

How to Deploy your ML Model in AWS EC2?

In this guide we will go over the steps to deploy your ML model app in Amazon EC2 service. Here the ML app is a flask based. However, the same steps and ideas can be followed to host apps built with other similar applications.

Note that we will not be covering how to build an ML model. Instead you will use the ready made code that trains the ML model and store it as a pickle file directly. This guide is about how to host an already running ML model in AWS EC2.

Below is the brief of the steps you will need to do.

Steps to Deploy your ML app in AWS EC2

Below are the steps we will follow to host and serve the ML model from AWS EC2.

Step 1. Download the [code](#). Then, build your ML model locally and start it as a flask app. We will use this model and host it in AWS EC2.

Step 2. Launch an EC2 instance in AWS. A free tier instance is sufficient for demo purposes.

Step 3. Connect to AWS EC2 instance using ssh

Step 4. Move your files to AWS Ec2 using Secure Copy (Scp)

Step 5. Install the necessary packages and run app.py to start the app.

You can now access the app from the browser in the designated URL you will get when you created the EC2 instance.

Pre Requisites

1. A working ML model built as a REST API, preferably with flask or similar framework. You can download the code for the model app [here](#).
2. AWS Account. If you don't have one, you can [create](#) it here. You will need a credit card to create one, however, to deploy your model, we are going to be using a free tier instance, which will not incur cost.
3. You will need Python with an IDE installed to build, debug and serve your app locally. Recommend [anaconda](#) for this. Alternates: [VS Code](#), [PyCharm](#) or [Spyder](#).

Step 1: Run the ML flask app in your local computer

1. Download the code directory zip file and extract contents.

2. Open terminal or command prompt and change directory to the code folder: ``cd downloaded_code_directory``
3. Run ``python app.py``

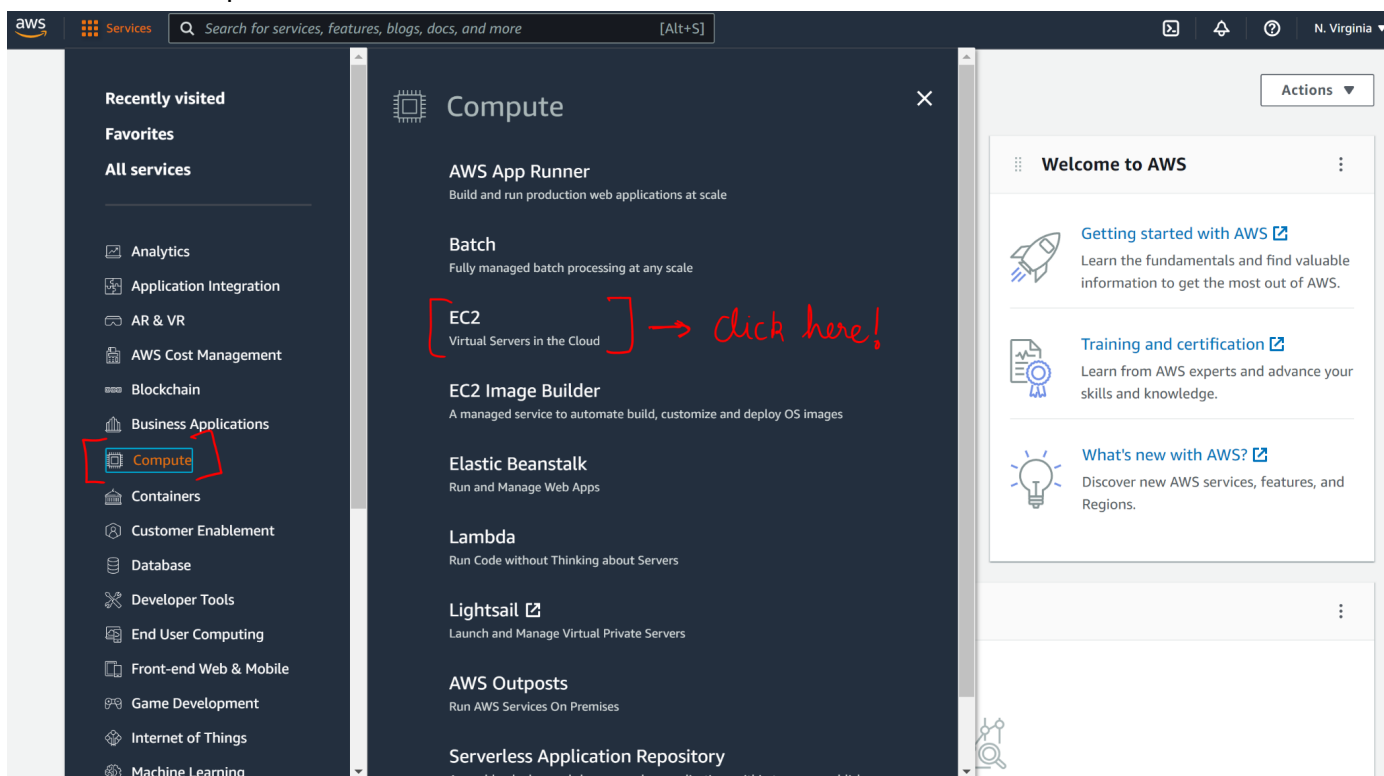
If everything goes well, the above step will start a flask app and give a local URL to access the app. Visit the URL to access your app screen.

