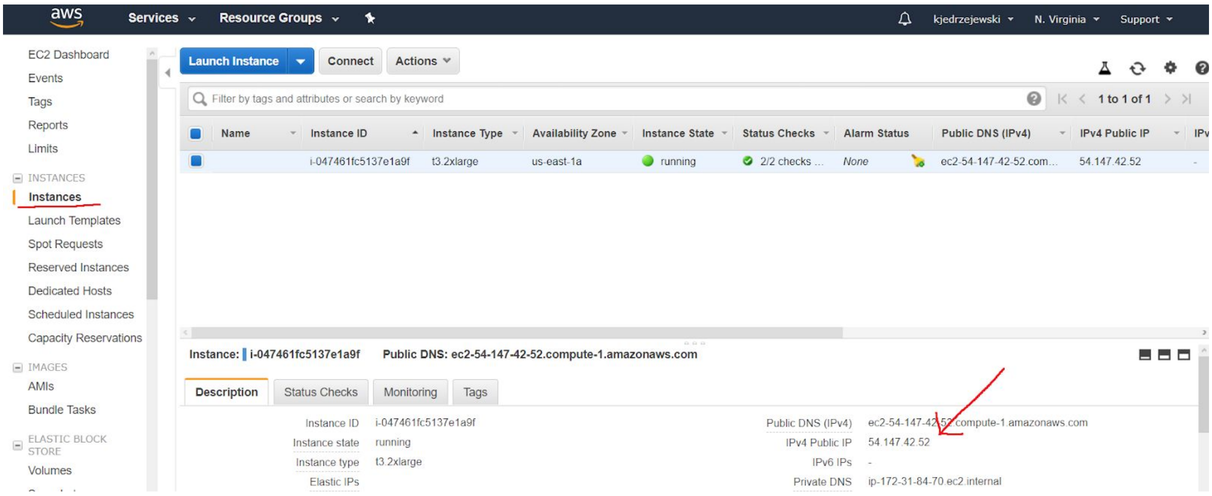
Save the generated key

Parameters

Type of key to generate:
● RSA ○ DSA ○ ECDSA ○ ED25519 ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

