

Labs 5-9

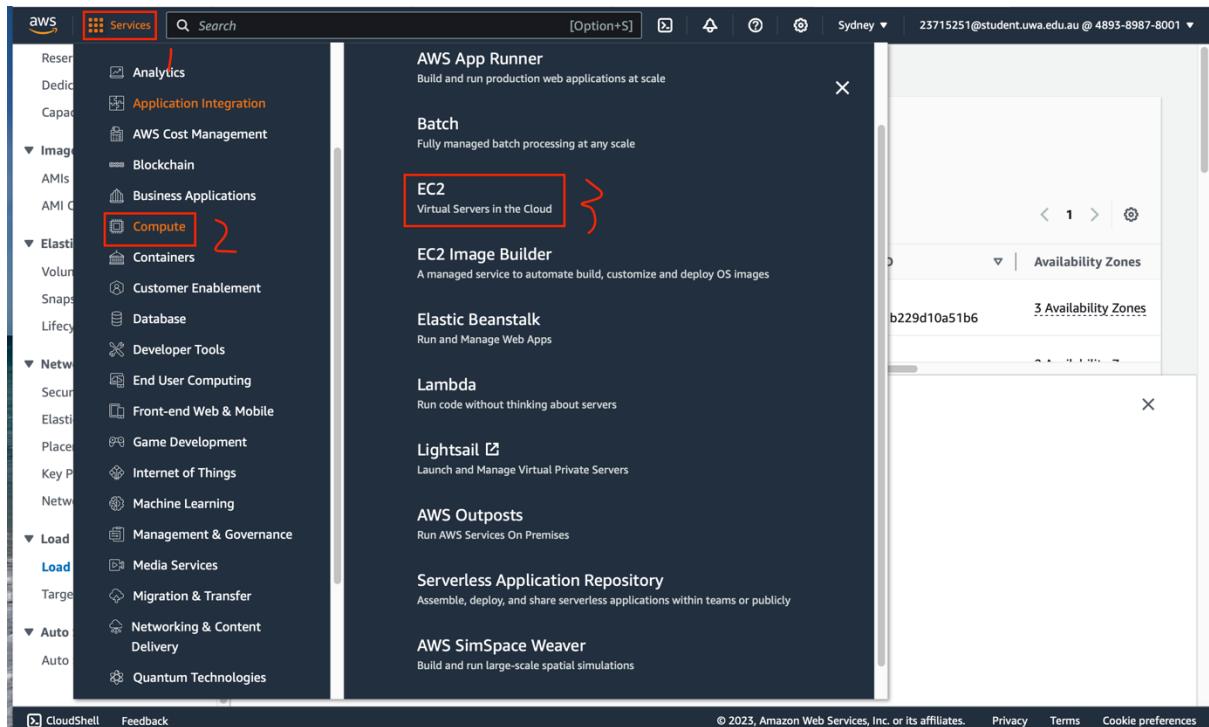
Name: Rongchuan Sun
Student Number: 23715251

Lab5 Networking

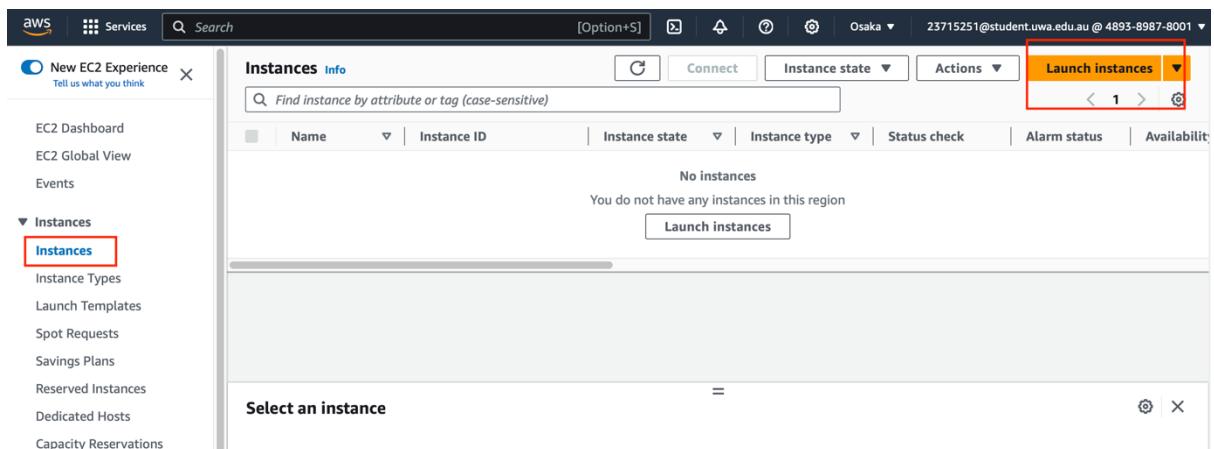
Setting up an Application Load Balance

[1] Create 2 EC2 instances in two different availability zones of a specific region.

We need to login AWS account. Click services, compute and EC2.



Click launch instance.



Create instance name

following the simple steps below.

Name and tags Info

Name Add additional tags

Choose ubuntu in quick start and choose free tier eligible for AMI.

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Recents Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE

 [Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type
ami-024e6faf93d85776 (64-bit (x86)) / ami-08ffd91d87f63bb09 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description
Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-05-16

Architecture **64-bit (x86)** AMI ID **ami-024e6faf93d85776** Verified provider

Choose instance type

▼ Instance type [Info](#)

Instance type

t2.micro

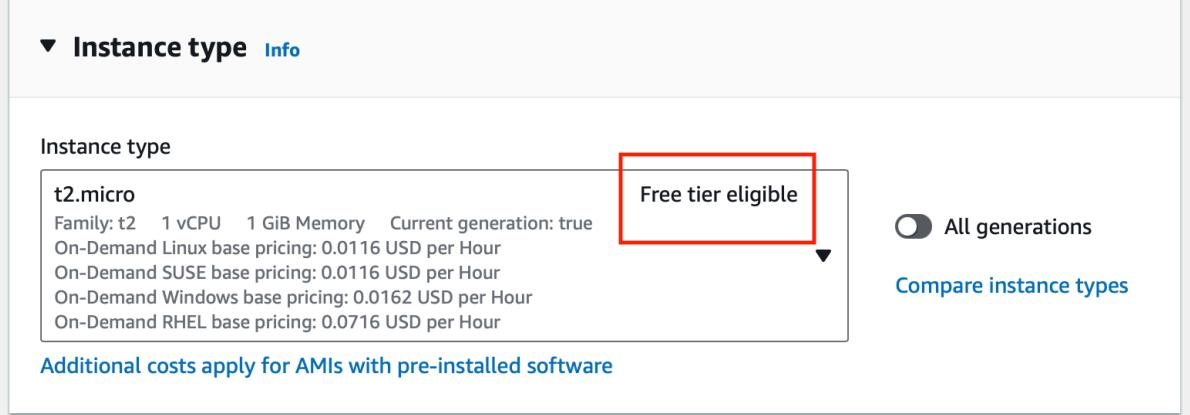
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0116 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand RHEL base pricing: 0.0716 USD per Hour

Free tier eligible

All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software



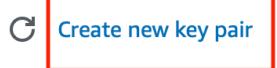
Create key pair

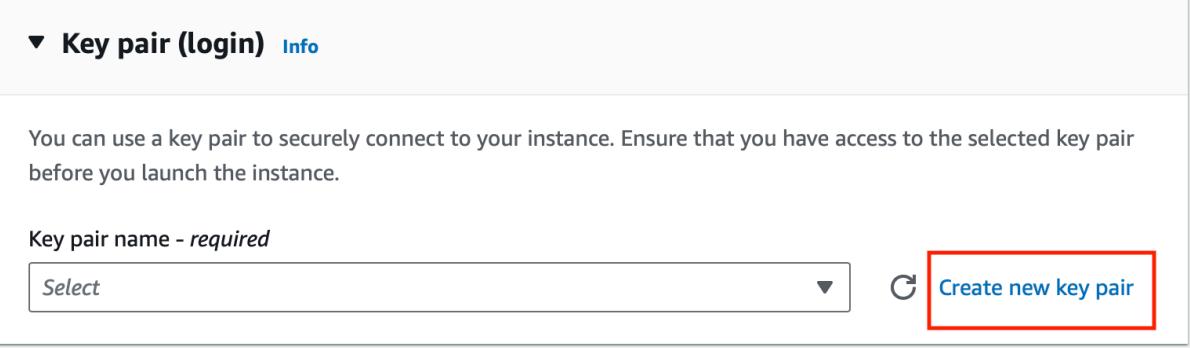
▼ Key pair (login) [Info](#)

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



Tap name and click create

Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.

The name can include up to 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

Private key file format

.pem
For use with OpenSSH

.ppk
For use with PuTTY

⚠️ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more ↗](#)

Cancel **Create key pair**

Choose the key pair created.

▼ 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*

[▼](#) **Create new key pair**

Network setting

Choose default VPC

Subnet is in us-east-2a

Public Ip is enable

Create security group named 23715251-lab5

▼ 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

23715251-lab5 [▼](#) [Create new key pair](#)

▼ Network settings [Info](#)

VPC - required [Info](#)

vpc-0d4dfbf0c6d88d5ae (default) [▼](#)
172.31.0.0/16 [vpc-0d4dfbf0c6d88d5ae](#)

Subnet Info

subnet-0b1d4d5f95cf2b531 VPC: vpc-0d4dfbf0c6d88d5ae Owner: 489389878001 Availability Zone: us-east-2a IP addresses available: 4091 CIDR: 172.31.0.0/20 [▼](#) [Create new subnet](#)

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

Security group name - required

launch-wizard-1 [▼](#)

▼ Network settings [Info](#)

VPC - required [Info](#)

vpc-0d4dfbf0c6d88d5ae (default) [▼](#)
172.31.0.0/16

Subnet Info

subnet-0b1d4d5f95cf2b531 VPC: vpc-0d4dfbf0c6d88d5ae Owner: 489389878001 Availability Zone: us-east-2a IP addresses available: 4083 CIDR: 172.31.0.0/20 [▼](#) [Create new subnet](#)

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

Security group name - required

23715251-lab5 [▼](#)

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](#)

23715251-lab5 created 2023-10-11T10:58:05.216Z [▼](#)

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0) [Remove](#)

▼ Summary

Number of instances [Info](#)

1 [▼](#)

Software Image (AMI)

Amazon Linux 2023 AMI 2023.2.2...[read more](#)
ami-036f5574583e16426

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

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

[Cancel](#) [Launch instance](#) [Review commands](#)

Click launch instance

Repeat above step to create another instance in different zone, that is us-east-2c.

They will be shown after finishing.

Instances (2/5) Info						
		C	Connect	Instance state ▾	Actions ▾	Launch instances ▾
<input type="text"/> Find Instance by attribute or tag (case-sensitive)						
	Name Edit	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>	23715251-us-east-2a	i-065d9ca3cf4ab7895	Stopped	t2.micro	-	User: arn:aws:iam::23715251:root
<input checked="" type="checkbox"/>	23715251-us-east-2a	i-070e9d19d487b2635	Running	t2.micro	2/2 checks passed	User: arn:aws:iam::23715251:root
<input checked="" type="checkbox"/>	23715251-us-east-2c	i-08e0f992f7179568c	Running	t2.micro	2/2 checks passed	User: arn:aws:iam::23715251:root

[2] Create the Application Load Balancer.

Click load balancers, and then click create button.

The screenshot shows the AWS EC2 Load Balancers page. On the left, there is a navigation sidebar with various services like Spot Requests, Savings Plans, Reserved Instances, etc. Under the 'Load Balancing' section, the 'Load Balancers' option is selected and highlighted with a red box. The main content area shows a table titled 'Load balancers' with one row: 'No load balancers'. Below the table, it says 'You don't have any load balancers in us-east-2'. A prominent orange button labeled 'Create load balancer' is located at the bottom right of the table area, also highlighted with a red box.

Enter load balancer name

Select scheme and IP address type.

EC2 > Load balancers > Create Application Load Balancer

Create Application Load Balancer Info

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

▶ How Elastic Load Balancing works

Basic configuration

Load balancer name Info
Name must be unique within your AWS account and can't be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme Info
Scheme can't be changed after the load balancer is created.
 Internet-facing
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)
 Internal
An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type Info
Select the type of IP addresses that your subnets use.
 IPv4
Recommended for internal load balancers.
 Dualstack
Includes IPv4 and IPv6 addresses.

Network mapping should be the same zone with two instances.

Network mapping Info
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC Info
Select the virtual private cloud (VPC) for your targets or you can [create a new VPC](#). Only VPCs with an internet gateway are enabled for selection. The selected VPC can't be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

-
vpc-0d4dfbf0c6d88d5ae
IPv4: 172.31.0.0/16

Mappings Info
Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

<input checked="" type="checkbox"/> us-east-2a (use2-az1)
Subnet
<input type="text" value="subnet-0b1d4d5f95cf2b531"/>
IPv4 address
Assigned by AWS
<input type="checkbox"/> us-east-2b (use2-az2)
<input checked="" type="checkbox"/> us-east-2c (use2-az3)
Subnet
<input type="text" value="subnet-0c159194f5494194d"/>
IPv4 address

Choose a security group created in instances.

Security groups [Info](#)

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups

Select up to 5 security groups ▾ [G](#)

23715251-lab5 [X](#)
sg-0ec108ebc616d034b VPC: vpc-0d4dfbf0c6d88d5ae

[b] Create a target group using the same VPC that you used to create the instances
Listeners and routing

Protocol is HTTP. Port is 80.

Click create target group.

Listeners and routing [Info](#)

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80 [Remove](#)

Protocol Port
HTTP : 80

Default action [Info](#)
Forward to: [Select a target group](#) ▾ [G](#)
[Create target group](#)

Listener tags - optional
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

[Add listener tag](#)
You can add up to 50 more tags.

[Add listener](#)

Enter target group name

Target group name

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol **Port**

HTTP	▼	:	80	▲
------	---	---	----	---

1-65535

IP address type

Only targets with the indicated IP address type can be registered to this target group.

IPv4
Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

IPv6
Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

VPV should be the same with the instances.

instance's default network interface (eth0). [Learn more](#)

VPC

Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

-	▼
- vpc-0d4dfbf0c6d88d5ae IPv4: 172.31.0.0/16	▼

Protocol version

HTTP1
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

HTTP2
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

gRPC
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Click next

Attributes

Info Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

Tags - optional
Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

Cancel **Next**

[c] Register targets in the target group

Choose two instances created.

Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (2/4)

Filter instances 2 matches

Name = 23715251-us-east-2a X
or Name = 23715251-us-east-2b X Clear filters

Instance ID	Name	State	Security groups	Zon
i-065d9ca3cf4ab7895	23715251-us-east-2a	Running	23715251-sg	us-e
i-0f050da4b34b9f98e	23715251-us-east-2b	Running	23715251-sg-2	us-e

2 selected

Click create target group.