Step 2: Launch a free tier micro instance on AWS

On creating an AWS account, launch a free tier EC2 instance.

1. [Login](#) to your AWS account from console.aws.amazon.com.

Then, Search 'EC2' in the search box in the top. Or you can find it in the list of services under 'Compute'



2. Launch an EC2 Instance

To launch an EC2 instance, you will have to go through a sequence of steps. An EC2 instance is nothing but a remote computer that will run at an Amazon Data Center that we can lease to host our ML app.

Let's go over the steps one by one.

(i) Click the **launch instance** button from the EC2 dashboard.

This will get you started with creating a new EC2 instance.

The screenshot shows the AWS Management Console for the EC2 service. The top navigation bar includes the AWS logo, a search bar, and a user profile icon. The left sidebar contains a navigation menu with options like 'New EC2 Experience', 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', 'Instances', 'Images', and 'Elastic Block Store'. The main content area is titled 'Resources' and shows a list of EC2 resources in the US East (N. Virginia) Region. A red box highlights the 'Launch instance' button, with a red arrow pointing to it and the handwritten text 'click to launch new instance'. The right sidebar shows 'Account attributes' and 'Explore AWS' options.

(ii) Choose an AMI Image that is eligible for Free Tier

The following screen will list the available EC2 instances, also called **Amazon Machine Image** (AMI). Pick one that is '**Free Tier Eligible**'. Other instance types will accrue cost.

aws Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia machinelearningplus

We are replacing this launch experience with a new launch experience, which we will continue to improve based on your feedback. Opt-in to the new experience by selecting the button on the right and give us feedback. For now you can still opt out once you have tried it. [Opt-in to the new experience](#)

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

Step 1: Choose an Amazon Machine Image (AMI)

Make sure it is free tier eligible

AMI	Root device type	Virtualization type	ENA Enabled	Architecture	Action
Ubuntu Server 22.04 LTS (HVM), SSD Volume Type - ami-09d56f8956ab235b3 (64-bit x86) / ami-02ddaf75821f25213 (64-bit Arm)	ebs	hvm	Yes	64-bit (x86)	Select
Ubuntu Server 22.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).					
Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-0c4f7023847b90238 (64-bit x86) / ami-0d70a59d7191a8079 (64-bit Arm)	ebs	hvm	Yes	64-bit (x86)	Select
Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).					
Microsoft Windows Server 2019 Base - ami-08ed5c5dd62794ec0	ebs	hvm	Yes	64-bit (x86)	Select
Microsoft Windows 2019 Datacenter edition, [English]					
Deep Learning AMI (Ubuntu 18.04) Version 60.1 - ami-077f4dcd20ed5ed5d	ebs	hvm	Yes	64-bit (x86)	Select
MXNet-1.8, TensorFlow-2.7, PyTorch-1.10, Neuron, & others. NVIDIA CUDA, cuDNN, NCCL, Intel MKL-DNN, Docker, NVIDIA-Docker & EFA support. For fully managed experience, check: https://aws.amazon.com/sagemaker					
Deep Learning AMI GPU PyTorch 1.11.0 (Amazon Linux 2) 20220328 - ami-023d5aa1ee956059c	ebs	hvm	Yes	64-bit (x86)	Select

(iii) Choose the instance type that belong to the selected AMI

In the following screen, it will ask to select the instance type with number of CPUs, RAM, memory limit etc.