3. In the instance screen, select an instance you created earlier, and copy it’s ***IPv4 Public IP*** value

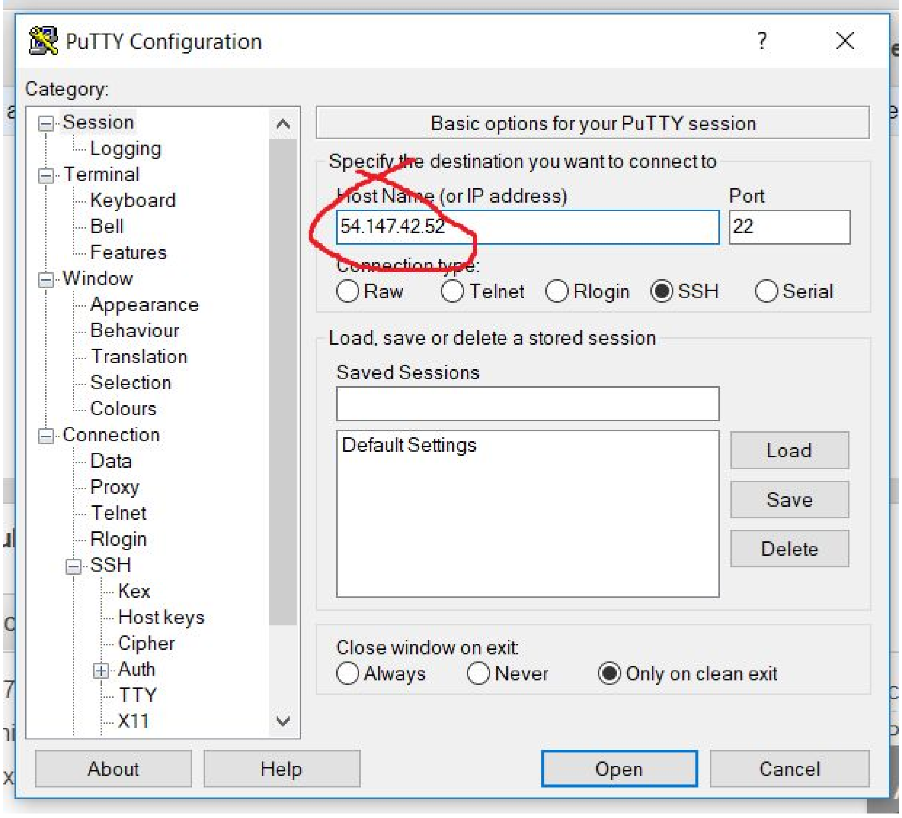


Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv
	i-047461fc5137e1a9f	t3.2xlarge	us-east-1a	running	2/2 checks ...	None	ec2-54-147-42-52.com...	54.147.42.52	-

Instance: i-047461fc5137e1a9f Public DNS: ec2-54-147-42-52.compute-1.amazonaws.com

Instance ID	i-047461fc5137e1a9f	Public DNS (IPv4)	ec2-54-147-42-52.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	54.147.42.52
Instance type	t3.2xlarge	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-84-70.ec2.internal

4. Open Putty. Enter the IP address from the previous step as the ***Host Name***



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 54.147.42.52

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

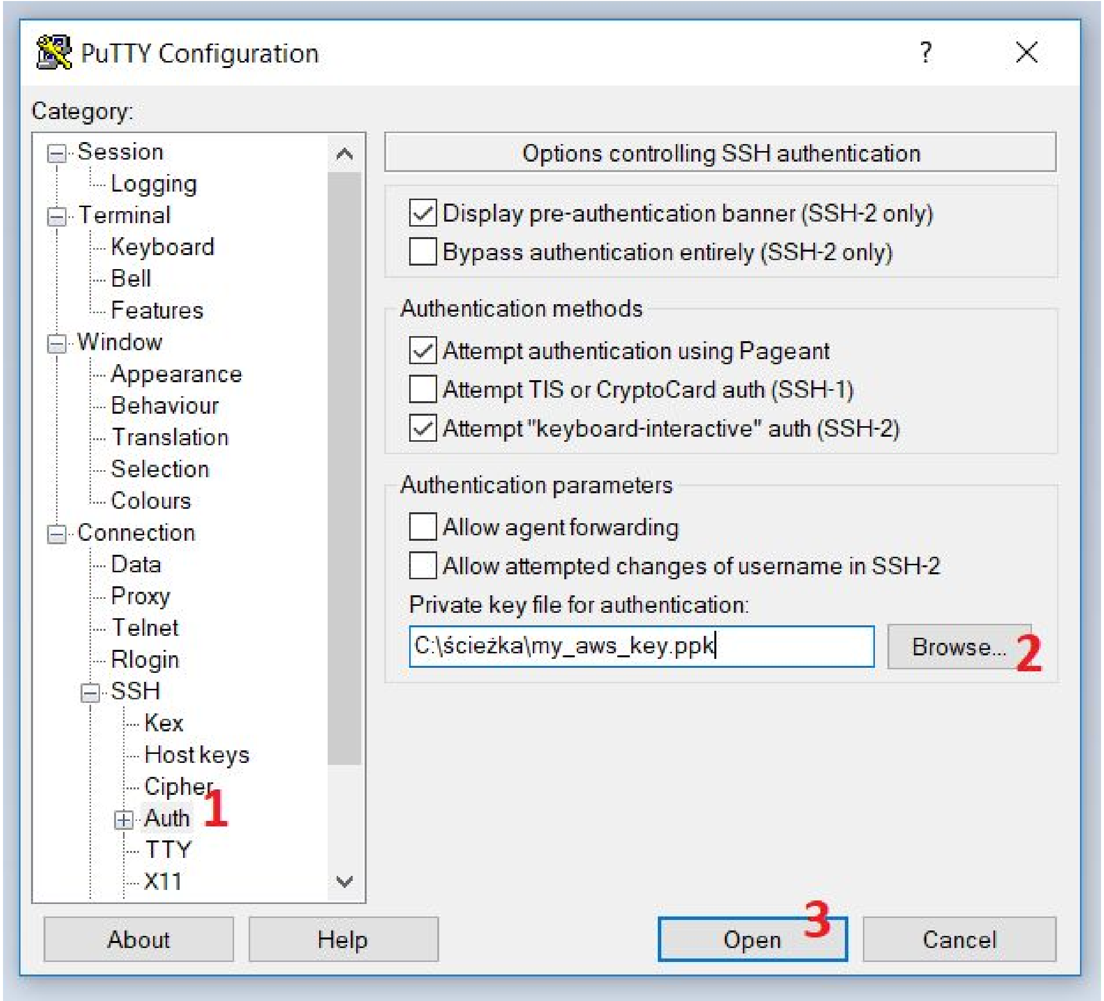
Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