0 selected

Ports for the selected instances
Ports for routing traffic to the selected instances.

80
1-65535 (separate multiple ports with commas)

Include as pending below

2 selections are now pending below. Include more or register targets when ready.

Review targets

Targets (2)

Remove	Health status	Instance ID	Name	Port	State	Security groups	Zone
X	Pending	i-035140a8b4cd32a0d	23715251-us-east-2a	80	Running	23715251-lab5	us-east-2a
X	Pending	i-09bab1b72fcf330b3	23715251-us-east-2b	80	Running	23715251-lab5	us-east-2b

2 pending

Cancel Previous Create target group

[d] Create a listener with a default rule Protocol: HTTP and Port 80 forwarding on to the target group.

Choose the target group created.

Listeners and routing [Info](#)

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80

Protocol	Port	Default action	Info
HTTP	: 80	Forward to	23715251-lab5 Target type: Instance, IPv4

Create target group

Listener tags - optional
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add listener tag
You can add up to 50 more tags.

Add listener

Click create.

Successfully created load balancer: 23715251-lab5
Note: It might take a few minutes for your load balancer to be fully set up and ready to route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.

EC2 > Load balancers > 23715251-lab5 > Create Application Load Balancer

Create Application Load Balancer

Suggested next steps

- Review, customize, or configure attributes for your load balancer and listeners using the Description and Listeners tabs within 23715251-lab5.
- Discover other services that you can integrate with your load balancer. Visit the Integrated services tab within 23715251-lab5.

View load balancer

Try and access the EC2 instance using its public IP address in a browser. The load balancer will not be working at this point because Apache 2 is not installed.

Before installing Apache 2 on your instances, you have to update your instances.

```
sudo apt-get update
```

choose the instance, and then click connect.

Instances (1/13) Info		Connect	Instance state	Actions	Launch instances
		Find Instance by attribute or tag (case-sensitive)			
	Name	Instance ID	Instance state	Instance type	Status check
<input type="checkbox"/>	i-04cff07504e1f65f4	i-04cff07504e1f65f4	Running	t2.micro	2/2 checks passed
<input type="checkbox"/>	ziqi-ohio4	i-0514b8539d2b28b46	Running	t2.micro	2/2 checks passed
<input type="checkbox"/>	ziqi-ohio5	i-0ddc6453d2d3f0e7e	Running	t2.micro	2/2 checks passed
<input type="checkbox"/>	ziqi-ohio3	i-0ef95b31bdeff26ab	Running	t2.micro	2/2 checks passed
<input type="checkbox"/>	23715251-us-east-2a	i-065d9ca3cf4ab7895	Stopped	t2.micro	-
<input checked="" type="checkbox"/>	23715251-us-east-2a	i-070e9d19d487b2635	Running	t2.micro	2/2 checks passed

Click connect

Connect to instance Info

Connect to your instance i-070e9d19d487b2635 (23715251-us-east-2a) using any of these options

EC2 Instance Connect Session Manager SSH client EC2 serial console

Instance ID
 [i-070e9d19d487b2635](#) (23715251-us-east-2a)

Connection Type
 Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address
 3.21.98.178

User name
Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ubuntu.

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

Cancel **Connect**

```
ubuntu@ip-172-31-14-235:~$ sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [857 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:10 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:11 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:12 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1059 kB]
Get:13 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [233 kB]
Get:14 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [15.6 kB]
Get:15 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [974 kB]
Get:16 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [175 kB]
Get:17 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [11.4 kB]
Get:18 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [157 kB]
Get:19 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [953 kB]
```

Then, on each instance, install apache2:

```
sudo apt install apache2
```

```
ubuntu@ip-172-31-14-235:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil libaprutil-dbd-sqlite3 libaprutil-ldap liblua5.3-0 mailcap mime-support
  ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser bzip2-doc
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil libaprutil-dbd-sqlite3 libaprutil-ldap liblua5.3-0 mailcap
  mime-support ssl-cert
0 upgraded, 13 newly installed, 0 to remove and 129 not upgraded.
Need to get 2137 kB of archives.
After this operation, 8505 kB of additional disk space will be used.
Do you want to continue? [Y/n] 
```

Edit the /var/www/html/index.html file to report the instance name.

Enter command below to edit the html file.

```
*** System restart required ***
Last login: Wed Oct 11 05:05:49 2023 from 3.16.146.5
ubuntu@ip-172-31-14-235:~$ sudo nano /var/www/html/index.html
```

Enter the instance name in body. Save and exit.

```
div.validator {
}
</style>
</head>
<body>
  <div class="main_page">
    <div class="page_header floating_element">
      
    <div>
      <span style="margin-top: 1.5em;" class="floating_element">
        23715251
      </span>
    </div>
    <div class="banner">
      <div id="about"></div>
      Instance 1
    </div>
  </div>
  <div class="content_section floating_element">
    <div class="content_section_text">
      <p>
        This is the default welcome page used to test the correct
        operation of the Apache2 server after installation on Ubuntu systems.
        It is based on the equivalent page on Debian, from which the Ubuntu Apache
        packaging is derived.
        If you can read this page, it means that the Apache HTTP server installed at
        this site is working properly. You should <b>replace this file</b> (located at
        <tt>/var/www/html/index.html</tt>) before continuing to operate your HTTP server.
      </p>
    </div>
  </div>
</body>
```

```

div.validator {
}
</style>
</head>
<body>
  <div class="main_page">
    <div class="page_header floating_element">
      
    <div>
      <span style="margin-top: 1.5em;" class="floating_element">
        23715251
      </span>
    </div>
    <div class="banner">
      <div id="about"></div>
      Instance 2
    </div>
  </div>
  <div class="content_section floating_element">
    <div class="content_section_text">
      <p>
        This is the default welcome page used to test the correct
        operation of the Apache2 server after installation on Ubuntu systems.
        It is based on the equivalent page on Debian, from which the Ubuntu Apache
        packaging is derived.
        If you can read this page, it means that the Apache HTTP server installed at
        this site is working properly. You should <b>replace this file</b> (located at
        <tt>/var/www/html/index.html</tt>) before continuing to operate your HTTP server.
      </p>
      <p>
        If you are a normal user of this web site and don't know what this page is
        about, this probably means that the site is currently unavailable due to
        maintenance.
        If the problem persists, please contact the site's administrator.
      </p>
    </div>
  </div>

```

215.0-1

Now verify that the load balancer is working: use a browser from your hostOS to access both of the EC2 instances by their respective IP address.



▲ 18.116.86.207



23715251

Ubuntu

Instance 2

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in**

/usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   '-- ports.conf
|-- mods-enabled
|   '-- *.load
|   '-- *.conf
|-- conf-enabled
|   '-- *.conf
|-- sites-enabled
|   '-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.





18.117.226.246



23715251

Ubuntu

Instance 1

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in** [**/usr/share/doc/apache2/README.Debian.gz**](#). Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the [**manual**](#) if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   '-- ports.conf
|-- mods-enabled
|   '-- *.load
|   '-- *.conf
|-- conf-enabled
|   '-- *.conf
|-- sites-enabled
|   '-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.



Lab 6 Practical Worksheet

[Step 1] Create an EC2 instance

[1] Create an EC2 micro instance using Ubuntu and SSH into it. Install the Python 3 virtual environment package.

I have shown the specific steps on how to create an instance. Therefore, here I only display the result of the instance created.

Instance summary for i-088d9b2960ef9e21c (23715251-lab6) Info		
Connect Actions		
Updated less than a minute ago		
Instance ID i-088d9b2960ef9e21c (23715251-lab6)	Public IPv4 address 18.223.133.235 [open address]	Private IPv4 addresses 172.31.41.253
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-18-223-133-235.us-east-2.compute.amazonaws.com [open address]
Hostname type IP name: ip-172-31-41-253.us-east-2.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-41-253.us-east-2.compute.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 18.223.133.235 [Public IP]	VPC ID vpc-0d4dfbf0c6d88d5ae	Auto Scaling Group name -
IAM Role -	Subnet ID subnet-0c159194f5494194d	
IMDSv2 Optional		

I need to connect it by ssh.

```
rongchuansun@RongchuandeMacBook-Pro ~ % cd Desktop
rongchuansun@RongchuandeMacBook-Pro Desktop % ssh -i "23715251-lab5.pem" ubuntu@ec2-18-223-133-235.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-18-223-133-235.us-east-2.compute.amazonaws.com (18.223.133.235)' can't be established.
ED25519 key fingerprint is SHA256:Um9v6iYbuaIAARdk1xWDo4+c6losHskNdInIlpTolbY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yea
Please type 'yes', 'no' or the fingerprint: yea
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'ec2-18-223-133-235.us-east-2.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1025-aws x86_64)

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

System information as of Fri Oct 13 06:29:59 UTC 2023

System load:  0.0          Processes:      96
Usage of /:  27.1% of 7.57GB  Users logged in:   0
Memory usage: 26%           IPv4 address for eth0: 172.31.41.253
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

135 updates can be applied immediately.
76 of these updates are standard security updates.
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: Thu Oct 12 13:24:30 2023 from 130.95.254.23
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-41-253:~$
```

```
sudo apt-get update
sudo apt-get upgrade
```

```
ubuntu@ip-172-31-41-253:~$ sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
ubuntu@ip-172-31-41-253:~$ sudo apt-get upgrade
Reading package lists... done
Building dependency tree... done
Reading state information... done
Calculating upgrade... done
The following packages will be upgraded:
  apport apt apt-utils base-files cloud-init distro-info dpkg ec2-hibinit-agent fwupd-signed git git-man grub-common grub-pc
  grub-pc-bin grub2-common hibagent initramfs-tools initramfs-tools-bin initramfs-tools-core iptables libapt-pkg6.0
  libfwupd2 libfwupdplugin5 libgpgme11 libip4tc2 libip6tc2 libldap-2.5-0 libldap-common libmm-glib0 libnetplan0
  libnss-systemd libpam-systemd libsgutils2-2 libsystemd0 libudev1 libunwind8 libxtables12 mdadm motd-news-config netplan.io
  openssh-client openssh-server openssh-sftp-server python-apt-common python3-apport python3-apt python3-debian
  python3-distro-info python3-distupgrade python3-problem-report python3-software-properties sg3-utils sg3-utils-udev
  software-properties-common sosreport systemd systemd-sysv tzdata ubuntu-advantage-tools ubuntu-minimal
  ubuntu-release-upgrader-core ubuntu-server ubuntu-standard udev ufw
65 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 25.8 MB of archives.
After this operation, 565 kB disk space will be freed.
```

```
sudo apt-get install python3-venv
```

```
[NO VM guests are running outdated hypervisor (QEMU) binaries on this host.  
[ubuntu@ip-172-31-41-253:~$ sudo apt-get install python3-venv  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  python3-pip-whl python3-setuptools-whl python3.10-venv  
The following NEW packages will be installed:  
  python3-pip-whl python3-setuptools-whl python3-venv python3.10-venv  
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.  
Need to get 2474 kB of archives.  
After this operation, 2888 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 python3-pip-whl all 22.0.2+dfsg-1ubuntu0.3 [1679 kB]  
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 python3-setuptools-whl all 59.6.0-1.2ubuntu0.22.04.1 [788 kB]  
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 python3.10-venv amd64 3.10.12-1~22.04.2 [572 4 B]  
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 python3-venv amd64 3.10.6-1~22.04 [1038 B]  
Fetched 2474 kB in 0s (27.0 MB/s)  
Selecting previously unselected package python3-pip-whl.  
(Reading database ... 94165 files and directories currently installed.)
```

It is easier now if you change the bash to operate as sudo

```
sudo bash
```

```
[NO VM guests are running outdated hypervisor (QEMU) binaries on this host.  
[ubuntu@ip-172-31-41-253:~$ sudo bash  
root@ip-172-31-41-253:/home/ubuntu#
```

[2] Create a directory with a path /opt/www/mysites and cd into that. Set up a virtual environment:

```
python3 -m venv myvenv
```

```
[NO VM guests are running outdated hypervisor (QEMU) binaries on this host.  
[ubuntu@ip-172-31-41-253:~$ sudo bash  
[root@ip-172-31-41-253:/home/ubuntu# python3 -m venv myvenv  
[root@ip-172-31-41-253:/home/ubuntu# mkdir opt  
[root@ip-172-31-41-253:/home/ubuntu# cd opt  
[root@ip-172-31-41-253:/home/ubuntu/opt# mkdir /www/mysites  
mkdir: cannot create directory '/www/mysites': No such file or directory  
[root@ip-172-31-41-253:/home/ubuntu/opt# mkdir www  
[root@ip-172-31-41-253:/home/ubuntu/opt# cd www  
[root@ip-172-31-41-253:/home/ubuntu/opt/www# mkdir mysites  
[root@ip-172-31-41-253:/home/ubuntu/opt/www# cd mysites  
[root@ip-172-31-41-253:/home/ubuntu/opt/www/mysites# python3 -m venv myvenv  
[root@ip-172-31-41-253:/home/ubuntu/opt/www/mysites#
```

[3] Activate your virtual environment and then:

```
source myvenv/bin/activate  
pip install Django
```

```

root@ip-172-31-41-253:/home/ubuntu/opt/wwc/mysites# source myvenv/bin/activate
(myvenv) root@ip-172-31-41-253:/home/ubuntu/opt/wwc/mysites# pip install django
Collecting django
  Downloading Django-4.2.6-py3-none-any.whl (8.0 MB)
    ━━━━━━━━━━━━━━━━ 8.0/8.0 MB 17.5 MB/s eta 0:00:00
Collecting asgiref<4,>=3.6.0
  Downloading asgiref-3.7.2-py3-none-any.whl (24 kB)
Collecting sqlparse>=0.3.1
  Downloading sqlparse-0.4.4-py3-none-any.whl (41 kB)
    ━━━━━━━━━━━━━━ 41.2/41.2 KB 6.0 MB/s eta 0:00:00
Collecting typing-extensions>=4
  Downloading typing_extensions-4.8.0-py3-none-any.whl (31 kB)
Installing collected packages: typing-extensions, sqlparse, asgiref, django
Successfully installed asgiref-3.7.2 django-4.2.6 sqlparse-0.4.4 typing-extensions-4.8.0
(myvenv) root@ip-172-31-41-253:/home/ubuntu/opt/wwc/mysites#

```

```

django-admin startproject lab
cd lab
python3 manage.py startapp polls

```

```

(myvenv) root@ip-172-31-41-253:/home/ubuntu/opt/wwc/mysites# django-admin startproject lab
(myvenv) root@ip-172-31-41-253:/home/ubuntu/opt/wwc/mysites# cd lab
(myvenv) root@ip-172-31-41-253:/home/ubuntu/opt/wwc/mysites/lab# ls
lab  manage.py
(myvenv) root@ip-172-31-41-253:/home/ubuntu/opt/wwc/mysites/lab# python3 manage.py startapp polls
(myvenv) root@ip-172-31-41-253:/home/ubuntu/opt/wwc/mysites/lab# ls
lab  manage.py  polls
(myvenv) root@ip-172-31-41-253:/home/ubuntu/opt/wwc/mysites/lab#

```

Stop and look at the files that have been created – the project files are to do with the running of the application. We will deal with the files as we go through.