For our app, since we are going with the Free Tier, pick the one with **'t2.micro'** in green with one CPU, 1GB memory. Then click **'Review and Launch'** blue button at the bottom.

aws Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia machinelearningplus.com

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

Step 2: Choose an Instance Type

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

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

Currently selected: t2.micro (ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	t2	t2.xlarge	8	32	EBS only	-	Moderate	Yes
<input type="checkbox"/>	t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	t3	t3.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	t3	t3.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	t3	t3.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes

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

(iv) Review and Launch

aws Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia machinelearningplus.com

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

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

⚠ Improve your instances' security. Your security group, launch-wizard-1, is open to the world.

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.

You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details [Edit AMI](#)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type - ami-09d56f8956ab235b3

Free tier eligible

Ubuntu Server 22.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Root Device Type: ebs Virtualization type: hvm

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

Security Groups [Edit security groups](#)

Security group name: launch-wizard-1

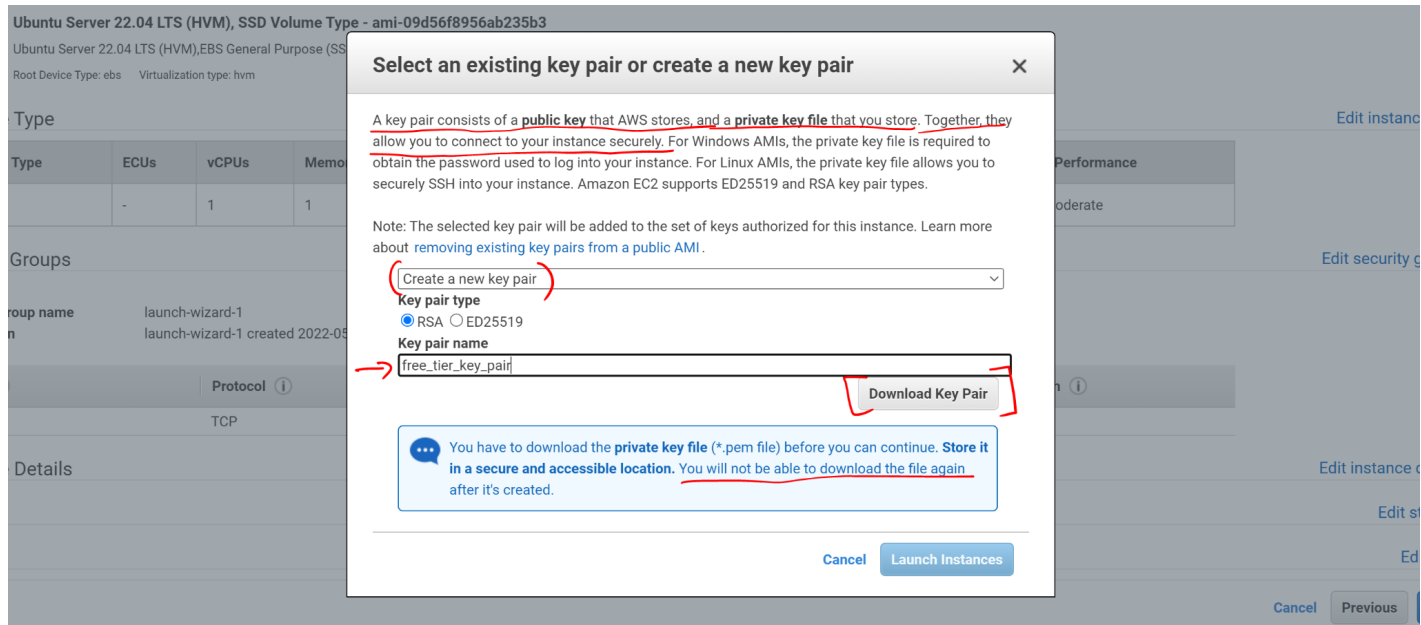
Description: launch-wizard-1 created 2022-05-02T14:01:35.783+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	

Cancel Previous **Launch**

(v) Create a Key Pair (don't ignore)

You will be greeted with a screen that will allow you to create a key pair. This step is **important**. A key pair is a file that is needed to connect to your AWS instance. You will be allowed to download the key pair only once. So download it now and store it safely. This is an additional layer of security that AWS imposes.



