

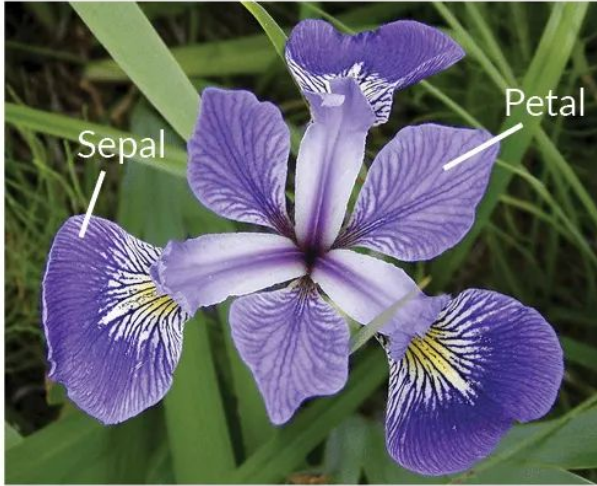
ML Model Deployment on AWS using Flask

Software Requirements

- VSCode
- Python (numpy, Flask, Scikit-learn)
- AWS Account

Goal

- Create the form a user can input values (features) to predict the type of flower.



Iris Versicolor



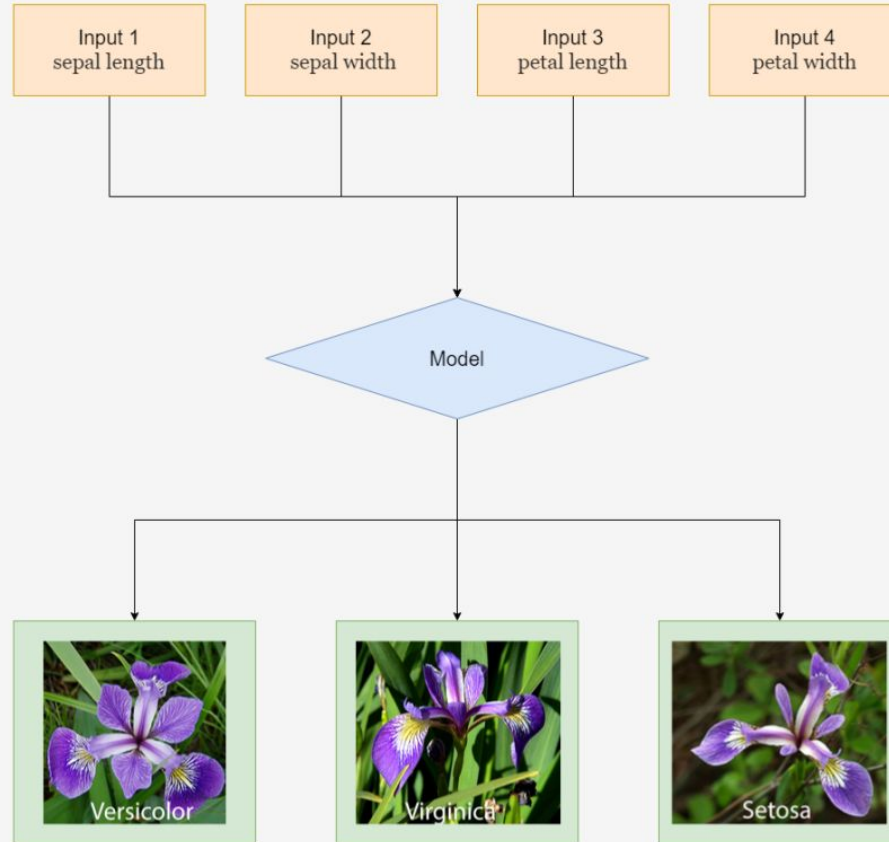
Iris Setosa



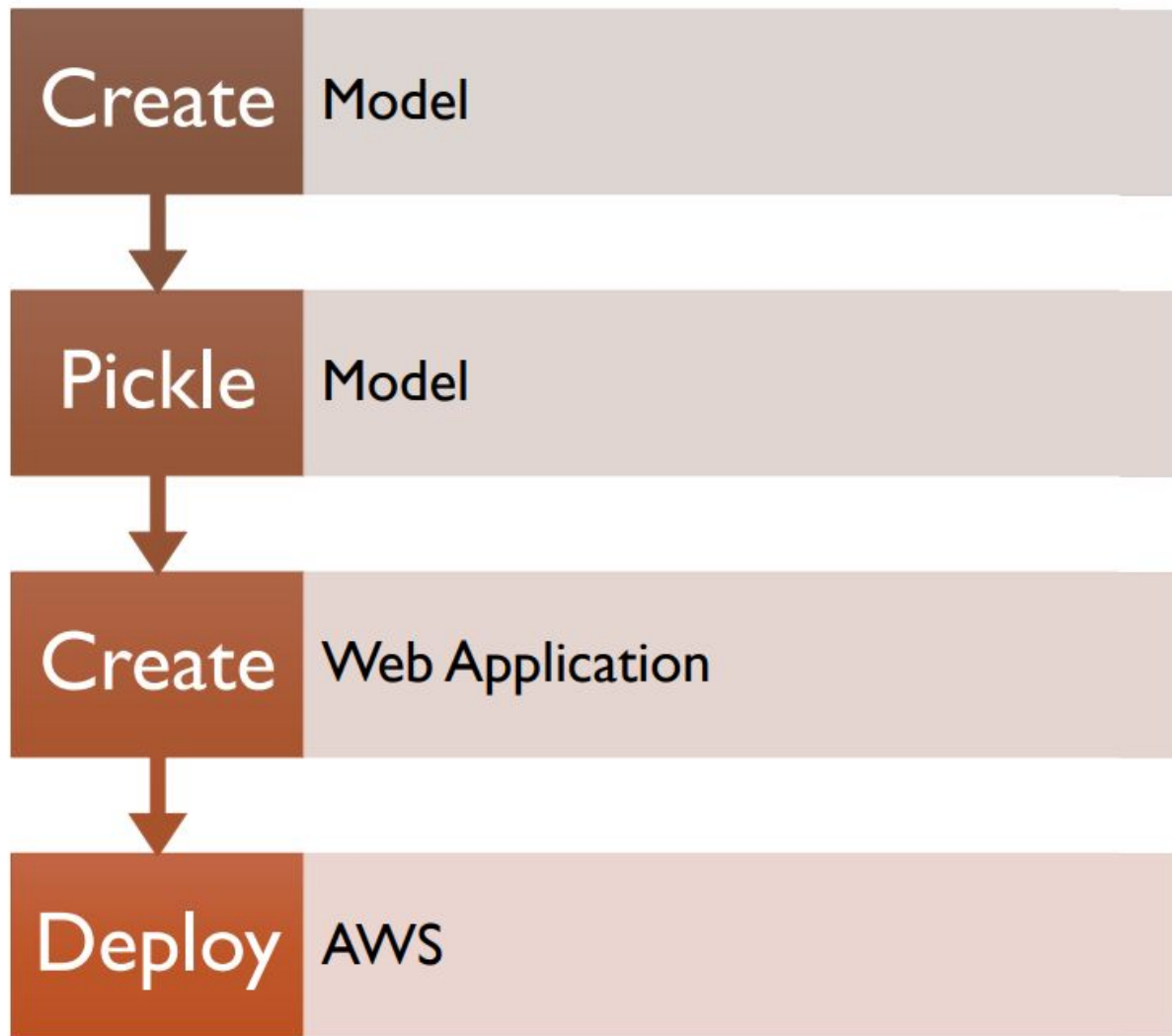
Iris Virginica

Process

Web Application



Deployment steps



1. Create model

- Create model_iris.py
- Import necessary libraries

```
from sklearn.metrics import accuracy_score
import numpy as np
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
```

- Load iris dataset

```
dataset=load_iris()
```

1. Create model (model_iris.py)

- Train machine learning model

```
# Getting Feature Names
names=dataset.feature_names

# Loading features and labels from the dataset
features=dataset.data
labels=dataset.target

# Splitting labels and features to training and testing sets
feature_train,feature_test,label_train,label_test = \
train_test_split(features,labels,test_size=0.2,random_state=42)

# Initialising Logistic Regression Model with max iterations=500
model=LogisticRegression(max_iter=500)

# Fitting Model with training features and labels
model.fit(feature_train,label_train)
```

- Test the model

```
#Predicting the labels for the testing features
label_pred=model.predict(feature_test)
#Finding the accuracy score
from sklearn.metrics import accuracy_score
accuracy_score(label_pred,label_test)
print(accuracy_score(label_pred, label_test))
```

Should get 1.0

May need to use pip to install all related libraries e.g. numpy, pandas, scikit-learn e.g.