[Step 2] Install and configure nginx

[1] install nginx

```

apt install nginx

((myvenv) root@ip-172-31-41-253:/home/ubuntu/opt/wwc/mysites/lab# apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3 libjbig0 libjpeg-turbo8 libjpeg8
  libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail
  libnginx-mod-stream libnginx-mod-stream-geoip2 libtiff5 libwebp7 libxpm4 nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3 libjbig0 libjpeg-turbo8 libjpeg8
  libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail
  libnginx-mod-stream libnginx-mod-stream-geoip2 libtiff5 libwebp7 libxpm4 nginx nginx-common nginx-core
0 upgraded, 20 newly installed, 0 to remove and 0 not upgraded.
Need to get 2691 kB of archives.
After this operation, 8339 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 fonts-dejavu-core all 2.37-2build1 [1041 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 fontconfig-config all 2.13.1-4.2ubuntu5 [29.1 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libdeflate0 amd64 1.10-2 [70.9 kB]

```

edit /etc/nginx/sites-enabled/default and replace the contents of the file with

```

server {
  listen 80 default_server;
  listen [::]:80 default_server;

  location / {
    proxy_set_header X-Forwarded-Host $host;
  }
}

```

```

proxy_set_header X-Real-IP $remote_addr;

proxy_pass http://127.0.0.1:8000;
}

(myvenv) root@ip-172-31-41-253:/home/ubuntu/opt/wwc/mysites/lab# vim /etc/nginx/sites-enabled/default
(myvenv) root@ip-172-31-41-253:/home/ubuntu/opt/wwc/mysites/lab# 
#
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    location / {
        proxy_set_header X-Forwarded-Host $host;
        proxy_set_header X-Real-IP $remote_addr;

        proxy_pass http://127.0.0.1:8000;
    }
}

```

[2] Once you have done this you can restart nginx

```
service nginx restart
```

```
((myvenv) root@ip-172-31-41-253:/home/ubuntu/opt/wwc/mysites/lab# vim /etc/nginx/sites-enabled/default
((myvenv) root@ip-172-31-41-253:/home/ubuntu/opt/wwc/mysites/lab# service nginx restart
```

[3] in your app directory: /opt/wwc/mysites/lab you can run

```
python3 manage.py runserver 8000
```

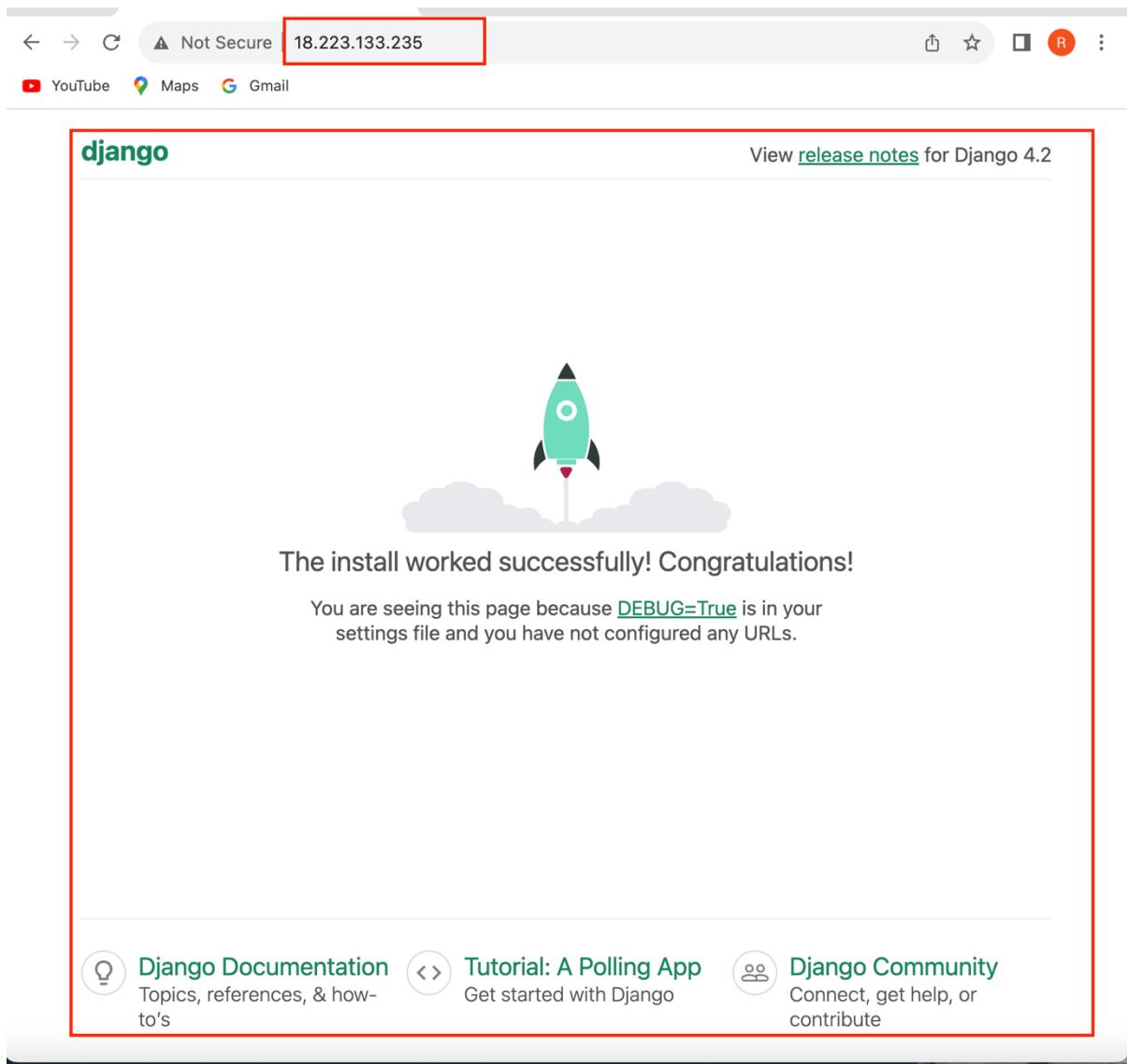
```
((myvenv) root@ip-172-31-41-253:/home/ubuntu/opt/wwc/mysites/lab# python3 manage.py runserver 8000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s)
contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
October 13, 2023 - 07:22:37
Django version 4.2.6, using settings 'lab.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.

Not Found: /polls/
[13/Oct/2023 07:22:45] "GET /polls/ HTTP/1.0" 404 2092
Not Found: /polls/
[13/Oct/2023 07:23:07] "GET /polls/ HTTP/1.0" 404 2092
Not Found: /polls/
[13/Oct/2023 07:23:15] "GET /polls/ HTTP/1.0" 404 2092
Not Found: /polls/
[13/Oct/2023 07:23:37] "GET /polls/ HTTP/1.0" 404 2092
Not Found: /polls/
[13/Oct/2023 07:23:45] "GET /polls/ HTTP/1.0" 404 2092
```

[4] Open a browser and enter the ip address of your ec2 instance, take a screenshot of what you see and stop your server with CONTROL-C



[Step 3] Change the code

[1] Following the steps outlined in the lecture, edit the following files (create the file if it does not exist)

edit polls/views.py

```
from django.http import HttpResponse

def index(request):
    return HttpResponse("Hello, world.")

_____py  admin.py  apps.py  migrations  models.py  tests.py  views.py
(myvenv) root@ip-172-31-41-253:/home/ubuntu/opt/wwc/mysites/lab/polls# vim views.py
(myvenv) root@ip-172-31-41-253:/home/ubuntu/opt/wwc/mysites/lab/polls#
```

```
from django.http import HttpResponseRedirect

def index(request):
    return HttpResponseRedirect("Hello, world.")

~
```

edit polls/urls.py

```
from django.urls import path
from . import views

urlpatterns = [
    path("", views.index, name='index'),
]
```

```
[myvenv] root@ip-172-31-41-253:/home/ubuntu/opt/wwc/mysites/lab/polls# vim urls.py
[myvenv] root@ip-172-31-41-253:/home/ubuntu/opt/wwc/mysites/lab/polls#
```

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.index, name='index'),
]
~
```

edit lab/urls.py

```
(myvenv) root@ip-172-31-41-253:/home/ubuntu/opt/wwc/mysites/lab/lab# vim urls.py
(myvenv) root@ip-172-31-41-253:/home/ubuntu/opt/wwc/mysites/lab/lab#
```

```
from django.urls import include, path
from django.contrib import admin

urlpatterns = [
    path('polls/', include('polls.urls')),
    path('admin/', admin.site.urls),
]
```

[2] now run

```
python3 manage.py runserver 8000
```

```
    Create your views here.  
IndentationError: unexpected indent  
[^C(myvenv) root@ip-172-31-14-235:/home/ubuntu/opt/wwc/mysites/lab# cd polls  
[(myvenv) root@ip-172-31-14-235:/home/ubuntu/opt/wwc/mysites/lab/polls# ls  
__init__.py __pycache__ admin.py apps.py migrations models.py tests.py urls.py views.py  
[(myvenv) root@ip-172-31-14-235:/home/ubuntu/opt/wwc/mysites/lab/polls# vim views.py  
[(myvenv) root@ip-172-31-14-235:/home/ubuntu/opt/wwc/mysites/lab/polls# cd ..  
[(myvenv) root@ip-172-31-14-235:/home/ubuntu/opt/wwc/mysites/lab# python3 manage.py runserver 8000  
Watching for file changes with StatReloader  
Performing system checks...  
  
System check identified no issues (0 silenced).  
October 12, 2023 - 08:55:15  
Django version 4.2.6, using settings 'lab.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CONTROL-C.  
  
[12/Oct/2023 08:55:49] "GET /polls HTTP/1.0" 301 0  
[12/Oct/2023 08:55:50] "GET /polls/ HTTP/1.0" 200 13  
[12/Oct/2023 08:55:56] "GET /polls/ HTTP/1.0" 200 13  
Not Found: /  
[12/Oct/2023 08:56:01] "GET / HTTP/1.0" 404 2162  
^C(myvenv) root@ip-172-31-14-235:/home/ubuntu/opt/wwc/mysites/lab#
```

[3] Type the url `http://<ip address of your EC2 instance>/polls/`, check that you get "Hello, world."
NOTE remember to put the `/polls/` on the end and you may need to restart nginx if it does not work.

16:56 5G

Hello, world.



[Step 4] Add an application load balancer (customize your ALB from the last lab).

[1] Create an application load balancer, specify the region subnet where your EC2 instance resides, create a listener with a default rule Protocol: HTTP and Port 80 forwarding.

Create Application Load Balancer Info

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

► **How Elastic Load Balancing works**

Basic configuration

Load balancer name Info
Name must be unique within your AWS account and can't be changed after the load balancer is created.
 A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme Info
Scheme can't be changed after the load balancer is created.

Internet-facing
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#) Info

Internal
An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type Info
Select the type of IP addresses that your subnets use.

IPv4
Recommended for internal load balancers.

Dualstack
Includes IPv4 and IPv6 addresses.

Choose the same region with the instance.

Network mapping Info

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC Info

Select the virtual private cloud (VPC) for your targets or you can [create a new VPC](#). Only VPCs with an internet gateway are enabled for selection. The selected VPC can't be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

-
vpc-0d4dfbf0c6d88d5ae
IPv4: 172.31.0.0/16



Mappings Info

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

us-east-2a (use2-az1)

Subnet

subnet-0b1d4d5f95cf2b531



IPv4 address

Assigned by AWS

us-east-2b (use2-az2)

Subnet

subnet-0bddae504ae0797aa



IPv4 address

Assigned by AWS

[2] Choose the security group, allowing HTTP traffic.

Security groups Info

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups

Select up to 5 security groups



23715251-lab5

sg-0ec108ebc616d034b VPC: vpc-0d4dfbf0c6d88d5ae



[3] For the target group, in the health check, specify /polls/ for the path.

Due to showing the steps to create the target group in lab 5, I won't show again. The only difference is that you need to add the specific path of polls.

Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

HTTP ▾

Health check path

Use the default path of "/" to ping the root, or specify a custom path if preferred.

/polls/

Up to 1024 characters allowed.

▶ Advanced health check settings

[4] Add your instance as a registered target.

Add the instance created into the target group.

Targets | Monitoring | Health checks | Attributes | Tags

Registered targets (1)

Filter targets

< 1 > ⚙

<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status...
<input type="checkbox"/>	i-088d9b2960ef...	23715251-lab6	80	us-east-2c	✗ unhealthy	Health checks...

Once you have created the ALB, you should see the health check fetch the /polls/ page every 30 seconds

You can now access the site using the url <http://<load balancer dns name>/polls/>

3.21.98.178/polls/ 23715251-lab6-1474235130.us-east-2.elb.amazonaws.com/polls/

Not Secure | 23715251-lab6-1474235130.us-east-2.elb.amazonaws.com/polls/

YouTube Maps Gmail

Hello, world.

Lab 7 Practical Worksheet

[Step 1] Create an EC2 instance

to create an EC2 instance that you will test your Fabric-based installation on.

We need to create an instance for lab 7. We only display the result due to repeat steps.

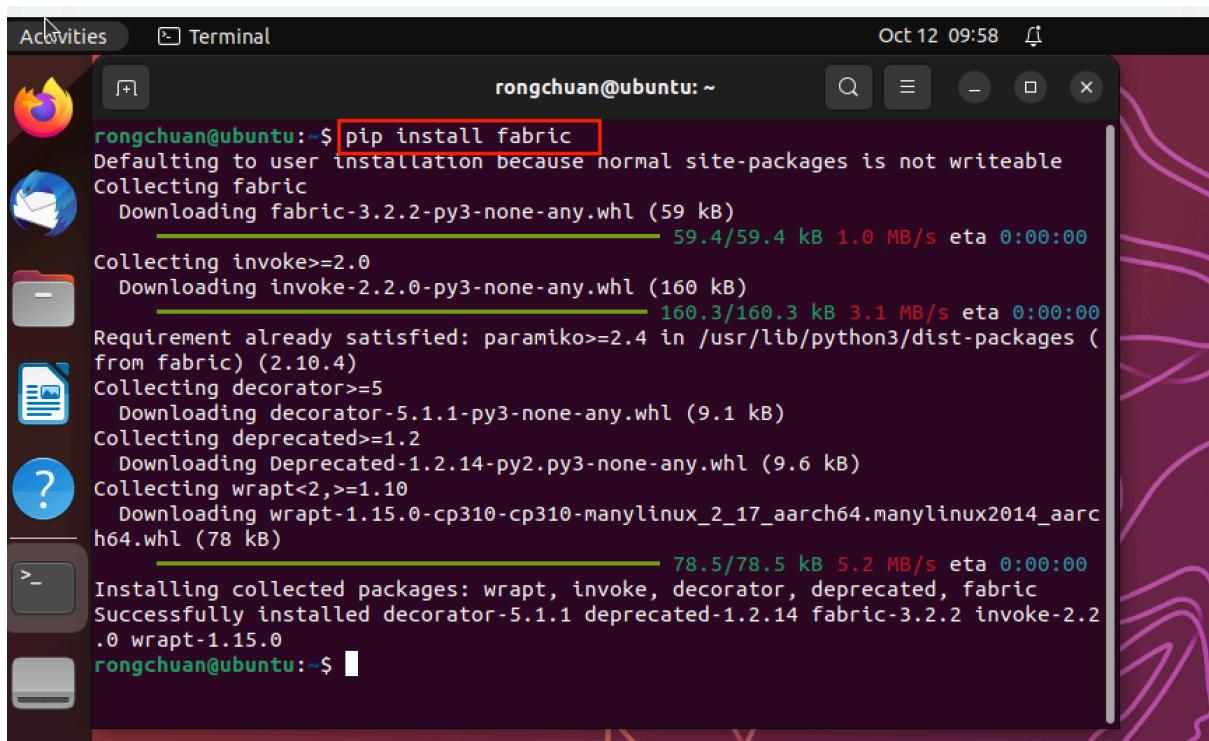
Instance summary for i-00c15e2187add13b4 (23715251-lab7) Info		
	Actions	
Updated less than a minute ago		
Instance ID i-00c15e2187add13b4 (23715251-lab7)	Public IPv4 address 18.216.177.146 [open address]	Private IPv4 addresses 172.31.35.59
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-18-216-177-146.us-east-2.compute.amazonaws.com [open address]
Hostname type IP name: ip-172-31-35-59.us-east-2.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-35-59.us-east-2.compute.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 18.216.177.146 [Public IP]	VPC ID vpc-0d4dfbf0c6d88d5ae	Auto Scaling Group name -
IAM Role -	Subnet ID subnet-0c159194f5494194d	
IMDSv2 Optional		

[Step 2] Install and configure fabric on your VM

NOTE do this on your VirtualBox VM

The easiest way to install fabric is to:

```
pip install fabric
```

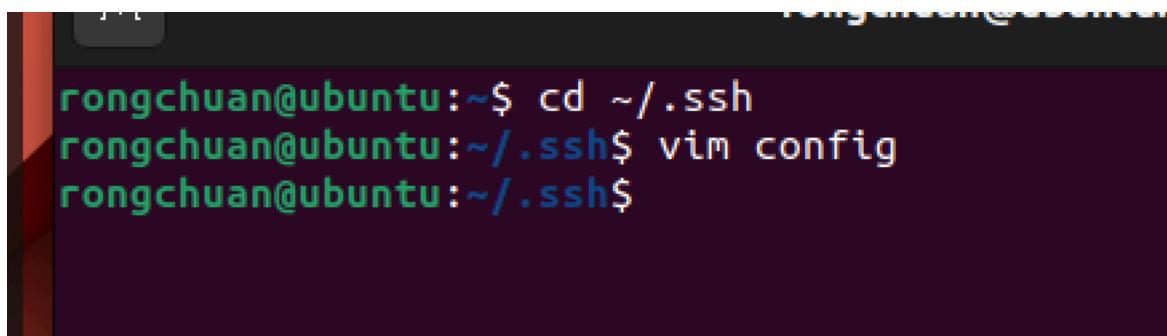


A screenshot of an Ubuntu desktop environment. In the top left, there's a dock with icons for Dash, Home, Terminal, and others. A terminal window is open in the center, showing the command `pip install fabric` being run. The output shows the installation of multiple packages, including fabric, invoke, decorator, deprecated, and wrapt. The terminal window has a dark theme with light-colored text.

```
rongchuan@ubuntu:~$ pip install fabric
Defaulting to user installation because normal site-packages is not writeable
Collecting fabric
  Downloading fabric-3.2.2-py3-none-any.whl (59 kB)
Collecting invoke>=2.0
  Downloading invoke-2.2.0-py3-none-any.whl (160 kB)
Requirement already satisfied: paramiko>=2.4 in /usr/lib/python3/dist-packages (from fabric) (2.10.4)
Collecting decorator>=5
  Downloading decorator-5.1.1-py3-none-any.whl (9.1 kB)
Collecting deprecated>=1.2
  Downloading Deprecated-1.2.14-py2.py3-none-any.whl (9.6 kB)
Collecting wrapt<2,>=1.10
  Downloading wrapt-1.15.0-cp310-cp310-manylinux_2_17_aarch64.manylinux2014_aarch64.whl (78 kB)
Installing collected packages: wrapt, invoke, decorator, deprecated, fabric
Successfully installed decorator-5.1.1 deprecated-1.2.14 fabric-3.2.2 invoke-2.2.0 wrapt-1.15.0
rongchuan@ubuntu:~$
```

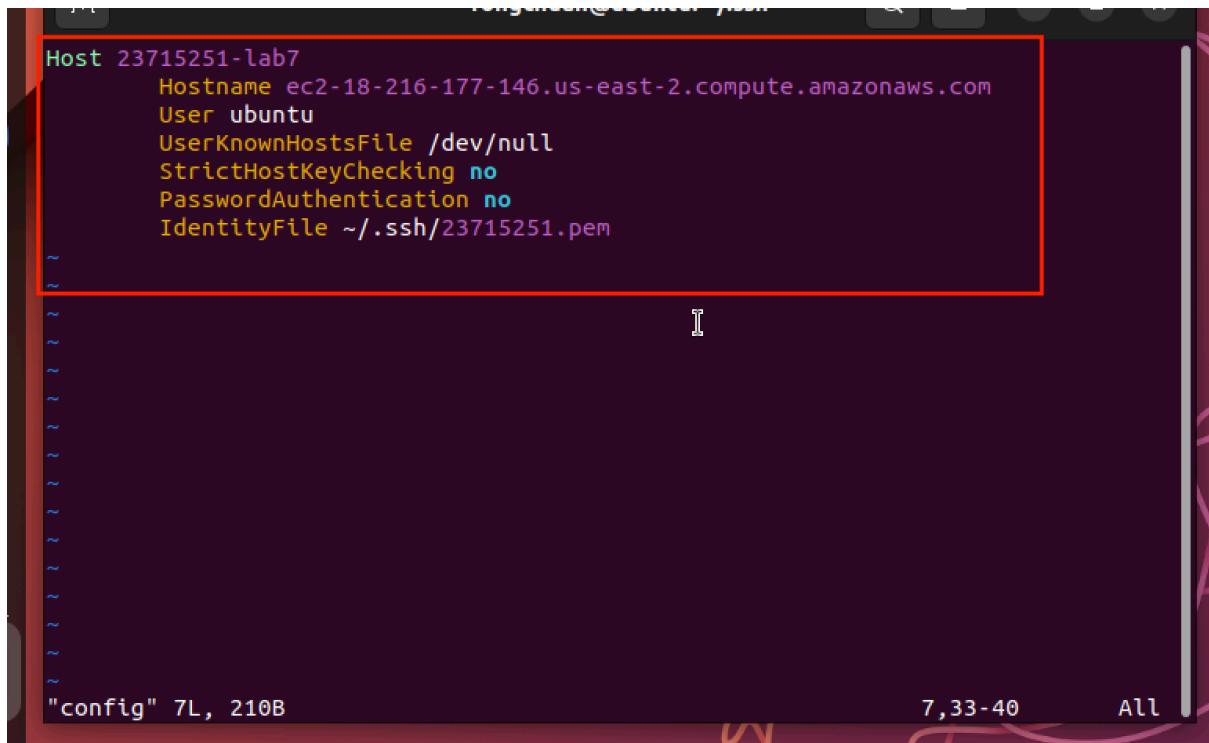
You will need to create a config file in `~/.ssh` with the contents:

```
Host <ec2instance>
  Hostname <EC2 instance public DNS>
  User ubuntu
  UserKnownHostsFile /dev/null
  StrictHostKeyChecking no
  PasswordAuthentication no
  IdentityFile <path to your private key>
```



A screenshot of a terminal window on an Ubuntu system. The user is navigating to the `~/.ssh` directory and opening the `config` file with `vim`. The terminal window has a dark theme.

```
rongchuan@ubuntu:~$ cd ~/.ssh
rongchuan@ubuntu:~/.ssh$ vim config
rongchuan@ubuntu:~/.ssh$
```



A screenshot of a terminal window titled "rongchuan@ec2-18-216-177-146:~". The window displays the contents of a file named "config". The file contains the following configuration:

```
Host 23715251-lab7
  Hostname ec2-18-216-177-146.us-east-2.compute.amazonaws.com
  User ubuntu
  UserKnownHostsFile /dev/null
  StrictHostKeyChecking no
  PasswordAuthentication no
  IdentityFile ~/.ssh/23715251.pem
```

The terminal shows the file has 7 lines and 210B in size. The status bar at the bottom right indicates "7, 33-40" and "All".

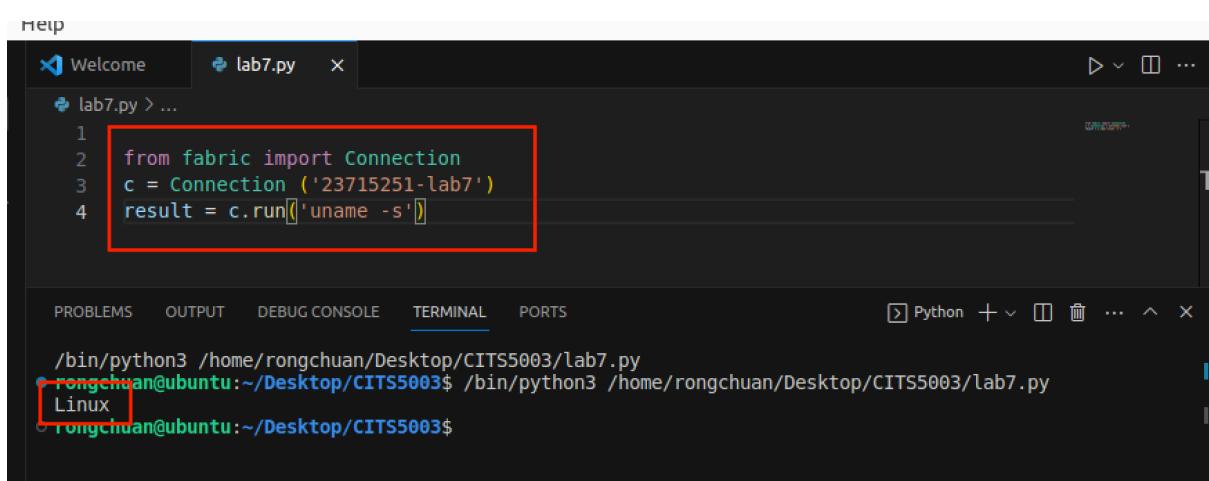
Use ssh to connect to you instance

Remember to replace with your EC2 name you used in the configuration - use your student number to identify resources.

You can test fabric from the command line:

```
python
>>> from fabric import Connection
>>> c = Connection('<ec2instance>')
>>> result = c.run('uname -s')
Linux
>>>
```

I wrote a python script with the code above to test.



A screenshot of the Visual Studio Code (VS Code) interface. The top navigation bar shows "Help", "Welcome", and "lab7.py". The main area shows a code editor with a Python script named "lab7.py". The script contains the following code:

```
1  from fabric import Connection
2  c = Connection ('23715251-lab7')
3  result = c.run(['uname -s'])
```

The code editor has a red box highlighting the first three lines of the script. Below the editor is a terminal window showing the execution of the script and its output:

```
/bin/python3 /home/rongchuan/Desktop/CITS5003/lab7.py
rongchuan@ubuntu:~/Desktop/CITS5003$ /bin/python3 /home/rongchuan/Desktop/CITS5003/lab7.py
Linux
rongchuan@ubuntu:~/Desktop/CITS5003$
```

The terminal window has a red box highlighting the word "Linux" in the output.

[Step 3] Write a python script to automate the installation of nginx

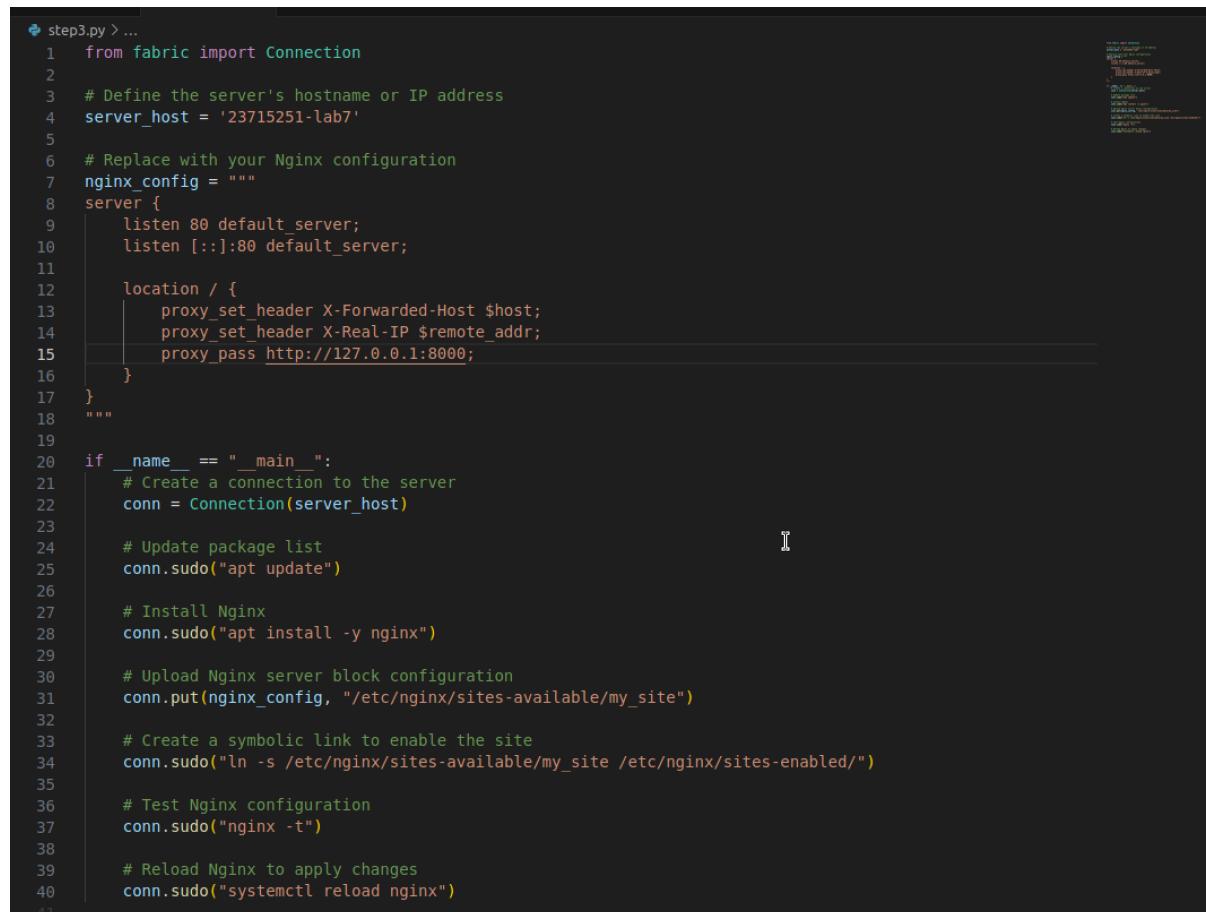
Write a python script using fabric to set up and configure nginx as you did for the Django app last week

This will consist of doing the same commands you would do manually to configure nginx but using the commands:

```
sudo("commands go here separated by ;")
```

and/or

```
run("Commands go here separated by ;")
```



```
❷ step3.py > ...
1  from fabric import Connection
2
3  # Define the server's hostname or IP address
4  server_host = '23715251-lab7'
5
6  # Replace with your Nginx configuration
7  nginx_config = """
8      server {
9          listen 80 default_server;
10         listen [::]:80 default_server;
11
12         location / {
13             proxy_set_header X-Forwarded-Host $host;
14             proxy_set_header X-Real-IP $remote_addr;
15             proxy_pass http://127.0.0.1:8000;
16         }
17     }
18 """
19
20 if __name__ == "__main__":
21     # Create a connection to the server
22     conn = Connection(server_host)
23
24     # Update package list
25     conn.sudo("apt update")
26
27     # Install Nginx
28     conn.sudo("apt install -y nginx")
29
30     # Upload Nginx server block configuration
31     conn.put(nginx_config, "/etc/nginx/sites-available/my_site")
32
33     # Create a symbolic link to enable the site
34     conn.sudo("ln -s /etc/nginx/sites-available/my_site /etc/nginx/sites-enabled/")
35
36     # Test Nginx configuration
37     conn.sudo("nginx -t")
38
39     # Reload Nginx to apply changes
40     conn.sudo("systemctl reload nginx")
41
```

Import Fabric:

```
from fabric import Connection
```

This line imports the Connection class from the Fabric library, which is used to connect to and interact with remote servers.

Define server_host:

```
server_host = '23715251-lab7'
```

This line specifies the hostname or IP address of the remote server you want to configure.

Nginx Configuration:

```
nginx_config = """
server {
    listen 80 default_server;
    listen [::]:80 default_server;
    location / {
        proxy_set_header X-Forwarded-Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_pass http://127.0.0.1:8000;
    }
}
"""
....
```

This block of text represents the Nginx server block configuration. It configures Nginx to listen on port 80 and forward incoming requests to a local application running on 127.0.0.1:8000. This is a basic reverse proxy configuration. You can customize this configuration to match your specific use case.

Install Nginx:

```
conn.sudo("apt install -y nginx")
```

Upload Nginx Configuration:

```
conn.put(nginx_config, "/etc/nginx/sites-available/my_site")
```

This line uploads the Nginx server block configuration specified in `nginx_config` to the server and saves it as `my_site` in the `/etc/nginx/sites-available/` directory.

Reload Nginx:

```
conn.sudo("systemctl reload nginx")
```

This command reloads Nginx to apply the new configuration without stopping the service. The web server should now be running with the updated configuration.

[Step 4] Update the python script to install your Django app

You will copy the files from your local directory to the new EC2 instance. You may find this easier if you create a tar/zip file first of the code and then untar/unzip on the EC2 machine.

Add the necessary commands to configure the virtual environment and clone your Django app from your local directory - this is basically taking the instructions you used in that lab and converting them to Fabric commands.

The final command should be the command to run the server - if you add an '&' at the end it will run the process in the background. Note - you would normally use Gunicorn to do this not manage.py runserver but we are keeping it simple.

```
❸ step-4.py > deploy_django_app
1   from fabric import Connection, task
2   import os
3
4   # Define the server's hostname or IP address
5   server_host = '23715251-lab7'
6   remote_dir = '/home/ubuntu' # The directory where you want to copy your Django app
7   local_app_archive = 'myproject.tar.gz' # The name of your local Django app archive
8   remote_app_archive = 'django_app.tar.gz' # The name of the remote Django app archive
9   venv_path = 'venv' # The name of your virtual environment
10  local_requirements_file = 'requirements.txt' # The name of your local requirements file
11
12 def deploy_django_app(c):
13     # Connect to the server
14     conn = Connection(c)
15
16     # Upload your Django app archive to the remote server
17     conn.put(local_app_archive, remote_dir + '/' + remote_app_archive)
18
19     # Upload the requirements.txt file to the remote server
20     conn.put(local_requirements_file, remote_dir + '/requirements.txt')
21
22     # Go to the remote directory
23     with conn.cd(remote_dir):
24         # Extract the Django app archive
25         conn.run(f'tar -xzf {remote_app_archive}')
26
27         # Create or activate the virtual environment
28         conn.sudo("apt update")
29         conn.sudo("apt install python3-venv")
30         conn.sudo("bash")
31         conn.run("python3 -m venv myenv")
32
33         conn.run(f'cd {remote_dir} && source myenv/bin/activate')
34
35
36         # Install app dependencies using pip
37         conn.run(f'pip install -r {remote_dir}/requirements.txt')
38
39         # Migrate the database
40         conn.run(f'python {remote_dir}/myproject/manage.py migrate')
41
42         # Start the Django server (in the background)
43         conn.run(f'python {remote_dir}/myproject/manage.py runserver &', pty=False)
44
45 if __name__ == "__main__":
46     # Execute the deployment task on the specified server
47     deploy_django_app(server_host)
```

Define `server_host` and other variables:

```
server_host = '23715251-lab7'
remote_dir = '/home/ubuntu'
local_app_archive = 'myproject.tar.gz'
remote_app_archive = 'django_app.tar.gz'
venv_path = 'venv'
local_requirements_file = 'requirements.txt'
```

These variables define the server's hostname or IP address, remote directory, names of local and remote app archives, virtual environment path, and the name of the local requirements file.

Define the `deploy_django_app` function:

```
def deploy_django_app(c):
    # Connect to the server
    conn = Connection(c)
```

This function is used to deploy the Django app to the remote server. It takes a connection (c) as an argument and establishes a connection to the remote server.

[Upload Django app archive and requirements.txt:](#)

```
conn.put(local_app_archive, remote_dir + '/' + remote_app_archive)
conn.put(local_requirements_file, remote_dir + '/requirements.txt')
```

[Change to the remote directory and extract the app archive:](#)

```
with conn.cd(remote_dir):
    conn.run(f'tar -xzf {remote_app_archive}')
```

It changes the working directory to the remote directory and then extracts the Django app archive using the tar command.

[Create and activate the virtual environment:](#)

```
conn.sudo("apt update")
conn.sudo("apt install python3-venv")
conn.sudo("bash")
conn.run("python3 -m venv myenv")
conn.run(f'cd {remote_dir} && source myenv/bin/activate')
```

This part ensures that Python 3's venv module is installed, creates a virtual environment named myenv, and activates it. This is where the Django app's dependencies will be installed.

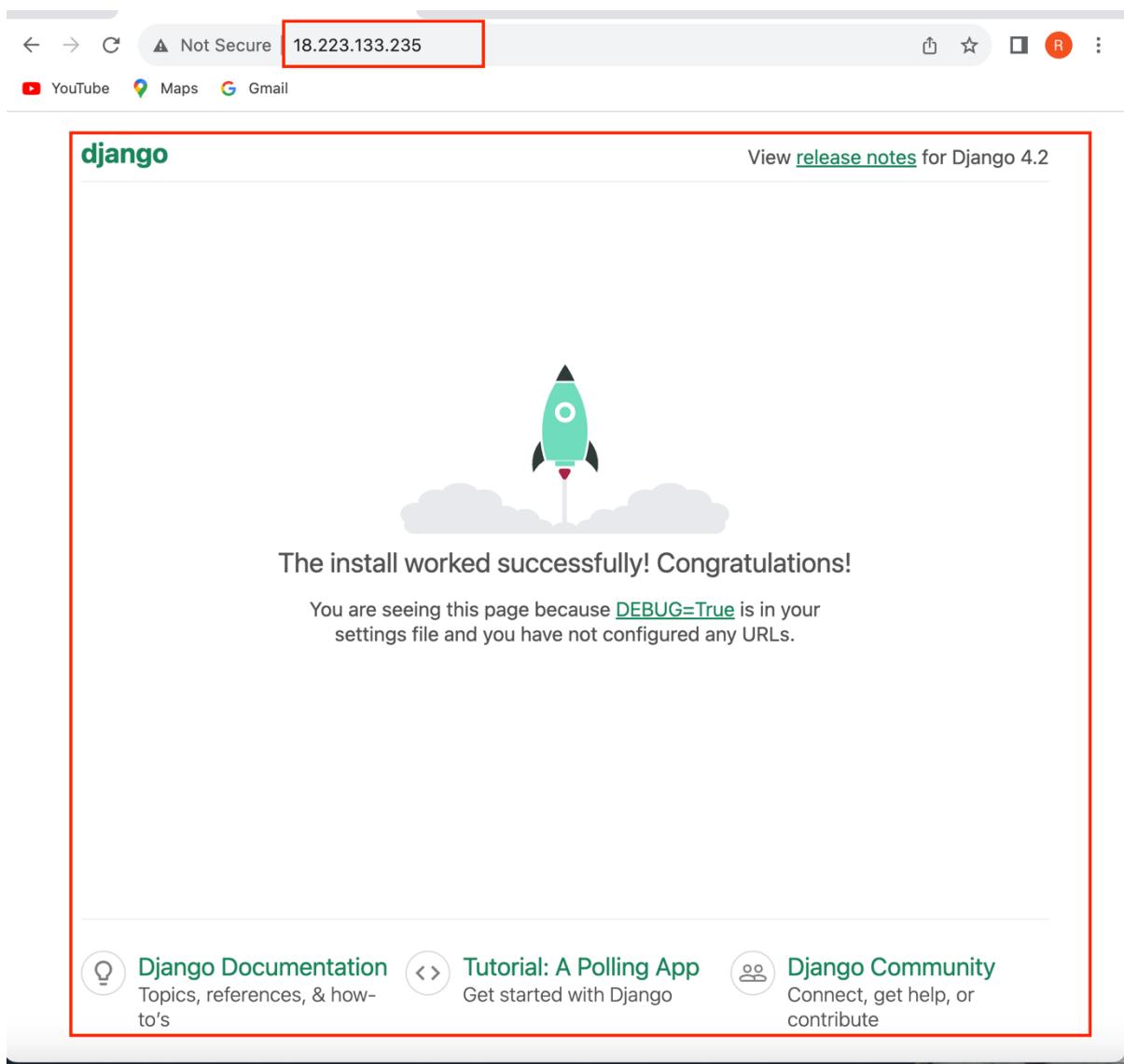
[Migrate the database:](#)

```
conn.run(f'python {remote_dir}/myproject/manage.py migrate')
```

[Start the Django server in the background:](#)

```
conn.run(f'python {remote_dir}/myproject/manage.py runserver &', pty=False)
```

This command starts the Django development server in the background so that it continues running even after the SSH session is closed.



Lab8 AI

Install and run jupyter notebooks

```
pip install notebook
```

```
rongchuan@ubuntu:~$ pip install notebook
Defaulting to user installation because normal site-packages is not writeable
Collecting notebook
  Downloading notebook-7.0.5-py3-none-any.whl (4.0 MB)
    4.0/4.0 MB 11.7 MB/s eta 0:00:00
Collecting tornado>=6.2.0
  Downloading tornado-6.3.3-cp38-abi3-manylinux_2_17_aarch64_manylinux2014_aarch
  64.whl (428 kB)
    428.1/428.1 kB 21.3 MB/s eta 0:00:00
Collecting jupyter-server<3,>=2.4.0
  Downloading jupyter_server-2.7.3-py3-none-any.whl (375 kB)
    375.6/375.6 kB 22.8 MB/s eta 0:00:00
Collecting jupyterlab-server<3,>=2.22.1
  Downloading jupyterlab_server-2.25.0-py3-none-any.whl (57 kB)
    57.2/57.2 kB 14.7 MB/s eta 0:00:00
Collecting notebook-shim<0.3,>=0.2
  Downloading notebook_shim-0.2.3-py3-none-any.whl (13 kB)
Collecting jupyterlab<5,>=4.0.2
  Downloading jupyterlab-4.0.7-py3-none-any.whl (9.2 MB)
    9.2/9.2 kB 23.8 MB/s eta 0:00:00
Collecting argon2-cffi
  Downloading argon2_cffi-23.1.0-py3-none-any.whl (15 kB)
Collecting jupyter-events>=0.6.0
  Downloading jupyter_events-0.7.0-py3-none-any.whl (18 kB)
```

jupyter notebook

```
rongchuan@ubuntu:~$ jupyter notebook
[I 2023-10-13 03:21:47.058 ServerApp] Package notebook took 0.0000s to import
[I 2023-10-13 03:21:47.062 ServerApp] Package jupyter_lsp took 0.0044s to import
[W 2023-10-13 03:21:47.063 ServerApp] A `jupyter_server_extension_points` function was not found in jupyter_lsp. Instead, a `jupyter_server_extension_paths` function was found and will be used for now. This function name will be deprecated in future releases of Jupyter Server.
[I 2023-10-13 03:21:47.066 ServerApp] Package jupyter_server_terminals took 0.0035s to import
[I 2023-10-13 03:21:47.066 ServerApp] Package jupyterlab took 0.0000s to import
[I 2023-10-13 03:21:47.081 ServerApp] Package notebook_shim took 0.0000s to import
[W 2023-10-13 03:21:47.081 ServerApp] A `jupyter_server_extension_points` function was not found in notebook_shim. Instead, a `jupyter_server_extension_paths` function was found and will be used for now. This function name will be deprecated in future releases of Jupyter Server.
[I 2023-10-13 03:21:47.081 ServerApp] jupyter_lsp | extension was successfully linked.
```

Set Up Python Environment

This lab is best done using a Python virtual environment for your packages, or within your own VM. If you are using a virtual environment, remember to activate it prior to installing packages.

To run the SageMaker commands contained within the notebook the following packages are required:

- sagemaker
- pandas

- ipykernel

This can be installed using pip:

```
pip install sagemaker pandas ipykernel
```

or

```
pip3 install sagemaker pandas ipykernel
```

```
rongchuan@ubuntu:~/Desktop$ cd ..
rongchuan@ubuntu:~$ pip3 install sagemaker pandas ipykernel
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: sagemaker in ./local/lib/python3.10/site-packages (2.192.0)
Requirement already satisfied: pandas in ./local/lib/python3.10/site-packages (2.1.1)
Requirement already satisfied: ipykernel in ./local/lib/python3.10/site-packages (6.25.2)
Requirement already satisfied: schema in ./local/lib/python3.10/site-packages (from sagemaker) (0.7.5)
Requirement already satisfied: packaging>=20.0 in ./local/lib/python3.10/site-packages (from sagemaker) (23.2)
Requirement already satisfied: PyYAML~=6.0 in ./local/lib/python3.10/site-packages (from sagemaker) (6.0.1)
Requirement already satisfied: attrs<24,>=23.1.0 in ./local/lib/python3.10/site-packages (from sagemaker) (23.1.0)
Requirement already satisfied: pathos in ./local/lib/python3.10/site-packages (from sagemaker) (0.3.1)
Requirement already satisfied: boto3<2.0,>=1.26.131 in ./local/lib/python3.10/site-packages (from sagemaker) (1.28.22)
Requirement already satisfied: platformdirs in ./local/lib/python3.10/site-packages
```

Run Hyperparameter Tuning Jobs

Create a bucket

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with the 'Amazon S3' logo at the top, followed by a 'Buckets' tab which is highlighted with a red box. Below the tabs are links for 'Access Points', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', and 'IAM Access Analyzer for S3'. Further down, there's a 'Block Public Access settings for this account' section and a 'Storage Lens' section with 'Dashboards' and 'AWS Organizations settings' options. At the bottom of the sidebar, there's a 'Feature spotlight' section with a blue button containing the number '7'. The main content area has a header 'Amazon S3 > Buckets'. It features an 'Account snapshot' section with statistics: 'Total storage' (544.9 MB), 'Object count' (431), 'Average object size' (1.3 MB), and a note about enabling advanced metrics. Below this is a 'Buckets (89)' section with an 'Info' link, a note that buckets are containers for data stored in S3, and four buttons: 'Create bucket' (highlighted with a red box), 'Copy ARN', 'Empty', and 'Delete'. There's also a search bar labeled 'Find buckets by name'.

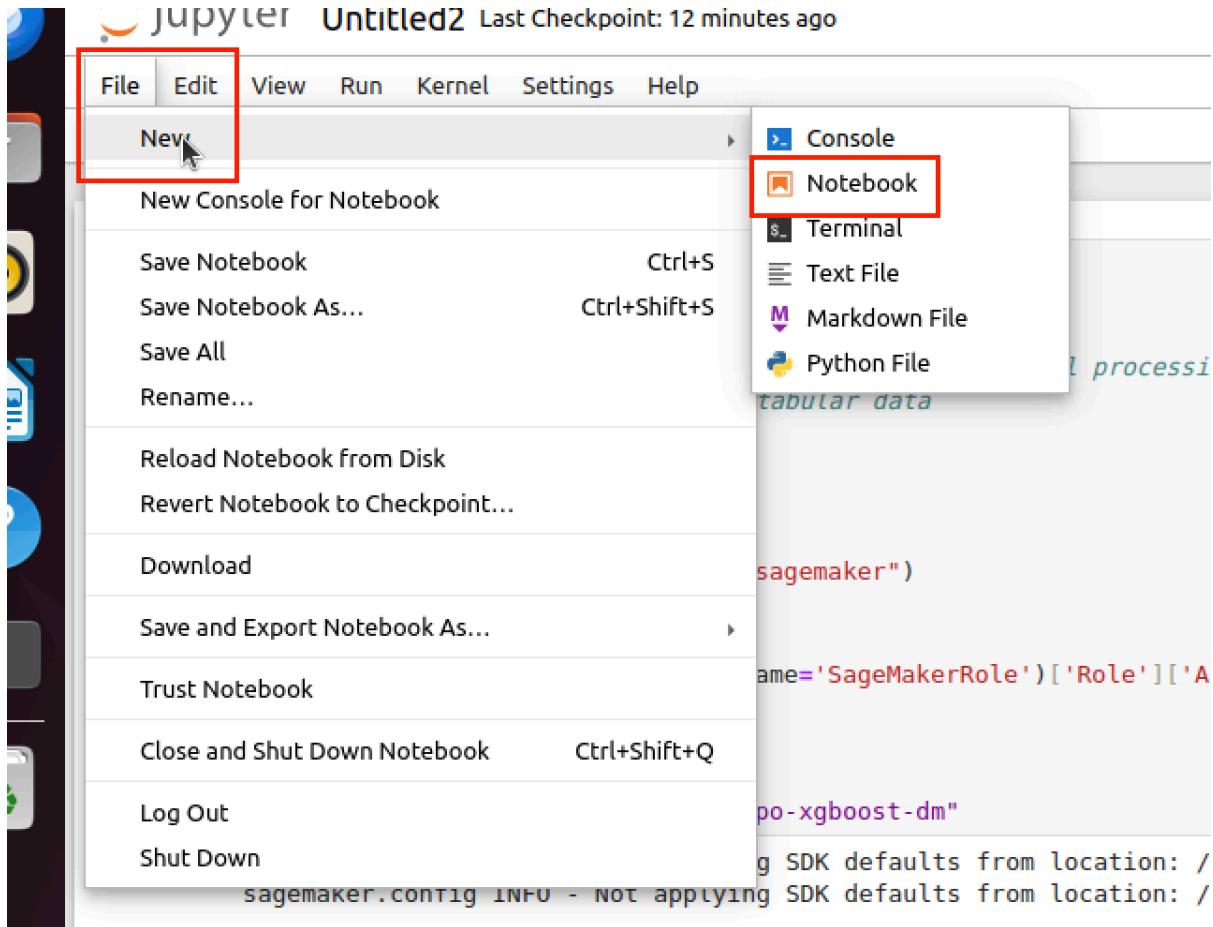
Buckets (89) Info				
Buckets are containers for data stored in S3. Learn more				
	 Copy ARN	Empty	Delete	Create bucket
<input type="text"/> 23715251 X				1 match
<	1	>		
Name	AWS Region	Access	Creation date	
23715251-lab8	US East (Ohio) us-east-2	Bucket and objects not public	October 14, 2023, 18:40:30 (UTC+08:00)	

Check role name in IAM.

Roles (15) Info		
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.		
	 Delete	Create role
<input type="text"/> Search		
<	1	>
<input type="checkbox"/> Role name		
<input type="checkbox"/>	AWSServiceRoleForGlobalAccelerator	
<input type="checkbox"/>	AWSServiceRoleForKeyManagementServiceMultiRegionKeys	
<input type="checkbox"/>	AWSServiceRoleForResourceExplorer	
<input type="checkbox"/>	AWSServiceRoleForServiceQuotas	
<input type="checkbox"/>	AWSServiceRoleForSupport	
<input type="checkbox"/>	AWSServiceRoleForTrustedAdvisor	
<input type="checkbox"/>	iamreadaccessrule	
<input type="checkbox"/>	SageMakerRole	

Prepare SageMaker session

Run jupyter notebook open notebook
Click file, new and then notebook



Run code in a box. Replace the region, rolename, studentID and bucketName.

```
[1]: import sagemaker
import boto3

import numpy as np # For matrix operations and numerical processing
import pandas as pd # For munging tabular data
from time import gmtime, strftime
import os

region = 'us-east-2'
smclient = boto3.Session().client("sagemaker")

iam = boto3.client('iam')
sagemaker_role = iam.get_role(RoleName='SageMakerRole')['Role']['Arn']

student_id = "23715251"
bucket = '23715251-lab8'
prefix = f"sagemaker/{student_id}-hpo-xgboost-dm"

sagemaker.config INFO - Not applying SDK defaults from location: /etc/xdg/xdg-ubuntu/sagemaker/config.yaml
sagemaker.config INFO - Not applying SDK defaults from location: /home/rongchuan/.config/sagemaker/config.yaml
```

Download Dataset

Please take some time to read about the data with more detail [here](#) Let's start by downloading the direct marketing dataset from UCI's ML Repository.

You can download the dataset manually or use the commands below. These commands should work for Linux and MacOS users.

```
[2]: !wget -N https://archive.ics.uci.edu/ml/machine-learning-databases/00222/bank-additional.zip  
!unzip -o bank-additional.zip  
  
--2023-10-15 08:18:38-- https://archive.ics.uci.edu/ml/machine-learning-databases/00222/bank-additional.zip  
Resolving archive.ics.uci.edu (archive.ics.uci.edu)... 128.195.10.252  
Connecting to archive.ics.uci.edu (archive.ics.uci.edu)|128.195.10.252|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: unspecified  
Saving to: 'bank-additional.zip'  
  
bank-additional.zip      [          =>          ] 434.15K   206KB/s   in 2.1s  
  
Last-modified header missing -- time-stamps turned off.  
2023-10-15 08:18:41 (206 KB/s) - 'bank-additional.zip' saved [444572]  
  
Archive: bank-additional.zip  
inflating: bank-additional/.DS_Store  
inflating: __MACOSX/bank-additional/.DS_Store  
inflating: bank-additional/.Rhistory  
inflating: bank-additional/bank-additional-full.csv  
inflating: bank-additional/bank-additional-names.txt  
inflating: bank-additional/bank-additional.csv  
inflating: __MACOSX/.bank-additional
```

Now lets read this into a Pandas data frame and take a look at the data.

```
[3]:  
* data = pd.read_csv("./bank-additional/bank-additional-full.csv", sep=";")  
pd.set_option("display.max_columns", 500) # Make sure we can see all of the columns  
pd.set_option("display.max_rows", 50) # Keep the output on one page  
data  
  
[3]:  
   age    job marital education default housing loan contact month day_of_week duration campaign pdays previous poutcome emp.  
0   56  housemaid married basic.4y   no    no    no  telephone   may     mon     261      1    999      0 nonexistent  
1   57    services married high.school unknown   no    no    no  telephone   may     mon     149      1    999      0 nonexistent  
2   37    services married high.school      no   yes   no  telephone   may     mon     226      1    999      0 nonexistent  
3   40    admin. married basic.6y      no   no   no  telephone   may     mon     151      1    999      0 nonexistent  
4   56    services married high.school      no   no   yes  telephone   may     mon     307      1    999      0 nonexistent  
... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ...  
41183  73    retired married professional.course      no   yes   no cellular nov     fri     334      1    999      0 nonexistent  
41184  46  blue-collar married professional.course      no   no   no cellular nov     fri     383      1    999      0 nonexistent  
41185  56    retired married university.degree      no   yes   no cellular nov     fri     189      2    999      0 nonexistent  
41186  44 technician married professional.course      no   no   no cellular nov     fri     442      1    999      0 nonexistent  
41187  74    retired married professional.course      no   yes   no cellular nov     fri     239      3    999      1    failure  
  
41188 rows x 21 columns
```

```
[3]: data = pd.read_csv("./bank-additional/bank-additional-full.csv", sep=";")
pd.set_option("display.max_columns", 500) # Make sure we can see all of the columns
pd.set_option("display.max_rows", 50) # Keep the output on one page
data
```

using	loan	contact	month	day_of_week	duration	campaign	pdays	previous	poutcome	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	y
no	no	telephone	may	mon	261	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	no	telephone	may	mon	149	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
yes	no	telephone	may	mon	226	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	no	telephone	may	mon	151	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	yes	telephone	may	mon	307	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
...
yes	no	cellular	nov	fri	334	1	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	yes
no	no	cellular	nov	fri	383	1	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	no
yes	no	cellular	nov	fri	189	2	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	no
no	no	cellular	nov	fri	442	1	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	yes
yes	no	cellular	nov	fri	239	3	999	1	failure	-1.1	94.767	-50.8	1.028	4963.6	no

Answer the following questions: - Which variables are categorical? - Which ones are numerical?

Numerical variables: the below picture is numerical variables.

```
[4]: data["no_previous_contact"] = np.where(
    data["pdays"] == 999, 1, 0
) # Indicator variable to capture when pdays takes a value of 999
data["not_working"] = np.where(
    np.in1d(data["job"], ["student", "retired", "unemployed"]), 1, 0
) # Indicator for individuals not actively employed
model_data = pd.get_dummies(data) # Convert categorical variables to sets of indicators
model_data
```

[5]:

	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	no_previous_contact	not_working	job_admin.	...
0	56	261	1	999	0	1.1	93.994	-36.4	4.857	5191.0	1	0	False	...
1	57	149	1	999	0	1.1	93.994	-36.4	4.857	5191.0	1	0	False	...
2	37	226	1	999	0	1.1	93.994	-36.4	4.857	5191.0	1	0	False	...
3	40	151	1	999	0	1.1	93.994	-36.4	4.857	5191.0	1	0	True	...
4	56	307	1	999	0	1.1	93.994	-36.4	4.857	5191.0	1	0	False	...
...
41183	73	334	1	999	0	-1.1	94.767	-50.8	1.028	4963.6	1	1	False	...
41184	46	383	1	999	0	-1.1	94.767	-50.8	1.028	4963.6	1	0	False	...
41185	56	189	2	999	0	-1.1	94.767	-50.8	1.028	4963.6	1	1	False	...
41186	44	442	1	999	0	-1.1	94.767	-50.8	1.028	4963.6	1	0	False	...
41187	74	239	3	999	1	-1.1	94.767	-50.8	1.028	4963.6	1	1	False	...

41188 rows × 67 columns

Categorical variables: the below picture is categorical variables.

Let's remove the economic features and duration from our data as they would need to be forecasted with high precision to use as inputs in future predictions.