Clicking 'Download Key Pair' will download a `.pem` file. In this case it will be called **'free_tier_key_pair.pem'**. Let's keep this safe. Once downloaded, click **'Launch Instances'**.

Wait for a few seconds and the instance will be launched.

You then will be able to see a live instance by clicking the instances button on the EC2 dashboard. We will be launching our flask app in this instance.

Instances (1/1) Info

Instance state = running

Clear filters

Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
i-0c354ee0a63157d1b	Running	t2.micro	Initializing	No alarms	us-east-1b	[Redacted]	[Redacted]

Public URL to access the app (once we have launched the app)

Instance: i-0c354ee0a63157d1b

Details Security Networking Storage Status checks Monitoring Tags

Instance summary Info

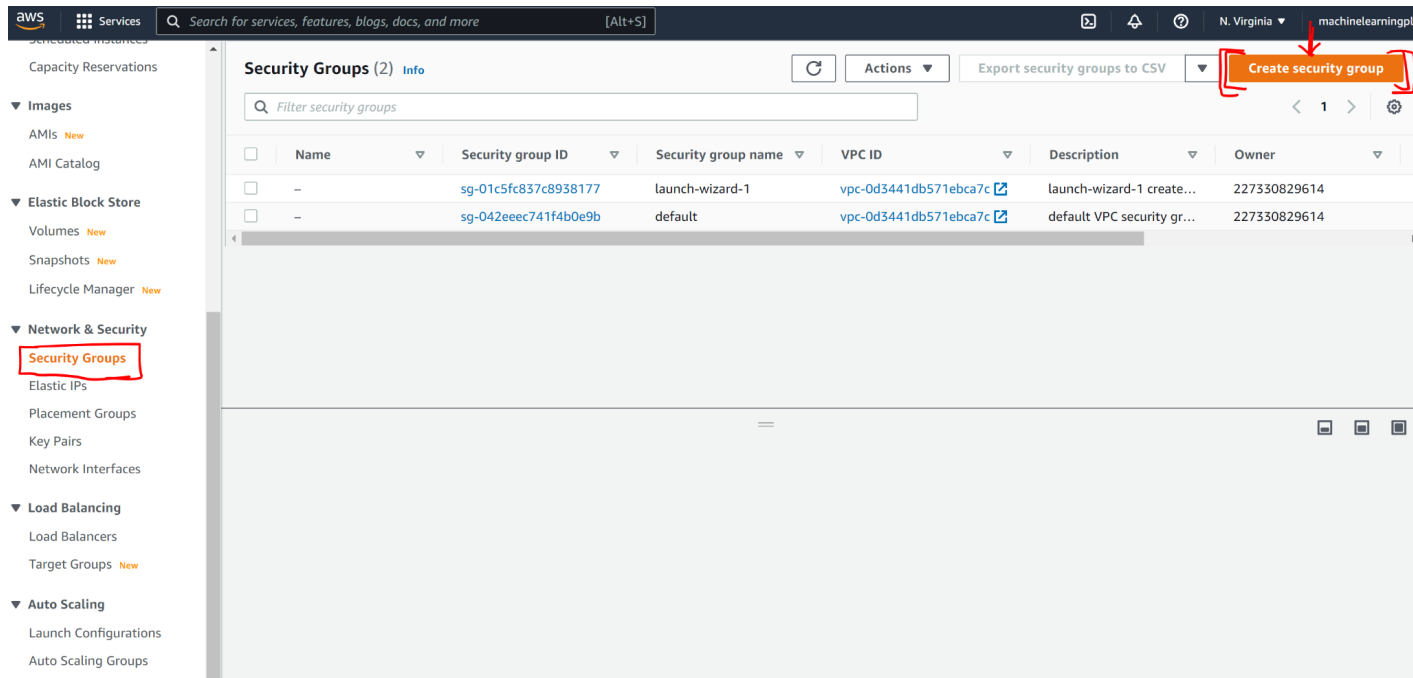
Instance ID	Public IPv4 address	Private IPv4 addresses
i-0c354ee0a63157d1b	[Redacted] open address	172.31.23.8
IPv6 address	Instance state	Public IPv4 DNS
-	Running	[Redacted] compute-1.amazonaws.com open address
Hostname type	Private IP DNS name (IPv4 only)	Answer private resource DNS name
IP name: ip-172-31-23-8.ec2.internal	ip-172-31-23-8.ec2.internal	-

Next, Let's create a security group.