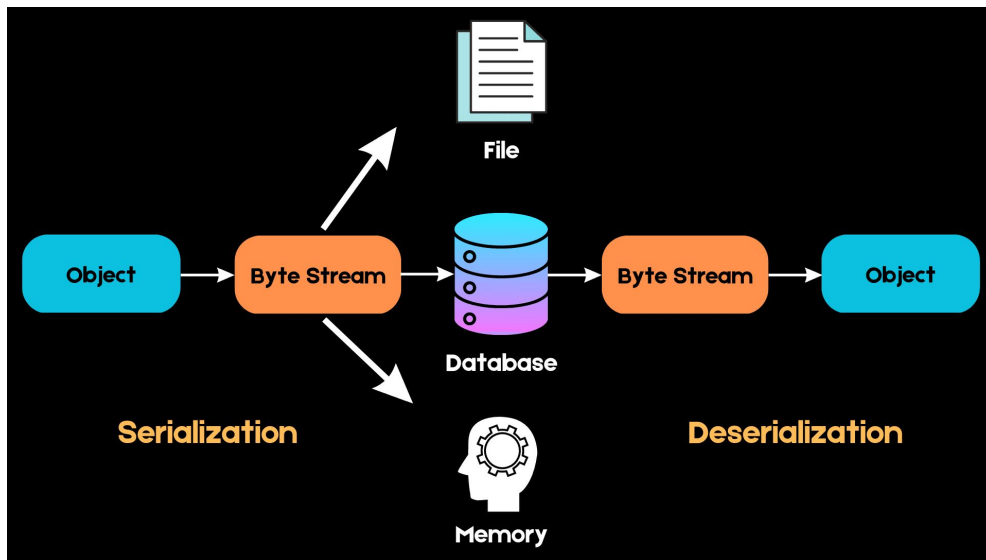
- pip install matplotlib
- pip install scikit-learn

2. Pickle model

Python Pickle

- A process of converting a Python object into a byte stream to store it in a file/database, **maintain program state** across sessions, or transport data over the network
- Don't have to re-run the script to obtain the Python object e.g. dataframe, dict.
- Pickle is a useful Python tool that allows you to save your ML models
- Save Python objects in file **model.pkl** with Pickle Dump function

```
import pickle
pickle.dump(model, open('model.pkl', 'wb+'))
```



<https://www.datacamp.com/tutorial/pickle-python-tutorial>

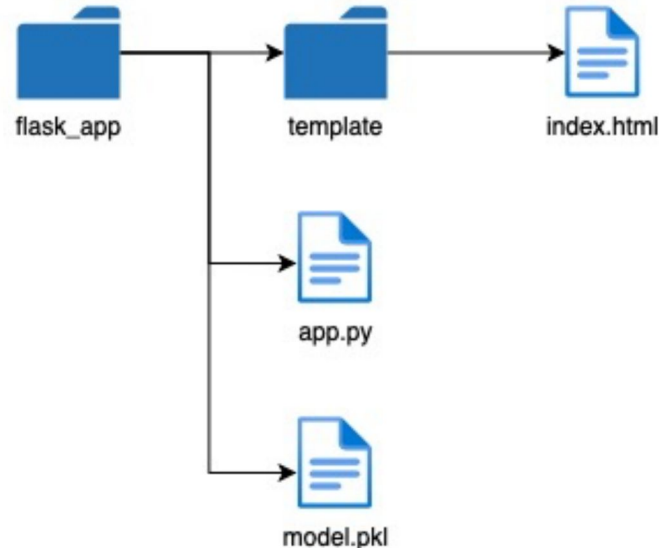
3. Create Web Application with Flask



Flask

Flask

- Lightweight Web development framework using Python
- Integrated support for unit testing
- Flexible