```
[5]: model_data = model_data.drop(["duration", "emp.var.rate", "cons.price.idx", "cons.conf.idx", "euribor3m", "nr.employed"], axis=1, )
[6]: model_data
```

[6]:

	age	campaign	pdays	previous	no_previous_contact	not_working	job_admin.	job_blue-collar	job_entrepreneur	job_housemaid	job_management	job_retried	j
0	56	1	999	0		1	0	False	False	False	True	False	False
1	57	1	999	0		1	0	False	False	False	False	False	False
2	37	1	999	0		1	0	False	False	False	False	False	False
3	40	1	999	0		1	0	True	False	False	False	False	False
4	56	1	999	0		1	0	False	False	False	False	False	False
...
41183	73	1	999	0		1	1	False	False	False	False	False	True
41184	46	1	999	0		1	0	False	True	False	False	False	False
41185	56	2	999	0		1	1	False	False	False	False	False	True
41186	44	1	999	0		1	0	False	False	False	False	False	False
41187	74	3	999	1		1	1	False	False	False	False	False	True

41188 rows × 61 columns

```
[5]: model_data = model_data.drop(["duration", "emp.var.rate", "cons.price.idx", "cons.conf.idx", "euribor3m", "nr.employed"], axis=1, )
[6]: model_data
```

[6]:

it	job_retired	job_self-employed	job_services	job_student	job_technician	job_unemployed	job_unknown	marital_divorced	marital_married	marital_single	marital_unknown
e	False	False	False	False	False	False	False	False	True	False	False
e	False	False	True	False	False	False	False	False	True	False	False
e	False	False	True	False	False	False	False	False	True	False	False
e	False	False	False	False	False	False	False	False	True	False	False
e	False	False	True	False	False	False	False	False	True	False	False
...
e	True	False	False	False	False	False	False	False	True	False	False
e	False	False	False	False	False	False	False	False	True	False	False
e	True	False	False	False	False	False	False	False	True	False	False
e	False	False	False	False	True	False	False	False	True	False	False
e	True	False	False	False	False	False	False	False	True	False	False

Split Data into training, validation and test

We'll then split the dataset into training (70%), validation (20%), and test (10%) datasets and convert the datasets to the right format the algorithm expects. We will use training and validation datasets during training. Test dataset will be used to evaluate model performance after it is deployed to an endpoint.

Amazon SageMaker's XGBoost algorithm expects data in the libSVM or CSV data format. For this lab, we'll stick to CSV. Note that the first column must be the target variable and the CSV should not include headers. Also, notice that although repetitive it's easier to do this

after the train|validation|test split rather than before. This avoids any misalignment issues due to random reordering.

```
[7]: train_data, validation_data, test_data = np.split(
    model_data.sample(frac=1, random_state=1729),
    [int(0.7 * len(model_data)), int(0.9 * len(model_data))],
)

pd.concat([train_data["y_yes"], train_data.drop(["y_no", "y_yes"], axis=1)], axis=1).to_csv(
    "train.csv", index=False, header=False
)
pd.concat(
    [validation_data["y_yes"], validation_data.drop(["y_no", "y_yes"], axis=1)], axis=1
).to_csv("validation.csv", index=False, header=False)
pd.concat([test_data["y_yes"], test_data.drop(["y_no", "y_yes"], axis=1)], axis=1).to_csv(
    "test.csv", index=False, header=False
)
```

Now we'll copy the file to S3 for Amazon SageMaker training to pickup.

```
[9]: boto3.Session().resource("s3").Bucket(bucket).Object(
    os.path.join(prefix, "train/train.csv")
).upload_file("train.csv")
boto3.Session().resource("s3").Bucket(bucket).Object(
    os.path.join(prefix, "validation/validation.csv")
).upload_file("validation.csv")
```

Setup Hyperparameter Optimization

```
[10]: from time import gmtime, strftime, sleep

# Names have to be unique. You will get an error if you reuse the same name
tuning_job_name = f"{23715251}-xgboost-tuningjob-01"

print(tuning_job_name)
tuning_job_config = {
    "ParameterRanges": {
        "CategoricalParameterRanges": [],
        "ContinuousParameterRanges": [
            {
                "MaxValue": "1",
                "MinValue": "0",
                "Name": "eta",
            },
            {
                "MaxValue": "10",
                "MinValue": "1",
                "Name": "min_child_weight",
            },
            {
                "MaxValue": "2",
                "MinValue": "0",
                "Name": "alpha",
            },
        ],
    },
}
```

```

        "MinValue": "0",
        "Name": "eta",
      },
      {
        "MaxValue": "10",
        "MinValue": "1",
        "Name": "min_child_weight",
      },
      {
        "MaxValue": "2",
        "MinValue": "0",
        "Name": "alpha",
      },
    ],
    "IntegerParameterRanges": [
      {
        "MaxValue": "10",
        "MinValue": "1",
        "Name": "max_depth",
      }
    ],
  },
  "ResourceLimits": {"MaxNumberOfTrainingJobs": 2, "MaxParallelTrainingJobs": 2},
  "Strategy": "Bayesian",
  "HyperParameterTuningJobObjective": {"MetricName": "validation:auc", "Type": "Maximize"},
}

```

23715251-xgboost-tuningjob-01

```

[11]: from sagemaker.image_uris import retrieve
# Use XGBoost algorithm for training
training_image = retrieve(framework="xgboost", region=region, version="latest")

s3_input_train = "s3://{}//train".format(bucket, prefix)
s3_input_validation = "s3://{}//validation".format(bucket, prefix)

training_job_definition = {
    "AlgorithmSpecification": {"TrainingImage": training_image, "TrainingInputMode": "File"},
    "InputDataConfig": [
        {
            "ChannelName": "train",
            "CompressionType": "None",
            "ContentType": "csv",
            "DataSource": {
                "S3DataSource": {
                    "S3DataDistributionType": "FullyReplicated",
                    "S3DataType": "S3Prefix",
                    "S3Uri": s3_input_train,
                }
            },
        },
        {
            "S3Uri": s3_input_train,
        }
    ],
    {
        "ChannelName": "validation",
        "CompressionType": "None",
        "ContentType": "csv",
        "DataSource": {
            "S3DataSource": {
                "S3DataDistributionType": "FullyReplicated",
                "S3DataType": "S3Prefix",
                "S3Uri": s3_input_validation,
            }
        },
    },
    "OutputDataConfig": {"S3OutputPath": "s3://{}//output".format(bucket, prefix)},
    "ResourceConfig": {"InstanceCount": 1, "InstanceType": "ml.m5.xlarge", "VolumeSizeInGB": 10},
    "RoleArn": sagemaker_role,
    "StaticHyperParameters": {
        "eval_metric": "auc",
        "num_round": "1",
        "objective": "binary:logistic",
        "rate_drop": "0.3",
        "tweedie_variance_power": "1.4",
    },
    "StoppingCondition": {"MaxRuntimeInSeconds": 43200},
}

```

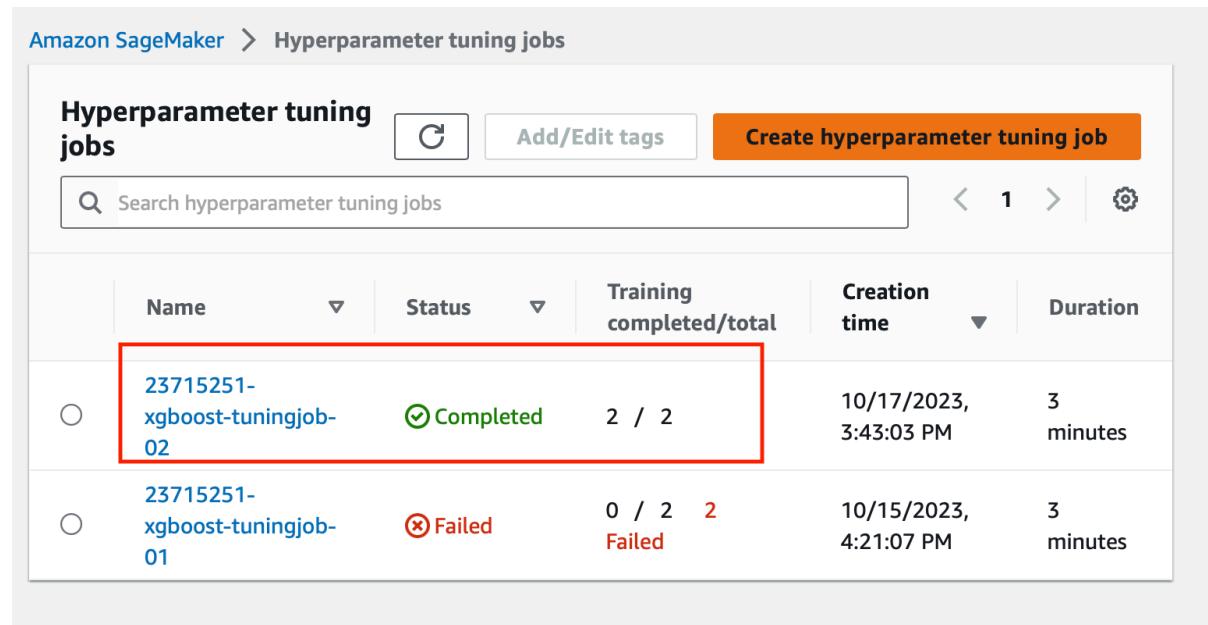
Now we can launch a hyperparameter tuning job by calling `create_hyper_parameter_tuning_job` API. After the hyperparameter tuning job is created, we can go to SageMaker console to track the progress of the hyperparameter tuning job until it is completed.

While Training you can take screenshots of the jobs you just launched on SageMaker-> Training -> Hyperparameter tuning jobs

```
[12]: #Launch Hyperparameter Tuning Job
smclient.create_hyper_parameter_tuning_job(
    HyperParameterTuningJobName=tuning_job_name,
    HyperParameterTuningJobConfig=tuning_job_config,
    TrainingJobDefinition=training_job_definition,
)

[12]: {'HyperParameterTuningJobArn': 'arn:aws:sagemaker:us-east-2:489389878001:hyper-parameter-tuning-job/23715251-xgboost-tuningjob-01',
       'ResponseMetadata': {'RequestId': 'f7345c49-5703-4807-9664-bb1ffcf2e98e',
                           'HTTPStatusCode': 200,
                           'HTTPHeaders': {'x-amzn-requestid': 'f7345c49-5703-4807-9664-bb1ffcf2e98e',
                                          'content-type': 'application/x-amz-json-1.1',
                                          'content-length': '130',
                                          'date': 'Sun, 15 Oct 2023 08:21:07 GMT'},
                           'RetryAttempts': 0}}
```

After running, we can check sagemaker and other file.



The screenshot shows the Amazon SageMaker console with the URL "Amazon SageMaker > Hyperparameter tuning jobs". The page displays a list of hyperparameter tuning jobs. The columns are: Name, Status, Training completed/total, Creation time, and Duration. There are two entries:

	Name	Status	Training completed/total	Creation time	Duration
<input type="radio"/>	23715251-xgboost-tuningjob-02	Completed	2 / 2	10/17/2023, 3:43:03 PM	3 minutes
<input type="radio"/>	23715251-xgboost-tuningjob-01	Failed	0 / 2 2 Failed	10/15/2023, 4:21:07 PM	3 minutes

23715251-lab8 Info

Objects Properties Permissions Metrics Management Access Points

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

 Find objects by prefix

 Show versions

< 1 > 

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	 sagemaker/	Folder	-	-	-

Amazon S3 > Buckets > [23715251-lab8](#) > [sagemaker/](#)

[Copy S3 URI](#)

sagemaker/

[Objects](#) [Properties](#)

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

[Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#)

[Actions ▾](#) [Create folder](#) [Upload](#)

Find objects by prefix

Show versions ◀ 1 ▶ ⚙

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	23715251-hpo-xgboost-dm/	Folder	-	-	-

Amazon S3 > Buckets > 23715251-lab8 > sagemaker/ > 23715251-hpo-xgboost-dm/

23715251-hpo-xgboost-dm/

Objects | Properties

Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Action Buttons:

Search Bar:

Show versions 1

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	output/	Folder	-	-	-
<input type="checkbox"/>	train/	Folder	-	-	-
<input type="checkbox"/>	validation/	Folder	-	-	-

Lab9 MoreAI

AWS Comprehend

AWS Comprehend offers different services to analyse text using machine learning. With Comprehend API, you will be able to perform common NLP tasks such as sentiment analysis, or simply detecting the language from the text.

"Amazon Comprehend can discover the meaning and relationships in text from customer support incidents, product reviews, social media feeds, news articles, documents, and other sources. For example, you can identify the feature that's most often mentioned when customers are happy or unhappy about your product."

For example, to detect the language used in a given text using boto3 you can use the following code:

Run code in notebook.

```
[1]: import boto3
client = boto3.client('comprehend')

# Detect Entities
response = client.detect_dominant_language(
    Text="The French Revolution was a period of social and political upheaval in France and its colonies beginning in 1789 and ending in 1799"
)

print(response['Languages'])

[{'LanguageCode': 'en', 'Score': 0.9983963966369629}]
```

This means that the detected language is 'en' (English) and has a confidence in the prediction greater than 0.99.

[Step1] Detecting Languages from text

[Step 1.1] Modify code

Based on the previous code, create a script that can recognize different languages. However, instead of language code (e.g., 'en' for English or 'it' for Italian) it should return the message "<predicted_language> detected with xx% confidence" where <predicted_language> correspond to the name of the language in English and the confidence (x.xx) is given as a percentage. For the previous example the result should look like this:

Output:

English detected with 99% confidence

```
[1]: import boto3
from langcodes import Language

client = boto3.client('comprehend')

# Detect Dominant Language
response = client.detect_dominant_language(
    Text="The French Revolution was a period of social and political upheaval in France and its colonies beginning in 1789 and ending in 1799"
)

# Extract the detected language and confidence score
detected_languages = response['Languages']

if detected_languages:
    # Sort the detected languages by confidence score (highest first)
    detected_languages.sort(key=lambda lang: lang['Score'], reverse=True)

    # Get the most confident language
    most_confident_language = detected_languages[0]

    # Get the language code from the detected language
    language_code = most_confident_language['LanguageCode']

    # Get the confidence score as a percentage
    confidence_percentage = most_confident_language['Score'] * 100

    # Use the langcodes library to get the English language name
    language_name = Language.make(language_code).display_name('en')

    output_message = f"{language_name} detected with {confidence_percentage:.2f}% confidence"
    print(output_message)
else:
    print("No language detected")
```

English detected with 99.84% confidence

[Step 1.2] Test your code with other languages

Test your code using the following texts in different languages:

Spanish: "El Quijote es la obra más conocida de Miguel de Cervantes Saavedra. Publicada su primera parte con el título de El ingenioso hidalgo don Quijote de la Mancha a comienzos de 1605, es una de las obras más destacadas de la literatura española y la literatura universal, y una de las más traducidas. En 1615 aparecería la segunda parte del Quijote de Cervantes con el título de El ingenioso caballero don Quijote de la Mancha."

```
[2]: import boto3
from langcodes import Language

client = boto3.client('comprehend')

# Detect Dominant Language
response = client.detect_dominant_language(
    Text="El Quijote es la obra más conocida de Miguel de Cervantes Saavedra. Publicada su primera parte con el título de El ingenioso hidalgo don Quijote de la Mancha a comienzos de 1605, es una de las obras más destacadas de la literatura española y la literatura universal, y una de las más traducidas. En 1615 aparecería la segunda parte del Quijote de Cervantes con el título de El ingenioso caballero don Quijote de la Mancha.")