5. (1) Using the left sidebar menu go to ***Connection*** -> ***SSH*** -> ***Auth***
(2) Open the private key from the .ppk file

(3) Click ***Open***



Options controlling SSH authentication

☑ Display pre-authentication banner (SSH-2 only)

☐ Bypass authentication entirely (SSH-2 only)

Authentication methods

☑ Attempt authentication using Pageant

☐ Attempt TIS or CryptoCard auth (SSH-1)

☑ Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

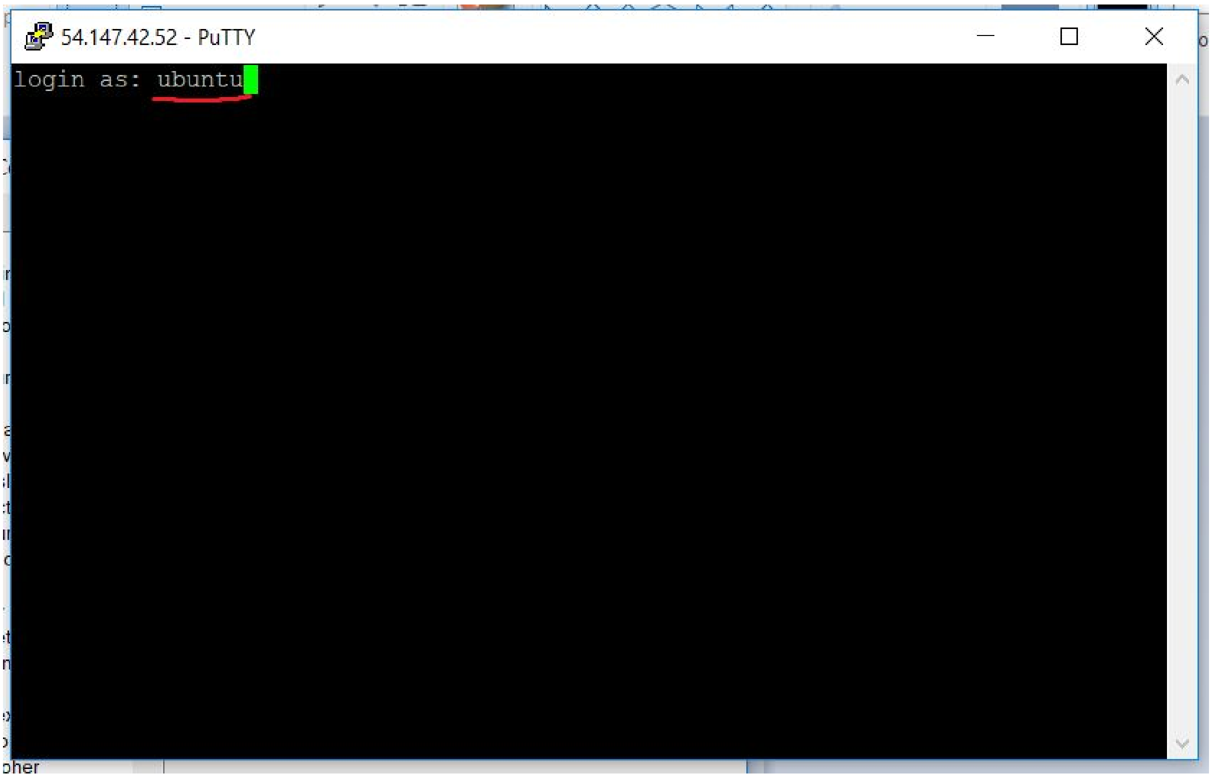
☐ Allow agent forwarding

☐ Allow attempted changes of username in SSH-2

Private key file for authentication:

C:\ścieżka\my_aws_key.ppk

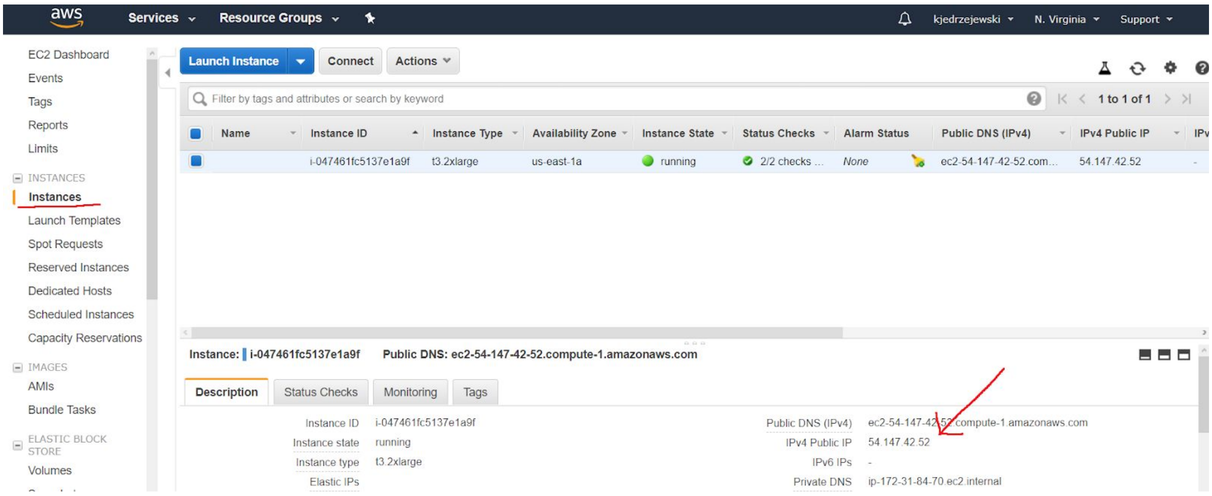
6. In the terminal screen that will open, put ***ubuntu*** as the user name



```
login as: ubuntu
```

4. Linux or MacOS - Connecting to an EC2 instance

1. Open the Terminal / bash / whatever you use
2. Move the .pem file that you downloaded in the point 3.13 to ***~/.ssh/*** directory, e.g.:
   ```
   mv /path/to/a/file.pem ~/.ssh/
   ```
3. Change permissions of this .pem file to 400, e.g.
   ```
   chmod 400 ~/.ssh/file.pem
   ```
4. In the instance screen, select an instance you created earlier, and copy it's ***IPv4 Public IP*** value



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv
	i-047461fc5137e1a9f	t3.2xlarge	us-east-1a	● running	✓ 2/2 checks ...	None	ec2-54-147-42-52.com...	54.147.42.52	-

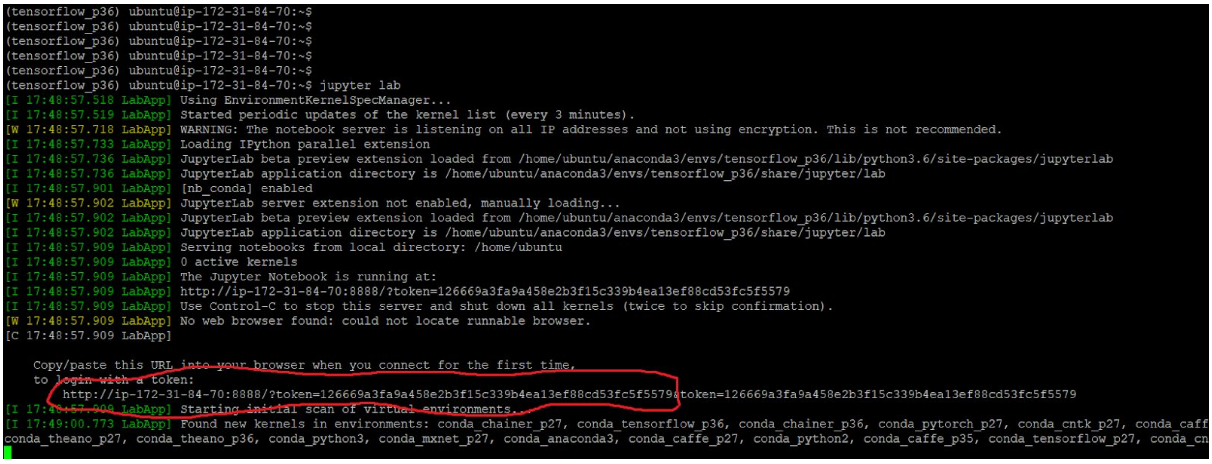
Instance: i-047461fc5137e1a9f Public DNS: ec2-54-147-42-52.compute-1.amazonaws.com