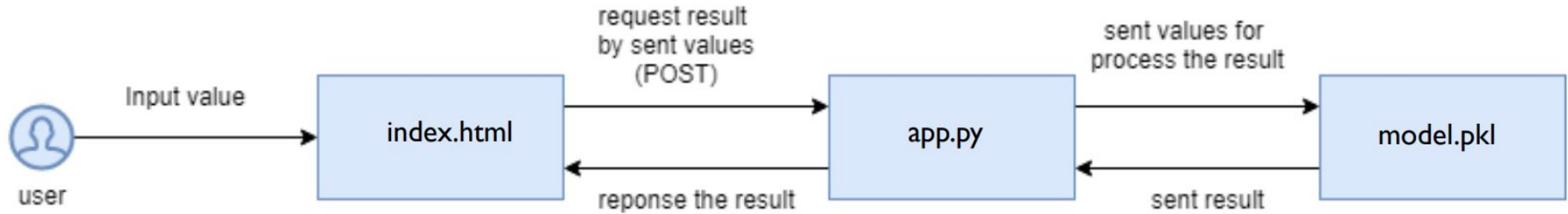


3. Create Web Application with Flask: UI

Iris Classifier

It is Iris Virginica

3. Create Web Application with Flask: Workflow



HTTP POST: submits data to be processed to the identified resource. The data is included in the body of the request. This may result in the creation of a new resource or the updates of existing resources or both.

3. Create Web Application with Flask: App.py

- Import libraries

Import Flask class

```
from flask import Flask, render_template, request, send_file, send_from_directory, jsonify
import pickle
import joblib
import numpy
```

- Create an instance of flask class. `__name__` is the shortcut of the name of the module, `template_folder` identifies the folder that contains related files.

```
app = Flask(__name__, template_folder='templates')
```

- Load model

```
model = joblib.load('model.pkl')
```

3. Create Web Application with Flask: App.py

- Route '/' for main page

```
@ app.route('/', methods=['POST', 'GET'])
def main():
    if request.method == 'GET':
        return render_template('index.html')
```

- Initial host with port 8080

```
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080)
```

- Route '/predict' for prediction

```
@ app.route('/predict', methods=['POST', 'GET'])
def predict():
    if request.method == 'GET':
        return render_template('index.html')
    if request.method == 'POST':
        features = [float(x) for x in request.form.values()]
        print(features)
        labels = model.predict([features])
        species = labels[0]
        if species == 0:
            s = "It is Iris Setosa"
        elif species == 1:
            s = "It is Iris VersiColor"
        else:
            s = "It is Iris Virginica"
        return s
```

3. Create Web Application with Flask: index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Iris Classifier</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  </head>
  <body>
    <h1><CENTER>Iris Classifier</CENTER></h1>
    <form id="myform" method="POST">
      <input type="text" name="sl" placeholder="Enter Sepal Length in cm"><br><br>
      <input type="text" name="sw" placeholder="Enter Sepal Width in cm"><br><br>
      <input type="text" name="pl" placeholder="Enter Petal Length in cm"><br><br>
      <input type="text" name="pw" placeholder="Enter Petal Width in cm"><br><br>
    </form>
    <button id="predict">Predict</button>
    <h2 id="result"></h2>
  </body>
</html>
```

Import ajax

Input form, button and results

3. Create Web Application with Flask: index.html

```
<script type="text/javascript">
$(function() {
  $('#predict').click(function() {
    event.preventDefault();
    var form_data = new FormData($('#myform')[0]);
    console.log(form_data);
    $.ajax({
      type: 'POST',
      url: '/predict',
      data: form_data,
      contentType: false,
      processData: false,
    })
    .done(function(data, textStatus, jqXHR){$('#result').text(data)})
    .fail(function(data){alert('error!')})
  })
})
</script>
```

Additional ajax part

3. Create Web Application with Flask: test run app.py

- You can test run app by type command in terminal in flask_app folder `python3 app.py`