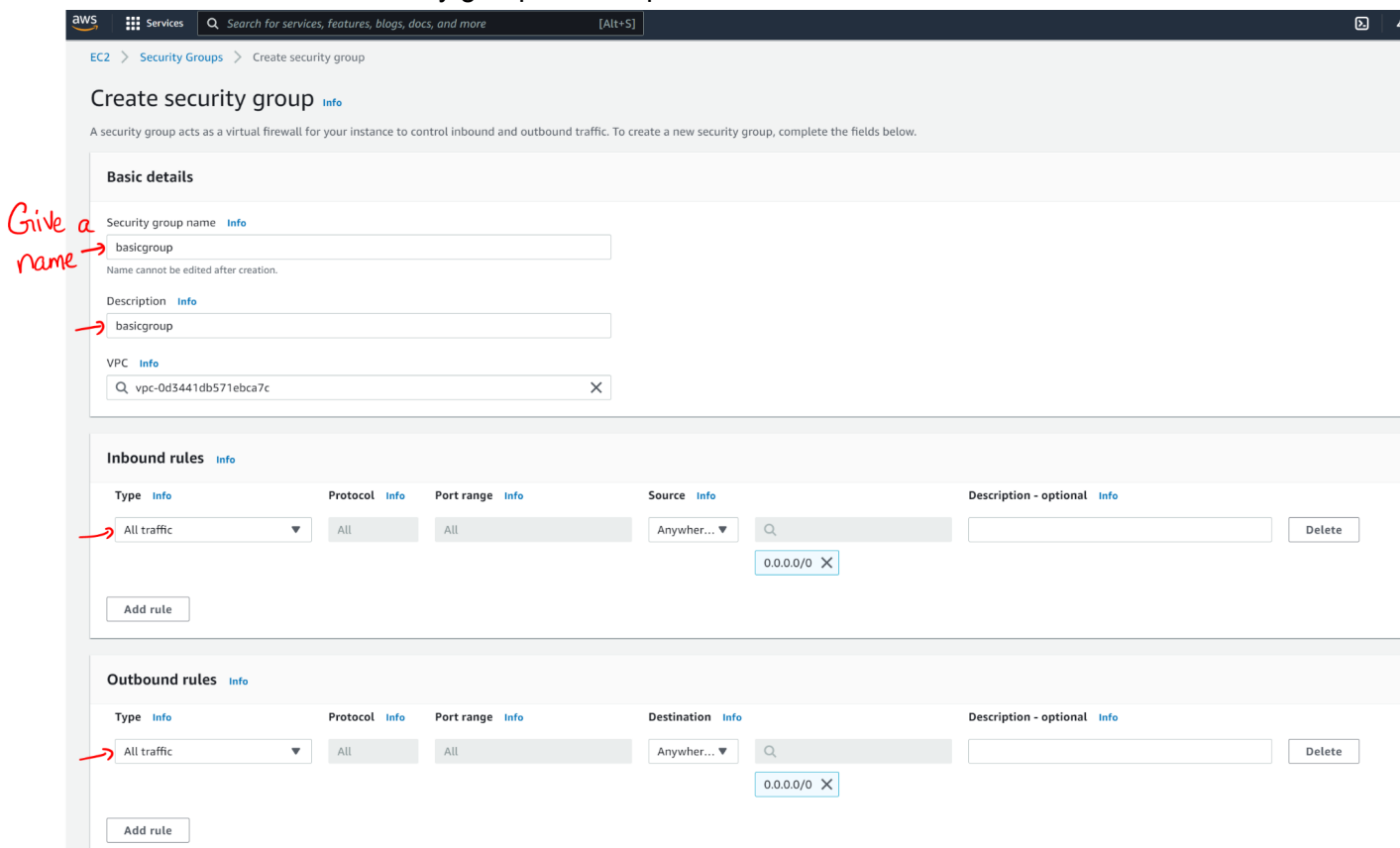
(vi) Create a Security Group

A security group lets us control who can send requests to the server (instance).