Instance ID	i-047461fc5137e1a9f	Public DNS (IPv4)	ec2-54-147-42-52.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	54.147.42.52
Instance type	t3.2xlarge	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-84-70.ec2.internal

5. Connect to the EC2 machine using ***ssh*** command. Use your .pem file, username ***ubuntu*** and the IP address from the previous step, e.g.
   ```
   ssh -i "~/.ssh/file.pem" ubuntu@54.147.42.52
   ```

5. Running jupyter lab, and connecting to it

1. Connect to your EC2 machine, e.g. as described in 3. or 4.
2. Go to the home directory, e.g.
   ```
   cd ~/
   ```
3. Clone github repository at https://github.com/kjedrzejewski/tensorflow_mle_workshops.git, e.g.
   ```
   git clone
   https://github.com/kjedrzejewski/tensorflow_mle_workshops.git
   ```
4. Go to the repository directory, e.g.
   ```
   cd tensorflow_mle_workshops
   ```
5. Activate the environment ***TensorFlow(+Keras2) with Python3 (CUDA 9.0 and Intel MKL-DNN)*** with:
   ```
   source activate tensorflow_p36
   ```
6. Start Jupyter Lab with:
   ```
   jupyter lab
   ```
7. From the output of the previous command copy Jupyter Lab URL, which looks like:
   ```
   http://ip-172-31-84-70:8888/?token=126669a3fa9a458e2b3f15c339
   b4ea13ef88cd53fc5f5579
   ```