```
PS C:\Users\USER\Desktop\iris> python3 app.py
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:8080
* Running on http://192.168.1.113:8080
Press CTRL+C to quit
127.0.0.1 - - [27/Oct/2023 15:46:05] "GET / HTTP/1.1" 200 -
[10.0, 20.0, 15.0, 7.0]
127.0.0.1 - - [27/Oct/2023 15:46:31] "POST /predict HTTP/1.1" 200 -
```

To install flask, type 'pip install flask'



Iris Classifier

Predict

It is Iris Virginica

4. Deploy to *AWS* EC2

Using Amazon EC2 IaaS

- Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud.
- It is designed to make web-scale cloud computing easier for developers.
- Benefits
 - simple web service interface allows you to obtain and configure capacity with minimal friction
 - reduce the time required to obtain and boot new server instances to minutes
 - quickly scale capacity, both up and down, as your computing requirements change
 - Pay for what you use



Amazon EC2

Getting Started with Amazon EC2



Getting Started with the AWS Management Console

1. Set up and log into your AWS account
2. Launch an Amazon EC2 instance
3. Configure your instance e.g. Security Group, elastic IP address
4. Connect to your instance
5. Install Flask, Numpy, Scikit-learn libraries
6. Run Flask
7. Once everything is done, terminate Instances: EC2 is free to start but you should terminate your instances to prevent additional charges.

AWS Academy Learner Lab



AWS Academy <notifications@instructure.com>
to me ▾

Thu, Aug 5, 3:00 PM (10 days ago)

You've been invited to participate in the AWS Academy Associate Services [5076]. Click here to accept this request.

Name: **Jim**

Email: jim.chantri@gmail.com



Welcome Aboard!

You've been invited to participate in the AWS Academy Associate Services [5076]. Click here to accept this request you account.

awsacademy.com/LMS_Login

CS-Dept Sem2/61 Sem1/62 Sem 2/62 Sem 1/63 ผู้ช่วยคณบดีฝ่ายสาร...



Student Login

(For students enrolled in a class)


学生の方はこちらからログインしてください。

已注册课程的学生请在这里登录

ALLv1-24645

Dashboard

Published Courses (20)



[AIT AT83.03 Y2022](#)
[ALLv1-24645](#)

Account


Dashboard

Courses

Calendar

Inbox

Get Started

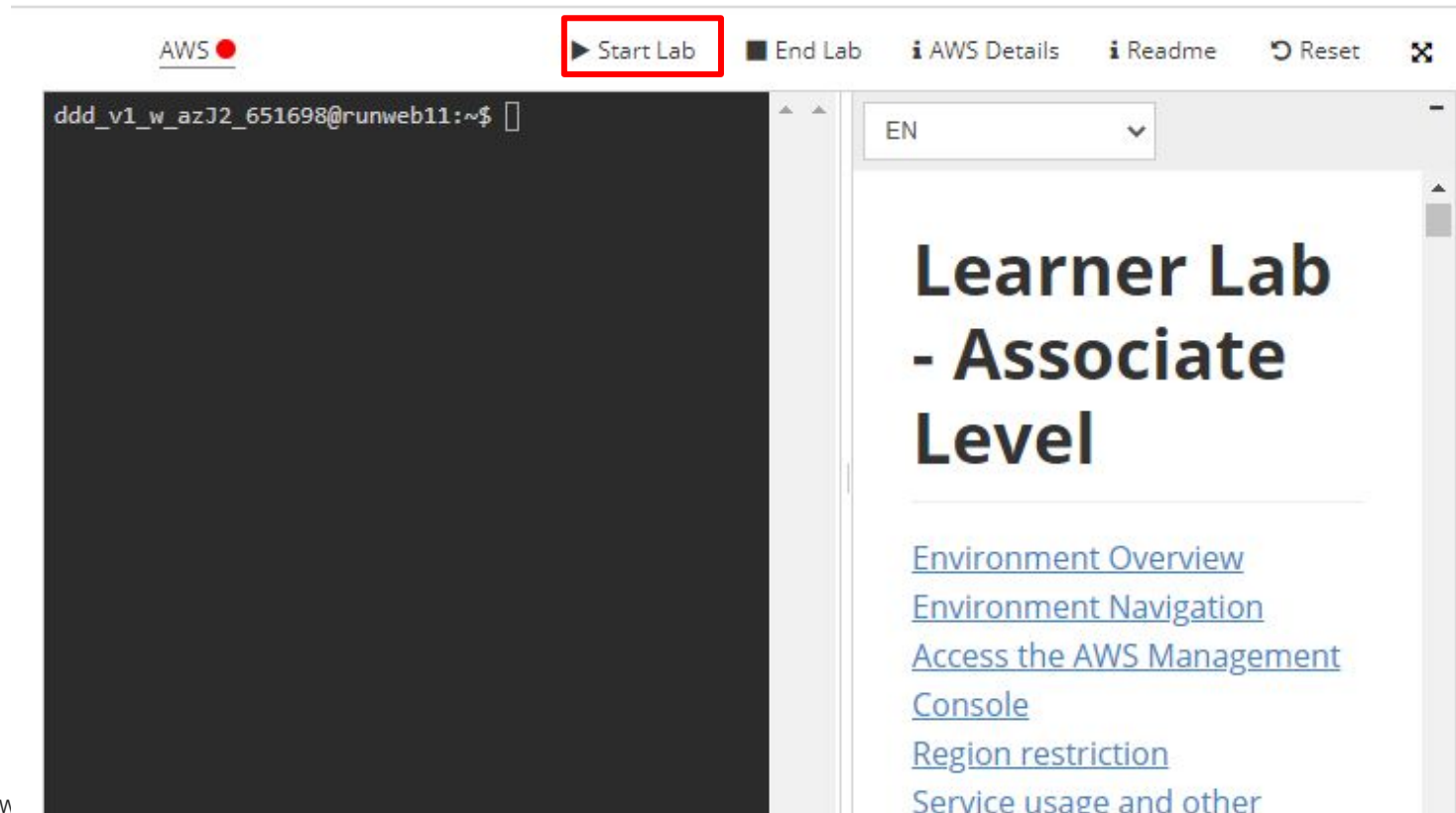



Select [Modules](#) to start the course.







▼ Learner Lab Associate Services

- [Learner Lab - Associate Services](#)
- [End of Course Feedback Survey](#)

Learner Lab - Associate Services



AWS 

 Start Lab  End Lab  AWS Details  Readme  Reset 