Under 'Network and Security' tab → Select 'Security Groups'. Then Click 'Create Security Group' to create one.



Give a name to the security group and keep it default.



Now, we need to change the security group for the instance to the new group we just created.

To do that, go to “Network and Security” → “Network Interfaces” → Right click on the instance and select “Change security groups”.

The screenshot displays the AWS Management Console interface for managing network resources. On the left-hand side, the navigation pane is visible, showing various AWS services categorized under 'Network & Security'. The 'Network Interfaces' option is highlighted with a red box and labeled 'Select Network Interfaces' in red text. The main content area shows the 'Network interfaces (1/1)' page. A table lists the available network interfaces, with the following details:

✓	Name	Network interface ID	Subnet ID	VPC ID	Availability Zone
✓	-	eni-0cb74421652cbb39f	subnet-0bd3ff0213ab36f3f	vpc-0d3441db571ebca7c	us-east-1b

A right-click context menu is open over the network interface, listing various actions. The 'Change security groups' option is highlighted with a red box. Below the table, the details for the selected network interface 'eni-0cb74421652cbb39f' are shown, including tabs for 'Details', 'Flow logs', and 'Tags'.

Then select the group we just created (*basicgroup*) and hit Save.

The screenshot shows the AWS Management Console interface for changing security groups on a network interface. The breadcrumb trail is: EC2 > Network Interfaces > eni-0cb74421652cbb39f > Change security groups. The main heading is 'Change security groups' with an 'Info' link. Below this is a description: 'Amazon EC2 evaluates all the rules of the selected security groups to control inbound and outbound traffic to and from your instance. You can use this window to add and remove security groups.'

The 'Network interface details' section shows the Network interface ID as eni-0cb74421652cbb39f.

The 'Associated security groups' section has a sub-header 'Associated security groups' and a description: 'Add one or more security groups to the network interface. You can also remove security groups.'

There is a search bar labeled 'Select security groups' with a magnifying glass icon. Below it is a list of security groups:

- launch-wizard-1 (sg-01c5fc837c8938177)
- launch-wizard-1
- default (sg-042eeec741f4b0e9b)
- default
- basicgroup (sg-08a4ede2e5cf9a295)
- basicgroup

A red arrow points to the 'basicgroup' entry. Below the list is a table with two columns: the first column contains 'launch-wizard-1' and the second column contains 'sg-01c5fc837c8938177'. To the right of the table is a 'Remove' button.


At the bottom right, there are 'Cancel' and 'Save' buttons. The 'Save' button is highlighted with a red box.

We are now all set to connect to the EC2 instance.

Step 3. Connect to AWS EC2 instance using ssh

The screenshot shows the AWS Management Console interface for the EC2 service. On the left is a navigation sidebar with options like 'EC2 Dashboard', 'Events', 'Tags', 'Limits', and 'Instances'. The main area displays a table of instances. One instance, 'i-0c354ee0a63157d1b', is in a 'Running' state. A context menu is open over this instance, listing actions such as 'Launch instances', 'Stop instance', 'Start instance', and 'Connect'. The 'Connect' option is highlighted with a red box. A red handwritten note next to it says 'Takes to a screen that gives diff options to connect'. Below the table, the 'Instance: i-0c354ee0a63157d1b' details are visible, including tabs for 'Details', 'Security', 'Networking', 'Storage', 'Status checks', and 'Monitor and troubleshoot'.

The following screen shows the instructions on how to connect to the ubuntu AWS EC2 instance from your local computer.

 Services [Alt+S]

EC2 > Instances > i-0c354ee0a63157d1b > Connect to instance

Connect to instance [Info](#)

Connect to your instance i-0c354ee0a63157d1b using any of these options


EC2 Instance Connect



Session Manager

SSH client


EC2 Serial Console


Instance ID

 i-0c354ee0a63157d1b

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is free_tier_key_pair.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 `chmod 400 free_tier_key_pair.pem`
4. Connect to your instance using its Public DNS:
 `ec2-3-80-25-198.compute-1.amazonaws.com`

Example:

 `ssh -i "free_tier_key_pair.pem" ubuntu@ec2-3-80-25-198.compute-1.amazonaws.com`

 **Note:** In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel

Let's start connecting.