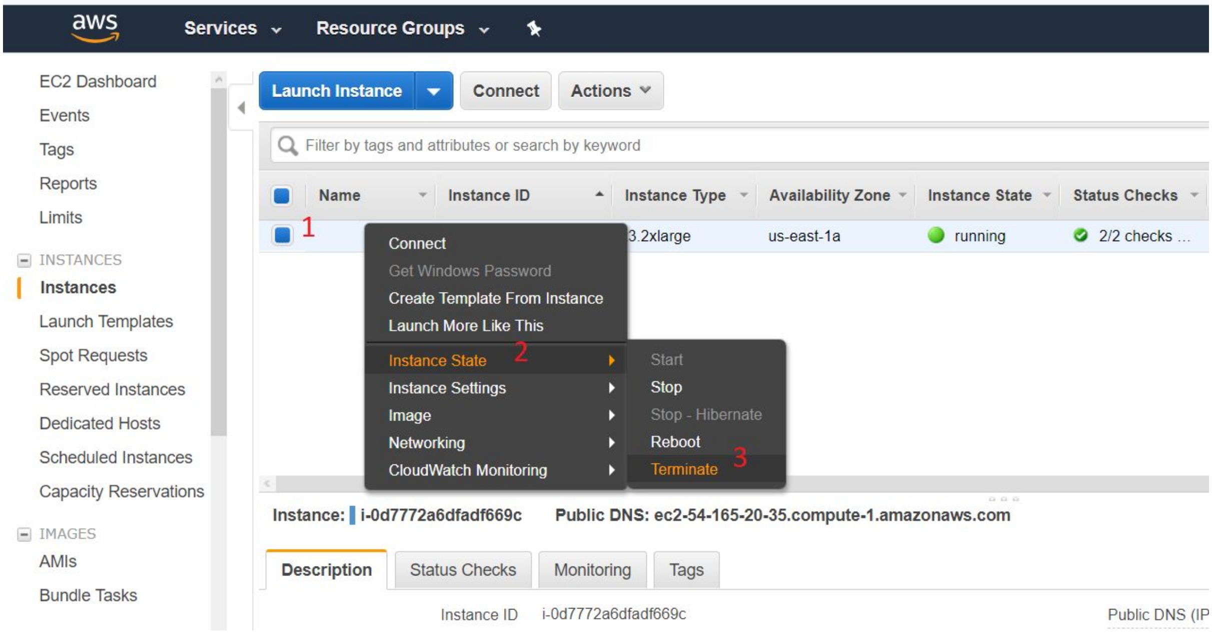
```
(tensorflow_p36) ubuntu@ip-172-31-84-70:~$
(tensorflow_p36) ubuntu@ip-172-31-84-70:~$
(tensorflow_p36) ubuntu@ip-172-31-84-70:~$
(tensorflow_p36) ubuntu@ip-172-31-84-70:~$
(tensorflow_p36) ubuntu@ip-172-31-84-70:~$
(tensorflow_p36) ubuntu@ip-172-31-84-70:~$ jupyter lab
[I 17:48:57.518 LabApp] Using EnvironmentKernelSpecManager...
[I 17:48:57.519 LabApp] Started periodic updates of the kernel list (every 3 minutes).
[W 17:48:57.718 LabApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not recommended.
[I 17:48:57.733 LabApp] Loading IPython parallel extension
[I 17:48:57.736 LabApp] JupyterLab beta preview extension loaded from /home/ubuntu/anaconda3/envs/tensorflow_p36/lib/python3.6/site-packages/jupyterlab
[I 17:48:57.736 LabApp] JupyterLab application directory is /home/ubuntu/anaconda3/envs/tensorflow_p36/share/jupyter/lab
[I 17:48:57.901 LabApp] [nb_conda] enabled
[W 17:48:57.902 LabApp] JupyterLab server extension not enabled, manually loading...
[I 17:48:57.902 LabApp] JupyterLab beta preview extension loaded from /home/ubuntu/anaconda3/envs/tensorflow_p36/lib/python3.6/site-packages/jupyterlab
[I 17:48:57.902 LabApp] JupyterLab application directory is /home/ubuntu/anaconda3/envs/tensorflow_p36/share/jupyter/lab
[I 17:48:57.909 LabApp] Serving notebooks from local directory: /home/ubuntu
[I 17:48:57.909 LabApp] 0 active kernels
[I 17:48:57.909 LabApp] The Jupyter Notebook is running at:
[I 17:48:57.909 LabApp] http://ip-172-31-84-70:8888/?token=126669a3fa9a458e2b3f15c339b4ea13ef88cd53fc5f5579
[I 17:48:57.909 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[W 17:48:57.909 LabApp] No web browser found: could not locate runnable browser.
[C 17:48:57.909 LabApp]

    Copy/paste this URL into your browser when you connect for the first time,
    to login with a token:
        http://ip-172-31-84-70:8888/?token=126669a3fa9a458e2b3f15c339b4ea13ef88cd53fc5f5579&token=126669a3fa9a458e2b3f15c339b4ea13ef88cd53fc5f5579
[I 17:48:57.909 LabApp] Starting initial scan of virtual environments...
[I 17:49:00.773 LabApp] Found new kernels in environments: conda_chainer_p27, conda_tensorflow_p36, conda_chainer_p36, conda_pytorch_p27, conda_cntk_p27, conda_caff
conda_theano_p27, conda_theano_p36, conda_python3, conda_mxnet_p27, conda_anaconda3, conda_caffe_p27, conda_python2, conda_caffe_p35, conda_tensorflow_p27, conda_cn
```

8. Now, replace address part in this URL, with the IP address of your machine (from point 3.3 or 4.4), it should now look like:
   ```
   http://54.147.42.52:8888/?token=126669a3fa9a458e2b3f15c339b4e
   a13ef88cd53fc5f5579
   ```
9. Open this address in the web browser

6. Deleting a machine

Go to the EC2 console (https://console.aws.amazon.com/ec2/v2/home)
(1) Select the machine you want to delete (and not pay for it any longer)
(2) Go to ***Instance State***
(3) Select ***Terminate***



Instance: i-0d7772a6dfadf669c Public DNS: ec2-54-165-20-35.compute-1.amazonaws.com

Instance ID i-0d7772a6dfadf669c Public DNS (IP