```
ddd_v1_w_azJ2_651698@runweb11:~$
```

EN

Learner Lab - Associate Level

- [Environment Overview](#)
- [Environment Navigation](#)
- [Access the AWS Management Console](#)
- [Region restriction](#)
- [Service usage and other](#)

Learner Lab - Associate Services



Go to AWS Management console

Detail of your cloud usage

AWS

03:57

▶ Start Lab

■ End Lab

i AWS Details

i Readme

↺ Reset



Used \$0 of \$100

```
ddd_v1_w_azJ2_651698@runweb11:~$
```

Close

Region: us-east-1

Lab ID: arn:aws:cloudformation:us-east-1:495477404114:stack/c37469a482054l936274fde0-11eb-999b-12a48490849f

Creation Time: 2021-08-15T08:47:28-0700

Start session at: 2021-08-15T08:47:29-0700

Remaining session time: 04:00:00(240 minutes)

AWS Management Console



AWS Management Console

Since AWS keeps updating their interface, the console may look a little different.

AWS services

Find Services

You can enter names, keywords or acronyms.

Example: Relational Database Service, database, RDS

▼ All services



Compute

EC2

Lightsail

ECR

ECS

EKS

Lambda

Batch

Elastic Beanstalk

Serverless Application
Repository



Management & Governance

AWS Organizations

CloudWatch

AWS Auto Scaling

CloudFormation

CloudTrail

Config

OpsWorks

Service Catalog

Systems Manager



AWS Cost Management

AWS Cost Explorer

AWS Budgets

AWS Marketplace
Subscriptions



Mobile

AWS Amplify

Mobile Hub

AWS AppSync

Access resources on the go



Access the Management Console using the AWS
Console Mobile App. [Learn more](#)

Explore AWS

Amazon RDS

Set up, operate, and scale your relational database in the
cloud. [Learn more](#)

Data Lake Storage

Build your data lake on the most secure, durable, and
scalable storage. [Learn more](#)

EC2 Dashboard: Launch Instance

The screenshot shows the AWS Management Console interface for the EC2 service in the N. Virginia region. The top navigation bar includes the AWS logo, 'Services', a search bar, and the region 'N. Virginia' which is highlighted with a red box. The left sidebar contains navigation links for 'New EC2 Experience', 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', 'Instances' (expanded), 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Scheduled Instances', 'Capacity Reservations', and 'Images'. The main content area displays a summary of EC2 resources: Instances (running) 0, Elastic IPs 0, Key pairs 1, Placement groups 0, Snapshots 0, Dedicated Hosts 0, Instances 0, Load balancers 0, Security groups 1, and Volumes 0. A blue informational box states: 'Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more'. Below this, the 'Launch instance' section provides a description: 'To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.' and features a prominent orange 'Launch instance' button with a dropdown arrow, which is pointed to by a red arrow. To the right of this button is the 'Service health' section with a refresh icon and a link to the 'AWS Health Dashboard'. The right sidebar lists 'Support resources' (VPC, Default VPC, Settings, EBS encryption, Zones, EC2 Serial, Default console) and 'Explore' (Amazon GuardDuty, detection workloads).

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

New EC2 Experience Tell us what you think

EC2 Dashboard

EC2 Global View

Events

Tags

Limits

▼ **Instances**

Instances **New**

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances **New**

Dedicated Hosts

Scheduled Instances

Capacity Reservations

▼ **Images**

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

Instances (running)	0	Dedicated Hosts	0
Elastic IPs	0	Instances	0
Key pairs	1	Load balancers	0
Placement groups	0	Security groups	1
Snapshots	0	Volumes	0

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance ▼

Service health

Refresh AWS Health Dashboard

Region

Launch an EC2 instance



EC2 > Instances > Launch an instance

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name

My Server

[Add additional tags](#)

Recents Quick Start

Amazon Linux
aws


macOS
Mac

Ubuntu
ubuntu

Windows
Microsoft

Red Hat
Red Hat

SUSE Li
SUSE


Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Description

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-09-19

An Amazon Machine Image (AMI) is a template that contains a software configuration (for example, an operating system, an application server, and applications). It serves as the basic unit of deployment for services delivered using EC2.

Configure Instance: Choose Resource Size (CPU, Memory)