I am on a Windows computer. The procedure is very similar on a linux/mac system as well. To make the connection with the remote AWS EC2 instance, `cd` to the folder that contains the Key Pair file, which in this case is `free_tier_key_pair.pem`.

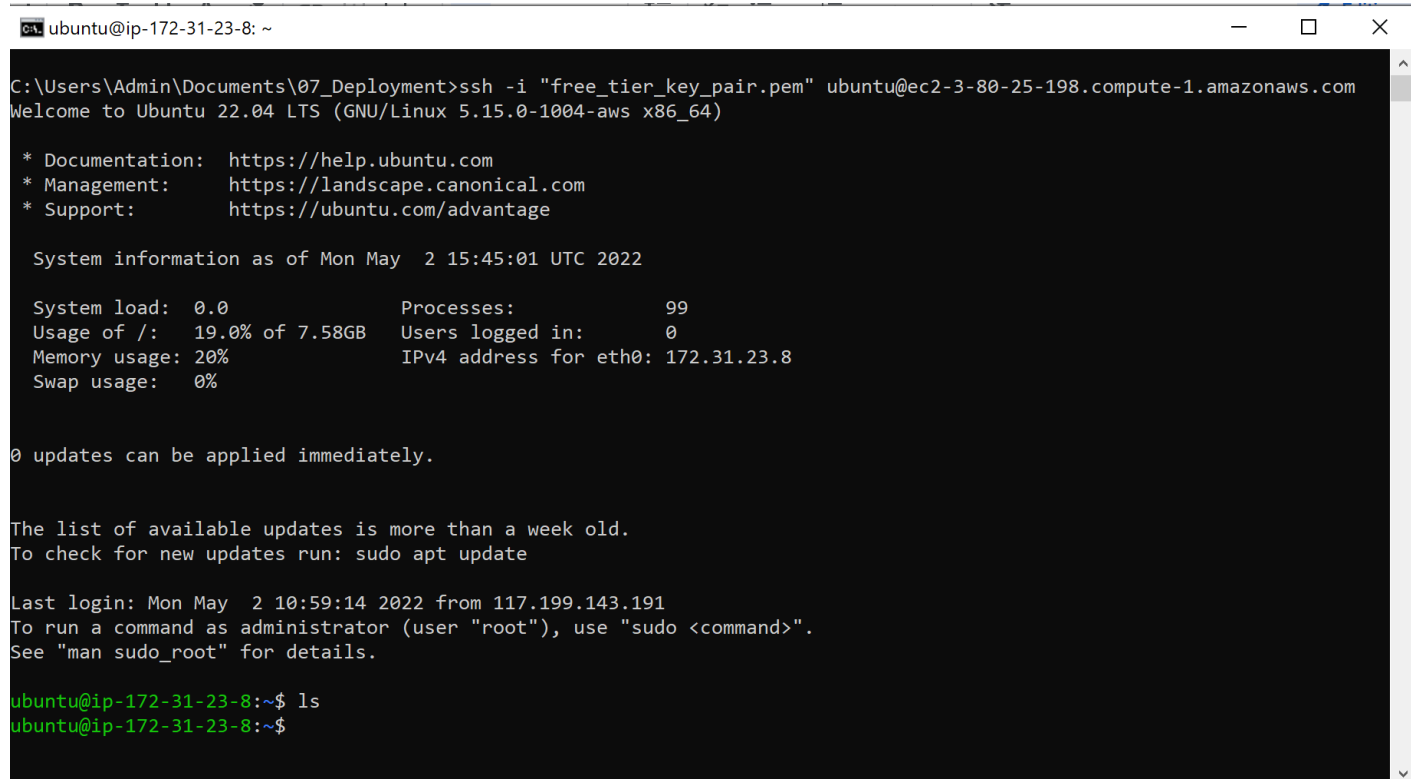
Then, as shown in the screen, type the following command from your command prompt if you are on windows, or the Terminal if you are on a Mac or Linux computer.

```
ssh -i "free_tier_key_pair.pem" ubuntu@ec2-3-80-25-198.compute-1.amazonaws.com
```

In the above command, you will need to replace the name of the `pem file` and the ec2 instance url.

The default username however in most cases is `ubuntu` so you can keep the `ubuntu@ec2..` part as it is.