# Extract the detected language and confidence score
detected_languages = response['Languages']

if detected_languages:
    # Sort the detected languages by confidence score (highest first)
    detected_languages.sort(key=lambda lang: lang['Score'], reverse=True)

    # Get the most confident language
    most_confident_language = detected_languages[0]

    # Get the language code from the detected language
    language_code = most_confident_language['LanguageCode']

    # Get the confidence score as a percentage
    confidence_percentage = most_confident_language['Score'] * 100

    # Use the langcodes library to get the English language name
    language_name = Language.make(language_code).display_name('en')

    output_message = f"{language_name} detected with {confidence_percentage:.2f}% confidence"
    print(output_message)
else:
    print("No language detected")
Spanish detected with 99.92% confidence
```

French: "Moi je n'étais rien Et voilà qu'aujourd'hui Je suis le gardien Du sommeil de ses nuits
Je l'aime à mourir Vous pouvez détruire Tout ce qu'il vous plaira Elle n'a qu'à ouvrir L'espace
de ses bras Pour tout reconstruire Pour tout reconstruire Je l'aime à mourir" [From the
Song: "Je l'Aime à Mourir" - Francis Cabrel]

```
[3]: import boto3
from langcodes import Language

client = boto3.client('comprehend')

# Detect Dominant Language
response = client.detect_dominant_language(
    Text="Moi je n'étais rien Et voilà qu'aujourd'hui Je suis le gardien Du sommeil de ses nuits Je l'aime à mourir Vous pouvez détruire Tout
)

# Extract the detected language and confidence score
detected_languages = response['Languages']

if detected_languages:
    # Sort the detected languages by confidence score (highest first)
    detected_languages.sort(key=lambda lang: lang['Score'], reverse=True)

    # Get the most confident language
    most_confident_language = detected_languages[0]

    # Get the language code from the detected language
    language_code = most_confident_language['LanguageCode']

    # Get the confidence score as a percentage
    confidence_percentage = most_confident_language['Score'] * 100

    # Use the langcodes library to get the English language name
    language_name = Language.make(language_code).display_name('en')

    output_message = f"{language_name} detected with {confidence_percentage:.2f}% confidence"
    print(output_message)
else:
    print("No language detected")

French detected with 99.88% confidence
```

Italian: "L'amor che move il sole e l'altre stelle." [Quote from "Divine Comedy" - Dante Alighieri]

```
[4]: import boto3
from langcodes import Language

client = boto3.client('comprehend')

# Detect Dominant Language
response = client.detect_dominant_language(
    Text="L'amor che move il sole e l'altre stelle."
)

# Extract the detected language and confidence score
detected_languages = response['Languages']

if detected_languages:
    # Sort the detected languages by confidence score (highest first)
    detected_languages.sort(key=lambda lang: lang['Score'], reverse=True)

    # Get the most confident language
    most_confident_language = detected_languages[0]

    # Get the language code from the detected language
    language_code = most_confident_language['LanguageCode']

    # Get the confidence score as a percentage
    confidence_percentage = most_confident_language['Score'] * 100

    # Use the langcodes library to get the English language name
    language_name = Language.make(language_code).display_name('en')

    output_message = f"{language_name} detected with {confidence_percentage:.2f}% confidence"
    print(output_message)
else:
    print("No language detected")

Italian detected with 99.65% confidence
```

[Step 2] Sentiment Analysis

Sentiment analysis (or opinion mining) uses NLP to determine whether data is positive, negative or neutral. Sentiment analysis is often performed on textual data to help businesses monitor brand and product sentiment in customer feedback, and understand customer needs.

Task: Use boto3 and AWS comprehend to create a python script for sentiment analysis. Use the previous 3 texts to test the script.

```
[6]: import boto3

# Initialize the AWS Comprehend client
client = boto3.client('comprehend')

# List of texts for sentiment analysis
texts = [
    "The French Revolution was a period of social and political upheaval in France and its colonies beginning in 1789 and ending in 1799.",
    "I absolutely love this product! It's amazing and works perfectly.",
    "I am not satisfied with the quality of the service I received. It was a terrible experience."
]

# Perform sentiment analysis for each text
for text in texts:
    response = client.detect_sentiment(Text=text, LanguageCode='en')
    sentiment = response['Sentiment']
    sentiment_score = response['SentimentScore']

    # Print the sentiment and sentiment score
    print(f"Sentiment: {sentiment}")
    print(f"Sentiment Score: {sentiment_score}")
    print()

Sentiment: NEUTRAL
Sentiment Score: {'Positive': 0.00021114654373377562, 'Negative': 0.00032865439425222576, 'Neutral': 0.9994567036628723, 'Mixed': 3.514676336
6356026e-06}

Sentiment: POSITIVE
Sentiment Score: {'Positive': 0.9998317956924438, 'Negative': 2.6213283490506e-05, 'Neutral': 7.574496703455225e-05, 'Mixed': 6.6205386247020
21e-05}

Sentiment: NEGATIVE
Sentiment Score: {'Positive': 0.00011092302156612277, 'Negative': 0.9996529817581177, 'Neutral': 8.811983570922166e-05, 'Mixed': 0.0001480063
5108258575}
```

[Step 3] Repeat steps from [Step2] for detecting entities.

Question 1: In your words describe what entities are.

Entities, in the context of natural language processing and text analysis, are specific pieces of information or elements within a text that have a well-defined meaning. These entities can represent various types of objects, such as people, organizations, dates, locations, products, and more. They are often nouns or noun phrases that refer to something distinct and identifiable in the text.

[Step 4] Repeat steps from [Step2] for detecting keyphrases.

Question 1: In your words describe what keyphrases are.

Keyphrases, in the context of natural language processing and text analysis, are specific words or phrases within a piece of text that are considered to be particularly informative, significant, or relevant to the content's meaning or subject matter. Keyphrases represent the essential concepts or ideas expressed in the text, and they help in summarizing the primary topics and information contained in the document.

[Step 5] Repeat steps from [Step2] for detecting syntax.

Question 1: In your words describe what syntaxes are.

Syntax is essential for understanding the grammatical structure of language and how words and phrases are combined to convey meaning. It plays a crucial role in various natural language processing tasks, including parsing, grammar checking, and machine translation. Syntax analysis helps computers understand the structure of text, which is valuable for tasks like language generation, information extraction, and text-to-speech synthesis.

AWS Rekognition

AWS Rekognition is the service of AWS that allows you to perform machine learning tasks on images.

Currently, given an image, AWS Rekognition allows you to:

1. **Label Recognition:** Automatically label objects, concepts, scenes, and actions in your images, and provide a confidence score.
2. **Image Moderation:** Automatically detect explicit or suggestive adult content, or violent content in your images, and provide confidence scores.
3. **Facial Analysis:** Get a complete analysis of facial attributes, including confidence scores
4. **Detect Text from an image:** Automatically detect and extract text in your images.

[Step 6] In an S3 bucket add some images to test your algorithms.

1. Add an image in an urban setting (To test later your label recognition script)



2. Add an image of a person on the beach (To test Image moderation)



3. Add an image with people showing their faces (For facial analysis)



4. Add an image with text (Extract text from an image)

**It was the best of
times, it was the worst
of times, it was the age
of wisdom, it was the
age of foolishness...**

Download the four images from google.
Create a new bucket for this lab.
Click bucket in S3, and then click create bucket.

The screenshot shows the AWS S3 console. On the left, a sidebar menu is open under 'Amazon S3' with the 'Buckets' option selected. The main content area displays an 'Account snapshot' with metrics: Total storage (543.9 MB), Object count (441), and Average object size (1.2 MB). A note says you can enable advanced metrics in the 'default-account-dashboard' configuration. Below this is a table titled 'Buckets (96) Info' showing a list of buckets. The 'Create bucket' button is highlighted with a red box. The first bucket listed is '23715251-lab8' in the US East (Ohio) region.

Click objects in the bucket created, and then click upload.

The screenshot shows the 'Objects' tab for the '23715251-lab9' bucket. The 'Upload' button is highlighted with a red box. Other buttons visible include 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and search/filter options.

Click add file, and then select the images downloaded.

The screenshot shows a user interface for uploading files. At the top, there is a dashed blue rectangular area with the text "Drag and drop files and folders you want to upload here, or choose Add files or Add folder." Below this, a button labeled "Add files" is highlighted with a red box. The main area is titled "Files and folders (0)" and contains the sub-instruction "All files and folders in this table will be uploaded." A search bar with the placeholder "Find by name" and a page navigation bar with arrows and the number "1" are also visible.

Review the images in the bucket.

	Name	Folder	Type	Size	
No files or folders					
You have not chosen any files or folders to upload.					
<input type="checkbox"/>	urban.jpeg	jpeg	October 15, 2023, 21:47:50 (UTC+08:00)	121.3 KB	Standard
<input type="checkbox"/>	text.png	png	October 15, 2023, 22:00:09 (UTC+08:00)	107.3 KB	Standard
<input type="checkbox"/>	face.jpeg	jpeg	October 15, 2023, 22:00:16 (UTC+08:00)	1013.6 KB	Standard
<input type="checkbox"/>	beach.webp	webp	October 15, 2023, 22:00:12 (UTC+08:00)	231.9 KB	Standard
<input type="checkbox"/>	Beach.jpeg	jpeg	October 15, 2023, 22:17:46 (UTC+08:00)	114.5 KB	Standard

[Step 7] Create scripts using boto3 and rekognition to test label recognition, image moderation, facial analysis and extracting text from images.

Test label recognition

```
[7]: import boto3

# Initialize the AWS Rekognition client
client = boto3.client('rekognition')

# Specify the S3 bucket and image name
bucket = '23715251-lab9'
image = 'urban.jpeg'

# Perform label recognition
response = client.detect_labels(
    Image={
        'S3Object': {
            'Bucket': bucket,
            'Name': image,
        }
    }
)

# Print the detected labels
print("Detected Labels:")
for label in response['Labels']:
    print(f"Label: {label['Name']} (Confidence: {label['Confidence']:.2f})")


```

Detected Labels:

Label: City (Confidence: 100.00)
Label: Urban (Confidence: 100.00)
Label: Architecture (Confidence: 100.00)
Label: Building (Confidence: 100.00)
Label: Cityscape (Confidence: 100.00)
Label: High Rise (Confidence: 99.77)
Label: Tower (Confidence: 93.91)
Label: Apartment Building (Confidence: 89.66)
Label: Downtown (Confidence: 89.35)
Label: Metropolis (Confidence: 84.34)
Label: Outdoors (Confidence: 63.23)
Label: Neighborhood (Confidence: 56.79)
Label: Condo (Confidence: 56.36)
Label: Housing (Confidence: 56.36)
Label: Office Building (Confidence: 55.69)
Label: Skyscraper (Confidence: 55.36)

Test image moderation

```
[10]: import boto3

rekognition = boto3.client('rekognition')

bucket_name = "23715251-lab9"
object_key = "Beach.jpeg"

response = rekognition.detect_labels(
    Image={
        'S3Object': {
            'Bucket': bucket_name,
            'Name': object_key
        }
    }
)

print("Detected Labels:")
for label in response['Labels']:
    print(f"Label: {label['Name']} (Confidence: {label['Confidence']:.2f})")


```

```

Detected Labels:
Label: Back (Confidence: 99.95)
Label: Body Part (Confidence: 99.95)
Label: Person (Confidence: 99.95)
Label: Adult (Confidence: 99.69)
Label: Female (Confidence: 99.69)
Label: Woman (Confidence: 99.69)
Label: Accessories (Confidence: 96.22)
Label: Glasses (Confidence: 96.22)
Label: Fun (Confidence: 95.17)
Label: Vacation (Confidence: 95.17)
Label: Bracelet (Confidence: 90.96)
Label: Jewelry (Confidence: 90.96)
Label: Nature (Confidence: 83.76)
Label: Outdoors (Confidence: 83.76)
Label: Sea (Confidence: 83.76)
Label: Water (Confidence: 83.76)
Label: Clothing (Confidence: 82.72)
Label: Shorts (Confidence: 82.72)
Label: People (Confidence: 81.93)
Label: Wristwatch (Confidence: 80.90)
Label: Footwear (Confidence: 73.03)
Label: Shoe (Confidence: 73.03)
Label: Beach (Confidence: 65.59)
Label: Coast (Confidence: 65.59)
Label: Shoreline (Confidence: 65.59)
Label: Electronics (Confidence: 57.94)
Label: Face (Confidence: 57.31)
Label: Head (Confidence: 57.31)
Label: Camera (Confidence: 56.40)
Label: Swimwear (Confidence: 55.37)

```



Test facial analysis

```

]: import boto3

# Initialize the AWS Rekognition client
client = boto3.client('rekognition')

# Specify the S3 bucket and image name
bucket = '23715251-lab9'
image = 'face.jpeg'

# Perform facial analysis
response = client.detect_faces(
    Image={
        'S3Object': {
            'Bucket': bucket,
            'Name': image,
        }
    },
    Attributes=['ALL'] # You can specify the attributes you want to analyze
)

# Print the detected facial attributes
print("Detected Facial Attributes:")
for face_detail in response['FaceDetails']:
    # FaceId is not directly accessible
    print(f"Gender: {face_detail['Gender']['Value']} (Confidence: {face_detail['Gender']['Confidence']:.2f})")
    print(f"Age Range: {face_detail['AgeRange']['Low']} - {face_detail['AgeRange']['High']}")
    print(f"Emotions: {', '.join([emotion['Type'] for emotion in face_detail['Emotions']])}")

# Print other general information about the faces detected
print("Number of faces detected:", len(response['FaceDetails']))

```

```

Detected Facial Attributes:
Gender: Female (Confidence: 99.91)
Age Range: 16 - 24
Emotions: HAPPY, SURPRISED, FEAR, SAD, CALM, ANGRY, DISGUSTED, CONFUSED
Gender: Female (Confidence: 99.92)
Age Range: 18 - 24
Emotions: SAD, HAPPY, CALM, FEAR, SURPRISED, ANGRY, CONFUSED, DISGUSTED
Gender: Female (Confidence: 99.98)
Age Range: 22 - 30
Emotions: HAPPY, CALM, SURPRISED, FEAR, SAD, ANGRY, CONFUSED, DISGUSTED
Number of faces detected: 3

```

Test extracting from image

```
[15]: import boto3

# Initialize the AWS Rekognition client
client = boto3.client('rekognition')

# Specify the S3 bucket and image name
bucket = '23715251-lab9'
image = 'text.png'

# Perform text detection
response = client.detect_text(
    Image={
        'S3Object': {
            'Bucket': bucket,
            'Name': image,
        }
    }
)

# Extract and print the detected text as a single sentence
detected_text = " ".join([item['DetectedText'] for item in response['TextDetections']])
print("Detected Text as a Single Sentence:", detected_text)
```

Detected Text as a Single Sentence: It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness... It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness...

End