▼ Instance type [Info](#)

Instance type

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory

On-Demand Linux pricing: 0.0116 USD per Hour

On-Demand Windows pricing: 0.0162 USD per Hour



[Compare instance types](#)

Choose t2.micro

Key pair (for login)

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Select



Create new key pair

Key pair (for login)

Key pair name

JimAITVirginia

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type



RSA

RSA encrypted private and public key pair



ED25519

ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format



.pem

For use with OpenSSH



.ppk

For use with PuTTY

- Don't lose it
- Specific to a region
- Save .pem file

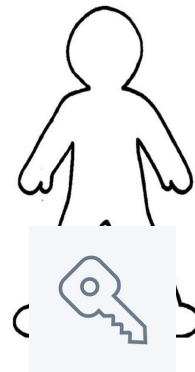
Cancel

Create key pair

Key pair



- Stored at EC2
- Public



- Stored at client
- Keep secret

2048-bit SSH-2 RSA

Network settings and Security Group



▼ Network settings [Get guidance](#)

[Edit](#)

Network [Info](#)

vpc-095151d43658e17bd

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group

☐ Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

☒ Allow SSH traffic from
Helps you connect to your instance

Anywhere
0.0.0.0/0

☐ Allow HTTPs traffic from the internet
To set up an endpoint, for example when creating a web server

☐ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

We'll add TCP with
port 8080 later.

Configure security group

- Allow TCP port 8080 for inbound rule

☒ Create security group ☐ Select existing security group

Security group name - *required*

My Security Group

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!\$*

Description - *required* [Info](#)

This security group allows SSH and Custom TCP port 8080

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Remove

Type [Info](#)

ssh ▼

Protocol [Info](#)

TCP

Port range [Info](#)

22

Source type [Info](#)

Anywhere ▼

Source [Info](#)

Q Add CIDR, prefix list or security

0.0.0.0/0 X

Description - *optional* [Info](#)

e.g. SSH for admin desktop

▼ Security group rule 2 (TCP, 8080, 0.0.0.0/0)

Remove

Type [Info](#)

Custom TCP ▼

Protocol [Info](#)

TCP

Port range [Info](#)

8080

Source type [Info](#)

Anywhere ▼

Source [Info](#)

Q Add CIDR, prefix list or security

Description - *optional* [Info](#)

e.g. SSH for admin desktop

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Configure storage

▼ Configure storage [Info](#)

[Advanced](#)

1x GiB ▼ Root volume

 Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage



Add new volume

0 x File systems

[Edit](#)

Summary

▼ Summary

Number of instances [Info](#)

Storage (volumes)

1 volume(s) - 8 GiB



Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.



Cancel

Launch instance

How long does it take for AWS to provision your instance?



Launch Status



Your instances are now launching

The following instance launches have been initiated: [i-02ccbd890badca9d9](#) [View launch log](#)



Get notified of estimated charges

[Create billing alerts](#) to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click **View Instances** to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the Instances screen. [Find out](#) how to connect to you

The instance is ready



aws Services ▾ Resource Groups ▾

awsstudent @ 5119-5080-2600 ▾ Singapore ▾ Support ▾

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

Instances

Launch Templates

Spot Requests

Reserved Instances

Dedicated Hosts

Capacity Reservations

IMAGES

AMIs

Bundle Tasks

Launch Instance ▾ Connect Actions ▾

search : i-02ccbd890badca9d9 Add filter

	Name ▾	Instance ID ▴	Instance Type ▾	Availability Zone ▾	Instance State ▾	Status Checks ▾	Alarm Status	Public I
<input type="checkbox"/>	Webapp	i-02ccbd890badca9d9	t2.micro	ap-southeast-1b	running	Initializing	None	ec2-52-

Instance: **i-02ccbd890badca9d9 (Webapp)** Public DNS: ec2-52-221-192-38.ap-southeast-1.compute.amazonaws.com

Description Status Checks Monitoring Tags

Instance ID i-02ccbd890badca9d9 Public DNS (IPv4) ec2-52-221-192-38.ap-southeast-

Connect to your instance



EC2 Instance Connect

Session Manager


SSH client

EC2 serial console

Instance ID

 i-0d889e6ad723ce02b (My Server)

Public IP address

 54.209.194.135

User name

ec2-user

Connect using a custom user name, or use the default user name ec2-user for the AMI used to launch the instance.



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

Connect

Once your instance is running, you can login using SSH and a key pair

Ubuntu terminal