This will make an `ssh` connection to the AWS EC2 instance. `ssh` stands for secure shell.

A screenshot of a Windows terminal window titled 'ubuntu@ip-172-31-23-8: ~'. The terminal shows the command 'ssh -i "free_tier_key_pair.pem" ubuntu@ec2-3-80-25-198.compute-1.amazonaws.com' being executed. The output shows the Ubuntu login banner, system information, and a prompt to run 'ls'.

```
ubuntu@ip-172-31-23-8: ~  
C:\Users\Admin\Documents\07_Deployment>ssh -i "free_tier_key_pair.pem" ubuntu@ec2-3-80-25-198.compute-1.amazonaws.com  
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-1004-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
System information as of Mon May  2 15:45:01 UTC 2022  
  
System load:  0.0           Processes:            99  
Usage of /:   19.0% of 7.58GB Users logged in:       0  
Memory usage: 20%          IPv4 address for eth0: 172.31.23.8  
Swap usage:   0%  
  
0 updates can be applied immediately.  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
Last login: Mon May  2 10:59:14 2022 from 117.199.143.191  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-23-8:~$ ls  
ubuntu@ip-172-31-23-8:~$
```

We are now connected to our AWS Ubuntu terminal. Now you can start typing the ubuntu shell commands from right here in your Windows machine.

Next, we need to copy the project files from local computer to the remote Ubuntu machine that we've leased.

Step 4. Move your files to AWS Ec2 using Secure Copy (scp)

Let's move the project folder to AWS.

To do this, you need to be in the Windows command prompt. So open a new command prompt and cd to the folder that contains the project directory and issue the following command to secure copy the files to EC2 instance.

```
scp -r -i "free_tier_key_pair.pem" ./flask_classification ubuntu@ec2-3-80-25-198.compute-1.amazonaws.com:~/
```

Again you will need to update the path to pem file and the ec2 url.

```
ubuntu@ip-172-31-23-8: ~
C:\Users\Admin\Documents\07_Deployment>scp -r -i "free_tier_key_pair.pem" ./flask_classification ubuntu@ec2-3-80-25-198.compute-1.amazonaws.com:~/
app.py                                100% 1387      4.4KB/s   00:00
catboost_training.json                100% 401        0.9KB/s   00:00
events.out.tfevents                  100% 106         0.3KB/s   00:00
learn_error.tsv                      100% 43          0.1KB/s   00:00
time_left.tsv                        100% 40          0.1KB/s   00:00
create_model.py                      100% 813        2.6KB/s   00:00
german_creditrisk_data.csv            100% 46KB       72.4KB/s   00:00
ml_model.pkl                         100% 7594       23.8KB/s   00:00
base-checkpoint.html                 100% 1496        1.8KB/s   00:00
index-checkpoint.html                100% 4782        8.3KB/s   00:00
base.html                            100% 1496        5.4KB/s   00:00
index.html                           100% 4782       17.1KB/s   00:00
```

Again ssh to the remote instance, and check if you can find the files there.

```
ubuntu@ip-172-31-23-8: ~
C:\Users\Admin\Documents\07_Deployment>ssh -i "free_tier_key_pair.pem" ubuntu@ec2-3-80-25-198.compute-1.amazonaws.com
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-1004-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon May  2 16:08:24 UTC 2022

System load:  0.0               Processes:    100
Usage of /:   20.4% of 7.58GB   Users logged in:  0
Memory usage: 23%              IPv4 address for eth0: 172.31.23.8
Swap usage:   0%

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Mon May  2 15:45:02 2022 from 117.199.141.85
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-23-8:~$ ls
flask_classification
ubuntu@ip-172-31-23-8:~$
```

Great!

All the files have been copied. Let's install the packages and start the app.

Step 5. Install the necessary packages and run app.py to start the app

```
sudo apt-get update
```

```
sudo apt-get -y install python3-pip
```

```
pip3 install <each of the following packages>
```

Packages needed:

```
catboost  
flask  
scikit-learn
```

Useful Tips and Tricks

1. If you have trouble connecting putty to ec2 instance read answer by [Darius](#)
2. To keep the server running even after disconnecting ssh

Answer 1: `nohup python3 app.py &`

Answer 2: Use [screen](#)

3. Install OpenCV Python on ubuntu
`sudo apt-get install python3-opencv`