```
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1014-aws x86_64)

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

System information as of Fri Oct 27 15:18:31 UTC 2023

System load:  0.0703125      Processes:            101
Usage of /:   35.6% of 7.57GB Users logged in:           0
Memory usage: 20%           IPv4 address for eth0: 172.31.37.174
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

11 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri Oct 27 05:05:55 2023 from 18.206.107.28
ubuntu@ip-172-31-37-174:~$
```

EC2 Instance Connect



Session Manager

SSH client


EC2 serial console

Instance ID

 i-0240328d714b14bcf (iris2)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is JimAITVirginia.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 `chmod 400 "JimAITVirginia.pem"`
4. Connect to your instance using its Public DNS:
 `ec2-52-87-195-140.compute-1.amazonaws.com`

Example:

 `ssh -i "JimAITVirginia.pem" ec2-user@ec2-52-87-195-140.compute-1.amazonaws.com`

Install necessary libraries

From Ubuntu, update and install related libraries

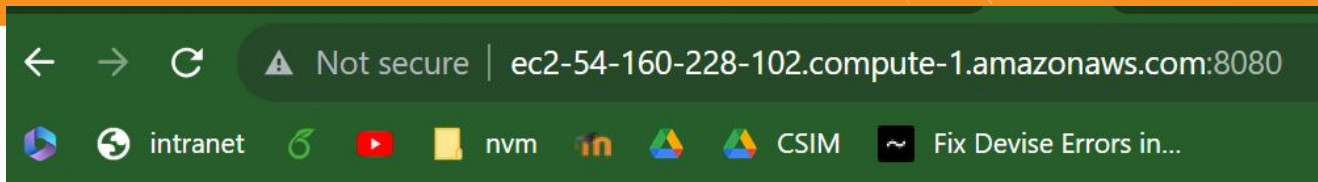
- `sudo su`
- `sudo update -y`
- `yum install python3-pip`
- `pip3 install numpy flask scikit-learn`

From your computer terminal, copy related files to Ubuntu

- `scp -r -i .\JimAITVirginia.pem .\iris_project\ ec2-user@ec2-52-87-195-140.compute-1.amazonaws.com:~/.`

From Amazon Linux, go to `iris_project` folder and run `app.py`

- `python3 app.py`
- From your web browser, type public DNS followed by 8080, e.g.
- `http://ec2-52-87-195-140.compute-1.amazonaws.com:8080/`



Iris Classifier

It is Iris Virginica

Addition tips



- To fix the IP address, use elastic IP address
- Use nohup to run app.py

Reference



- <https://medium.com/shapeai/deploying-flask-application-with-ml-models-on-aws-ec2-instance-3b9a1cec5e13>
- <https://aws.amazon.com/pm/ec2/>
- <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html>
- <https://flask.palletsprojects.com/en/3.0.x/>
- <https://www.kaggle.com/datasets/uciml/iris>